Enabling Data Trustworthiness and User Privacy in Mobile Crowdsensing

Haiqin Wu, *Student Member, IEEE*, Liangmin Wang, *Member, ACM*,
Guoliang Xue, *Fellow, IEEE*, Jian Tang, *Fellow, IEEE*, and Dejun Yang, *Senior Member, IEEE*

***Abstract*— Ubiquitous mobile devices with rich sensors and advanced communication capabilities have given rise to mobile crowdsensing systems. The diverse reliabilities of mobile users and the openness of sensing paradigms raise concerns for data trustworthiness, user privacy, and incentive provision. Instead of considering these issues as isolated modules in most existing researches, we comprehensively capture both conflict and inner-relationship among them. In this paper, we propose a holistic solution for trustworthy and privacy-aware mobile crowdsensing with no need of a trusted third party. Specifically, leveraging cryptographic technologies, we devise a series of protocols to enable benign users to request tasks, contribute their data, and earn rewards anonymously without any data linkability. Meanwhile, an anonymous trust/reputation model is seamlessly integrated into our scheme, which acts as reference for our fair incentive design, and provides evidence to detect malicious users who degrade the data trustworthiness. Particularly, we first propose the idea of limiting the number of issued pseudonyms which serves to efficiently tackle the anonymity abuse issue. Security analysis demonstrates that our proposed scheme achieves stronger security with resilience against possible collusion attacks. Extensive simulations are presented which demonstrate the efficiency and practicality of our scheme.**

***Index Terms*— Mobile crowdsensing, data trustworthiness, user privacy, incentive fairness.**

I. Introduction

THE recent proliferation of mobile devices (*e.g.*, smartphones and smartwatches) have given rise to a novel sensing paradigm, namely mobile crowdsensing. With a rich set of on-board powerful sensors (*e.g.*, camera, GPS, and biomedical sensor), as well as the advanced communication technologies (*e.g.*, 4G/5G, WiFi, and Bluetooth), mobile users are able to jointly participate in crowdsensing tasks and flexibly collect sensory data from their nearby environments or activities. Due to many inherent benefits such as lightweight deployment cost, rich sensing resources, and large-scale spatial-temporal coverage, a variety of mobile crowdsensing applications have been fostered, which spans different domains including environmental monitoring, business, smart transportation, and assistive healthcare ([9], [10], [24], [29]).

Despite these promising applications, three key issues still exist in mobile crowdsensing that might critically hinder the large-scale and successful deployment. First, the sensory data provided by participants with diverse reliabilities may not be all trustworthy due to various objective factors (*e.g.*, poor sensor quality and ambient noise) or subjective reasons (*e.g.*, malicious intent), which may degrade the data quality and discourage the involvement of data collectors who request the sensory data. Therefore, from the perspective of system, it is necessary to improve the data trustworthiness/quality. Second, the crowdsensing data are usually tagged with spatial and temporal information about the sensing context and may reveal user's sensitive information (*e.g.*, location and health status [3]). More severely, private information may be inferred if some data are linked. For example, the frequent participation of movie rating-related tasks may disclose the user's interest in a specific type of movies. The continuous observation of sensory data contributed by the same participant may be used to track the user's trajectory. Hence, from the perspective of users, there is an inherent necessity to provide them with a privacy-aware mobile crowdsensing scheme. Third, inevitably, participants need to consume time and their own resources such as battery and computing power for data sensing and upload. Considering these consumptions, a user would be unwilling to join the sensing tasks unless favorable incentives (*e.g.*, rewards) are provided as compensation. Essentially, designing reasonable incentive mechanisms can achieve a win-win situation for both the system and users, in which adequate data are obtained for better availability of the system, and users will earn a number of rewards.

Many research efforts have been devoted to data trustworthiness, privacy preservation, and incentive design for mobile crowdsensing, such as trust/reputation evaluation models designed to evaluate the data trustworthiness ([14], [28]),

Manuscript received June 23, 2019; accepted September 17, 2019; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Ren. Date of publication October 28, 2019; date of current version December 17, 2019. This work was supported in part by the National Science Foundation (NSF) under Grant 1717197, Grant 1717315, and Grant 1525920, in part by the National Natural Science Foundation of China under Grant U1736216 and Grant 61702233, in part by the National Key Research and Development Program under Grant 2017YFB1400703, and in part by the Graduate student innovation projects of Jiangsu province under Grant KYCX17_1810. The information reported here does not reflect the the policy of the funding agencies. The work of H. Wu was supported in part by the Chinese Scholarship Council under Grant 201708320241. *(Corresponding authors: Guoliang Xue; Liangmin Wang.)*

H. Wu and L. Wang are with the Department of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China (e-mail: haiqinwu02@gmail.com; wanglm@ujs.edu.cn).

G. Xue is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: xue@asu.edu).

J. Tang is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: jtang02@syr.edu).

D. Yang is with the Department of Computer Science, Colorado School of Mines, Golden, CO 80401 USA (e-mail:djyang@mines.edu).

Digital Object Identifier 10.1109/TNET.2019.2944984

1063-6692 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

privacy protection solutions proposed for anonymous data collection ([11], [25], [35], [38]), and various incentive schemes ([40], [41], [44]). However, these solutions only address part of these three issues separately, which fail to capture the inner-relationship among these issues and address all of them collectively. Although in recent researches, anonymous reputation systems ([15], [26], [27], [34]) and privacy-aware incentives ([21], [22], [31], [36], [42], [43]) were tentatively studied, they still have many limitations such as the dependence on a trusted third party (TTP), suffering from heavy computation cost, limited to the single-report task scenario, and ignorance of incentive fairness. More importantly, they do not indeed consider these issues in a holistic perspective.

Addressing these above issues simultaneously is a nontrivial challenge, as both conflict and correlation coexist among them. For example, there is a conflict between data trustworthiness and user privacy, as some selfish or malicious users may anonymously report falsified or invalid sensory data. Similarly, potential abuse attacks may occur in privacy-aware incentives, in which malicious users may want to earn more rewards by anonymously submitting redundant reports or stealing others' credentials. Hence, providing privacy protection may lead to anonymity abuse, making it difficult to guarantee the data trustworthiness and track the user's accountability. On the other hand, data trustworthiness has close correlations with incentives. Specifically, users should be rewarded fairly based on their data reliability, and in turn we should stimulate users to contribute more reliable data. However, it is also challenging to quantify user's contribution in a comprehensive and privacy-aware manner, with fair incentive provision.

In this paper, we propose a scheme enabling data trustworthiness and user privacy for general multi-report[1] mobile crowdsensing, in which fair incentives are provided to motivate users' reliable contributions. Compared with our previous work PTISense [39], we further release the strong assumption of a *trusted* pseudonym authority and carefully elaborate an enhanced solution which is resilient against collusion attacks between any two *honest-but-curious* entities under a non-TTP model. Moreover, we also specify the detailed pseudonym generation process and present how to evict malicious users. The main contributions are summarized as follows.

- We present a novel mobile crowdsensing system model by introducing two *honest-but-curious* entities: the group manager and the pseudonym authority, which separate the duty of each entity to ease the server-side burden and serves as the foundation to achieve anonymous user authentication and pseudonym-based data submission.
- Leveraging (partially) blind signature, we propose a trustworthy and privacy-aware scheme by integrating an anonymous reputation model. Particularly, we embed the anonymous reputation level/feedback into tokens to achieve privacy-aware trust evaluation and reputation update at the server and the group manager, respectively. In the entire crowdsensing process, besides anonymous identity verification via group signature, blind signature-based protocols are designed to enable user's anonymous authorization verification at the untrusted pseudonym authority, and to make data unlinkable to individual entity or even possible colluding entities. Based on the data quality and feedback, a fair reward allocation method is devised to stimulate user's reliable participation.
- To prevent malicious users from abusing pseudonyms, we propose a protocol to limit the number of pseudonyms issued to each user and design a certificate-based pseudonym generation method via hash chain. Additionally, we provide two flexible revocation methods to evict malicious users from tasks or the whole system, depending on their specific misbehaviors.

The remainder of this paper is organized as follows. In Section II, we review related work. In Section III, we introduce preliminaries. We then present our proposed scheme in Section IV. The security analysis is presented in Section V and performance evaluations are presented in Section VI. Finally, Section VII concludes this paper.

II. Related Work

A. Data Trustworthiness

To improve the quality of sensory data in mobile crowdsensing, researchers focused on how to evaluate sensory data and maintain the user's reputation. Huang *et al.* [14] proposed a reputation system by employing Gomeprtz function to compute the participant's reputation score. Taking the user privacy requirement into consideration, in [15], each user was assigned multiple pseudonyms and relied on a TTP to transform the reputation among user's multiple pseudonyms. Leveraging the blind signature and cloaking techniques, a similar solution IncogniSense [6] was proposed. Without using a TTP, Wang *et al.* [34] proposed ARTSense to solve the problem of "trust without identity". To further address the issues of long time delay and high bandwidth, Ma *et al.* [26] proposed two privacy-aware reputation management schemes in the edge computing enhanced mobile sensing. Nevertheless, no incentives are provided in these solutions.

Recently, Gisdakis *et al.* [13] proposed a secure and accountable mobile sensing system that preserves the user privacy and provides incentives based on the Shapley value. [2] addressed the conflict between user privacy and data accuracy maximization. They proposed a coalition strategy to allow users to cooperate for k-anonymity protection. Besides additional user-side communication cost, the privacy of a user would be revealed once one of the cooperative users is compromised. We note that the above two schemes all assume the existence of an available evaluation scheme for anonymous users and lack a detailed illustration of how to seamlessly integrate it into the privacy-aware crowdsensing system, especially suitable for the multi-report scenario.

B. Privacy Preservation

Different privacy protection techniques were proposed to protect the user's location privacy, identity privacy, and data privacy. For example, k-anonymity technique [25] was widely used in location-based mobile sensing. For identity privacy, sensing reports were anonymously submitted to the server

[1] The task requires each participant to submit multiple data.

with pseudonyms ([11], [42]) rather than their real identities. Moreover, cryptographic solutions ([31], [38]) were proposed to ensure data confidentiality.

To achieve anonymity and unlinkability, AnonySense [30] was proposed for mobile crowdsensing systems, in which mix network and k-anonymity technology were adopted to anonymize the communication and de-associate the sensing data from their sources. However, this scheme lacks provable privacy guarantees. In [8], an enhanced privacy-aware method PEPSI was proposed to protect the privacy of participants and data consumers. Besides privacy-aware data sensing, several works addressed the privacy-aware task allocation ([35], [38]), data aggregation ([18], [38], [45]), data publishing ([33], [37]) in mobile crowdsensing.

C. Incentive Provision

Incentive mechanisms are designed to encourage user's participation, either with monetary or non-monetary strategies. Auction-based incentives ([40], [41], [44]) have been widely studied in mobile crowdsensing, which focus on how to select participating users (*i.e.*, winners) and determine the payment. Winner selection and payment determination are mostly based on each user's bid information, regardless of the actual data quality. It is not truly fair since the data quality of each winner may not be at the same level. Moreover, users in these schemes are considered *selfish* but *rational*, which is reasonable but neglects the user's malicious intent (*e.g.*, submit low-quality data). For improvements, some quality-aware incentives have been proposed ([2], [16], [20], [32]), which incorporates the data quality into different auction models. Besides auction model, Gao *et al.* [12] formulated a mathematical model to evaluate the data quality which needs to be maximized under a limited task budget. Leveraging Sharply value, [19] proposed a fair reputation-based incentive mechanism to allocate the task budget. Despite its availability, it requires users to form coalitions, hence induces extra communication cost. From the perspective of system, these quality-aware incentives fail to give rigid punishment (*e.g.*, remove them from the system) to the malicious users. None of the above incentives have carefully considered the user's privacy requirements.

Some privacy-aware incentives have been proposed for mobile crowdsensing. One research line focuses on achieving user anonymity when performing tasks and earning rewards. Zhang *et al.* [43] first solved this problem by using pseudonym, cryptography, and hash function. They concentrated on the user's identity privacy while ignoring the data linkage privacy. The same problem also exists in [31]. Li *et al.* [21], [22] have made great contributions to design privacy-aware incentives, in which [22] fits for the single-report tasks while the two schemes proposed in [21] are more general to support both single-report and multi-report tasks. Specifically, the first scheme relies on a TTP while the second solution adopts the blind signature and commitment techniques to preserve privacy. Despite its effectiveness, the TTP-free scheme bears large user-side computation cost. Since the server allocates each user the same number of rewards, data reliability and incentive fairness are not really taken into account. Another research line focuses on protecting user's bid privacy in auction-based incentives ([17], [23], [36]), in which user anonymity, data linkage, and the data trustworthiness are not covered.

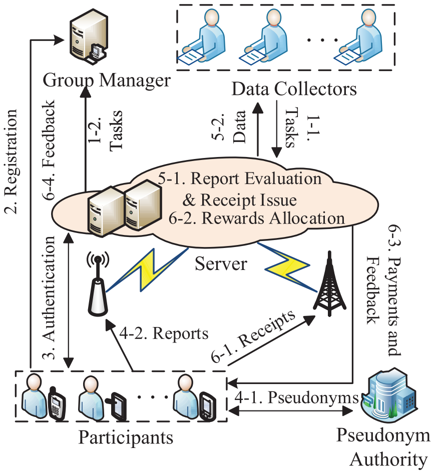


Fig. 1. System architecture. The label 2 represents Step 2, and the label 1-2 represents the second subphase in Step 1. The other labels represent the corresponding steps or subphases.

In this paper, instead of designing auction-based privacy/quality-aware incentives, we follow the research line of enabling user privacy (*e.g.*, identity and data linkage) and data reliability throughout the whole mobile crowdsensing, simultaneously offering a fair budget allocation strategy to incentivize user's high-quality contribution. Particularly, we also aim to provide effective countermeasures to deal with malicious users and possible collusion attacks under a TTP-free model, which solves the critical limitations in [39].

III. Preliminaries

A. System Architecture

The mobile crowdsensing system considered in this paper (refer to Fig. 1 on the following page) consists of a server, a group manager (GM), a pseudonym authority (PA), multiple data collectors (DCs), and a set of participants/users (we use participants and users interchangeably).

1) **Server** publishes tasks received from the DCs and assigns them to a set of qualified users. After collecting the sensory data from the users, the server evaluates the data and allocates rewards to the users according to a certain strategy. Meanwhile, a feedback is returned to the users for later reputation updates.
2) **GM** is in charge of registering participants, generating task request tokens, and managing reputation database.
3) **PA** is responsible for issuing a certain number of valid pseudonyms and corresponding certificates to the authorized participants for the reports and receipts submission.
4) **DCs** create sensing tasks by specifying some requirements, such as the type of sensor reading, the sensing time/location, number of data reports required from each participant, the minimum reputation level, the reward budget, and the task lifetime.

5) **Participants** are mobile users carrying with them sensing-capable mobile devices. They register themselves with the GM and collect sensory data for the requested tasks. All data are then transmitted to the server via cellular networks or WiFi.

From a high-level perspective, the DCs first create sensing tasks and then forward them to the servers. Next, the server groups all received tasks and publishes them to the GM and participants in the vicinity of area of interest (Step 1). Since the privacy and reputation of the DCs are not under our consideration, in this paper, we consider the DCs and the server the same party for simplicity.

To perform a task, a user must register with the GM and obtain a task request token (Step 2). After that, the user submits the task request token and a group signature to the server for anonymous authentication of user identity and task legitimacy. The server will assign the corresponding task to the user if the authentication succeeds (Step 3). Only when assigned, the user can request a certain number of pseudonyms from the PA, then anonymously submits the reports to the server (Step 4). For each report, the server evaluates its reliability and issues a receipt to the user, meanwhile the valid data reports are returned to the DCs (Step 5). Finally, the user can submit all his/her collected receipts to the server for rewards redemption before a deadline. Meanwhile, the user also obtains a reputation update token from the server, which is used for reputation update at the GM (Step 6).

B. Threat Model

Threats to Trustworthiness. Unauthenticated users may submit forged data to the server. For legitimate users, they may also exhibit malicious behaviors, including submitting false data intentionally or submitting low-quality/inaccurate data due to the malfunction or low quality of sensors. Both false data and very low-quality data (*e.g.*, below a trust threshold) are considered invalid. Furthermore, multiple users may collusively send false data to destroy the crowdsensing applications. As in [21], [34], we consider the majority of the data valid and accurate. The robustness of our scheme against malicious users will be analyzed in Section VI-B.2.

Threats to Privacy. When requesting a task credential (*i.e.*, task request token), the curious GM may want to know which tasks the user is interested in. During the pseudonym request phase, the PA may infer the relationship between the real identity of a user and his/her joined tasks. Moreover, when reporting data, the server may be curious about which tasks the user has performed and which reports are submitted. In a nutshell, the GM, the server, and the PA may try to infer the correlations between users and their private information.

Threats to Incentives. Leveraging the privacy preservation, greedy users may abuse anonymity and try to earn more rewards by submitting redundant reports for each task. Selfish users may want to earn rewards from a task without contributing any data or submitting insufficient reports as required. In addition, some malicious users may try to use tokens of different tasks interchangeably, or use a token twice (double-spending problem). More severely, an adversary may usurp others' tokens by compromising them.

In this paper, we make the following assumptions.

- We assume that the communications between users and the server are anonymized by Mix Networks [7] or Onion Routing (unidentified in the network layer).
- We assume that the GM, the server, and the PA are all "*honest-but-curious*" (*i.e.*, semi-honest), which means that they will conform to the designated protocols, but are curious and may try to infer more privacy of the users. In reality, collusions may exist but would degrade the reputation of the collusive entities. In this paper, we assume that there may be collusions between any two of the semi-honest entities, whereas, the three entities would not collude due to the reputation concerns.

C. Design Goals

The following are our design requirements:

R1 Data Trustworthiness. To resist against forged data from the external attackers, all participants should be authenticated before task assignment. In addition, it is necessary to establish anonymous reputation assessment and management schemes to mitigate data trustworthiness threats from internal attackers. Specifically, tasks requested by low-reputation participants are supposed to be rejected by the server. Reports with very low trust values should be detected and removed. Moreover, it is infeasible for the greedy users to submit redundant reports.

R2 Privacy Protection: Given a user, the adversaries cannot infer if the user has requested/accepted a specific task, or whether two tasks have been performed by the same user. When reporting data, the sensing reports should also be unlinkable, *i.e.*, adversaries cannot link any report to its contributor, and link multiple reports submitted by the same user. For reward redemption, the linkage between rewards a user earns and the data submitted by the user should be unknown.

R3 Fairness: Users should be rewarded fairly according to a certain reward allocation strategy in a privacy-preserving manner. Misbehaving users cannot increase their rewards by abusing pseudonyms, double-spending, or stealing tokens. Those having made no contributions, submitting invalid data, or even not assigned tasks will earn nothing.

D. Primitives

Group Signature [4]. A group signature scheme is a cryptographic primitive for anonymous identity authentications. Generally, this scheme consists of five algorithms. The key generation algorithm $\mathsf{KeyGen}()$ outputs a group public key vk and a group secret key gsk. If a user P_i wants to join the group, he/she will perform a Join protocol with the GM. Then P_i obtains a member secret key msk_i, and the GM obtains some relevant information Y_i from P_i which will be included into gsk. To sign a message m, P_i runs $\mathsf{Sign}(msk_i, m)$ and obtains σ. Anyone obtaining vk can verify σ by performing $\mathsf{Verify}(vk, m, \sigma)$. If necessary, the GM can identify and trace the signer with $\mathsf{Open}(gsk, m, \sigma)$. Finally, using algorithm $\mathsf{Revoke}(gsk, Y_i)$, the member with Y_i can be evicted from the group. Group signature has two properties: anonymity and traceability, which captures our security requirements.

TABLE I
NOTATION SETTINGS

Notations	Description
N	Number of participants in the system
η	Number of malicious participants in the system
M	Number of tasks in each task group
n_i	Number of reports required from each user for task T_i
B_i	Total reward budget paid for a task T_i
vk, gsk	Group public and secret keys
msk_i	Member secret key of participant P_i
pk_x, sk_x	Key-pair (public and private keys) of entity x
H	A cryptographic hash function
τ	Task request token
α	Receipt identifier
$[m]_{pk_s}$	Ciphertext of message m encrypted by pk_s
$\{m\}_{sk_s}$	Signature of message m signed by sk_s

Blind Signature and Partially Blind Signature. Blind signature (BS) [5] enables a user to obtain a signature from a signer on his/her private message m without disclosing m to the signer. Take the blind RSA signature as an example (used in this paper), suppose that the public-private keys of the signer are (e, Q) and (d, Q), where Q is the public modulo. The user first chooses a blinding factor b relatively prime to Q, and computes $m' = m \cdot b^e mod\ Q$. The signer signs on m' with d. With the signature $\{m'\}_d$, the user can obtain the real signature on m by removing b as $\{m\}_d = (\{m'\}_d \cdot b^{-1}) mod\ Q$. Besides blindness and unlinkability, the user cannot forge a valid signature from $\{m'\}_d$ for another message. Partially blind signature (PBS) [1] is similar to BS except for some common/public information added in the signature.

Trust and Reputation. As in [34], we quantify the reliabilities of reports and users with *trust* and *reputation*, respectively. Particularly, *reputation level* is introduced for anonymous trust evaluation, which is a discrete approximation deduced from a user's reputation. Users with different reputations may demonstrate the same reputation level, hence the server cannot differentiate users based on their reputation levels.

In Section IV, we will show how to adopt group signature and (partially) BS for privacy-aware authentication, data submission, and reward allocation. Moreover, trust assessment and reputation update are to be conducted based on the anonymous sensing reports. For ease of presentation, the notations used in this paper are listed in Table I.

IV. THE EPTSENSE SCHEME

In this section, we present our scheme EPTSense, an **E**nhanced scheme achieving the goals on "**P**rivacy Preservation" and "Data **T**rustworthiness"[2] for mobile crowd-**S**ensing. Compared with our prior scheme PTISense [39], the key issues to be addressed are three-folded. The first is how to deal with the possible collusion attacks between any two *honest-but-curious* entities. The second is to loose the strong assumption about a *trusted* PA. The last is to specify a pseudonym generation method with lightweight cost at the PA. Note that, we use "enhanced" to emphasize the security improvement instead of the efficiency. Before describing EPTSense, we first give an overview and the basic idea of our approach.

A. Overview

Our entire scheme consists of seven required phases and one alternative phase. First, the system is initialized with key distribution (Section IV-B). Then each user registers with the GM and gets the member secret key and task request token (Section IV-C). Before allocating tasks, the server anonymously verifies the task request token and returns approval or rejection messages to the user (Section IV-D). Only users who receive the approval message are able to request a certain number of pseudonyms from the PA, with which the sensing reports can be submitted anonymously (Section IV-E). For each report received, the server will evaluate its trust and issue a feedback-embedded receipt (Section IV-F). After collecting all receipts, users are able to redeem rewards from the server in a fair way. Meanwhile, a reputation update token is also returned to the user for reputation update at the GM (Section IV-G). During the whole process, if the reputation of a user or the trust value of a submitted report does not satisfy the minimum requirement, the corresponding user will be removed from the system or the reports submitted by the same user will be revealed and tagged as invalid (Section IV-H).

Particularly, to tackle the possible collusions between the GM and the server in PTISense, leveraging the BS technology, we propose privacy-aware protocols to let the user and the GM (server, respectively) both involved in creating the task request (reputation update, respectively) tokens, instead of allowing the GM or the server to unilaterally sign on the identity-related information. To protect the privacy from the semi-honest PA, our basic idea is to enable the PA to anonymously verify the task authorization of a requested user, meanwhile not sacrificing the task privacy. Moreover, we devise a efficient pseudonym generation method based on the hash chain, which lowers the PA's pseudonym storage cost to a constant.

B. Initialization

In this phase, a certificate authority first delivers a public-private key pair to the server, the GM, and the PA, respectively. The GM performs KeyGen() to generate a pair of group public-private keys (vk, gsk).

The server is responsible for grouping all tasks received from the DCs (*e.g.*, indexed $1, 2, \ldots, M$ in the order of their reception time), and publishes these tasks to the mobile users and the GM. To improve the data trustworthiness, the server also presets two parameters: a reputation threshold $\epsilon \in [0, 1]$ and a trust threshold $\delta \in [0, 1]$, below which the user and the data report are considered malicious and invalid, respectively.

C. Participant Registration

If a participant P_i wants to join a task for the first time, he/she must register with the GM and obtain msk_i through an interactive Join protocol [4] supporting dynamic update.

[2] In essence, the goal of achieving *fair incentives* is to stimulate user's reliable and high-quality contribution, which also serves for the data trustworthiness. Therefore, we omit the *incentive fairness* in the description.

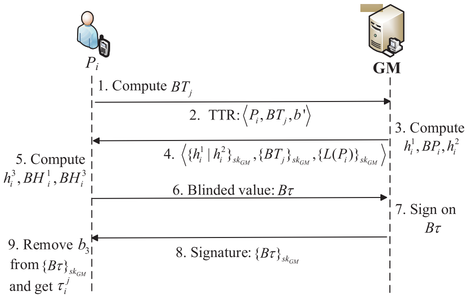

P_i
GM
1. Compute BT_j
2. TTR: $\langle P_i, BT_j, b' \rangle$
3. Compute h_i^1, BP_i, h_i^2
4. $\langle \{h_i^1 | h_i^2\}_{sk_{GM}}, \{BT_j\}_{sk_{GM}}, \{L(P_i)\}_{sk_{GM}} \rangle$
5. Compute h_i^3, BH_i^1, BH_i^3
6. Blinded value: $B\tau$
7. Sign on $B\tau$
8. Signature: $\{B\tau\}_{sk_{GM}}$
9. Remove b_3 from $\{B\tau\}_{sk_{GM}}$ and get τ_i^j

Fig. 2. The interactive process for request token generation.

To join a task T_j, P_i needs to send some private information and acquire a task request token from the GM. As shown in Fig. 2 (on the following page), first, to protect the user's task preferences, the requested task identifier (*i.e.*, ID) is blinded using the BS. Specifically, P_i chooses a nonce b relatively prime to the GM's public modulus Q. Then he/she computes the blinded task $BT_j = T_j \cdot b^{pk_{GM}} mod\ Q$. Meanwhile, P_i chooses another blinding factor b' which is shared with the GM to construct a blind commitment h_i^2 (explained later). In a Task Token Request (TTR), P_i sends $\langle BT_j, b' \rangle$ to the GM with his/her real identity.

The GM maintains a reputation table for each member with an initial reputation. After receiving the TTR from P_i, the GM first derives a hash value $h_i^1 = H(P_i|R(P_i)|BT_j)$ by binding the user's identity P_i, his/her newest reputation $R(P_i)$, and the blinded task ID BT_j. Then the GM computes the user's blind identity $BP_i = P_i \cdot b'^{pk_{GM}}\ mod\ Q$ using b', and derives another hash value h_i^2 by binding BP_j and BT_j (*i.e.*, $h_i^2 = H(BP_i|BT_j)$), so that the PA can later anonymously validate that P_i is really assigned task T_j. Finally, the GM returns $\langle \{h_i^1|h_i^2\}_{sk_{GM}}, \{BT_j\}_{sk_{GM}}, \{L(P_i)\}_{sk_{GM}} \rangle$ to P_i, in which $L(P_i)$ is the reputation level of P_i.

On the user side, P_i extracts h_i^1, h_i^2 and removes b from $\{BT\}_{sk_{GM}}$ to obtain $\{T_j\}_{sk_{GM}}$, then he/she computes $h_i^3 = H(h_i^2|1)$ and chooses two blinding factors b_1, b_2 to blind h_i^1 and h_i^3, respectively (the blinded values are denoted as BH_i^1 and BH_i^3). After that, P_i computes $BH_i^1|BH_i^3|\{T_j\}_{sk_{GM}}|L(P_i)_{sk_{GM}}$ which is blinded using another blinding factor b_3 (the blinded value is denoted by $B\tau$). Through a round of interaction with the GM, P_i removes b_3 and is able to obtain the task request token for T_j as follows.

$$\tau_i^j = \{BH_i^1|BH_i^3|\{T_j\}_{sk_{GM}}|L(P_i)_{sk_{GM}}\}_{sk_{GM}}. \tag{1}$$

Note that, $L(P_i)$ is included in the token so that P_i can demonstrate his/her reliability to the server without any linkage to his/her identity. On the other hand, the BS technique applied to the token generation ensures that the GM is oblivious to the task request token issued to a known user. In this case, even if the request token is exposed to the server who may collude with the GM, both entities still cannot infer the relationship between a user and his/her requested tasks.

D. Participant Authentication and Task Assignment

With the desired task request token τ_i^j and msk_i, P_i can send a task request and authenticate himself/herself to the server anonymously. Specifically, P_i chooses a random number r and generates a group signature on r (as the message) with his/her member secret key msk_i, which is sent along with r, τ_i^j, and T_j. Let p_i^0 denote a random pseudonym[3] generated by P_i, then the anonymous task request message $\Re_i$ is as follows:

$$P_i \rightarrow server : \Re_i = \langle p_i^0, T_j, \{r\}_{msk_i}, r, \tau_i^j \rangle. \tag{2}$$

Upon receiving $\Re_i$, based on r and vk, the server can verify the group signature anonymously. If it succeeds, P_i is considered a legitimate member. To further verify τ_i^j, the server performs the following steps:

1) It verifies the authenticity of τ_i^j by checking the signature of the GM with pk_{GM}. This ensures that no one can forge a valid request token.
2) It verifies $\{T_j\}_{sk_{GM}}$ and ensures that τ_i^j is indeed issued for the task.

If all steps succeed and τ_i^j has not been used before, τ_i^j is considered authentic and correct. Next, the server extracts $L(P_i)$ from τ_i^j and determines whether to approve the request according to the task requirements. If approved, τ_i^j is stored and tagged as *approved* to prevent double-spending problem. Additionally, the server returns an approval message A_i as

$$server \rightarrow P_i :$$
$$A_i = \langle \{BH_i^1\}_{sk_s}, \{BH_i^3, n_j + 1\}_{sk_s}, \{T_j|L(P_i)\}_{sk_s} \rangle, \tag{3}$$

where $\{BH_i^3, n_j + 1\}_{sk_s}$ is the server's PBS and $n_j + 1$ is the common knowledge known by the user and the server, $\{T_j|L(P_i)\}_{sk_s}$ is regarded as an *anonymous reputation certificate* (ARC) [34], demonstrating P_i's reputation level when performing T_j. After receiving A_i, P_i removes b_1, b_3 and gets $\{h_i^1\}_{sk_s}, \{h_i^3, n_j + 1\}_{sk_s}$ (for task authorization verification).

On the contrary, if the server rejects the request, a rejection message $\{BH_i^1, 0\}_{sk_s}$ is returned to P_i, in which b_1 can be removed to derive $\{h_i^1, 0\}_{sk_s}$ as the request feedback (to be described in Section IV-G).

E. Pseudonym Request and Report Submission

Before report submission, P_i needs to get $n_j + 1$ pseudonyms and the corresponding certificates from the PA, where n_j is the number of reports required from each user. Specifically, P_i first generates $n_j + 1$ public-private key pairs $\{(pk_i^k, sk_i^k)\}_{k=1}^{n_j+1}$. To protect the identity and task privacy, P_i uses the blinded identity BP_i and the blinded task ID BT_j to request pseudonyms. Additionally, to enable the PA to generate certificates based on the task lifetime without revealing the specific task ID, P_i also selects a start time t_s and an end time t_e that are very close to the start and end time of T_j, respectively. Finally, a pseudonym request is sent to the PA in the form $\langle BP_i, t_s, t_e, n_j + 1, \{pk_i^k\}_{k=1}^{n_j+1}, BT_j, \{h_i^3, n_j + 1\}_{sk_s} \rangle$.

Upon reception, the PA first verifies the PBS $\{h_i^3, n_j + 1\}_{sk_s}$, then it verifies if P_i is really assigned T_j by checking $H(H(BP_i|BT_j)|1) \overset{?}{=} h_i^3$. If both hold, the PA prepares $n_j + 1$ pseudonyms and certificates, as shown in Algorithm 1.

[3] Here the pseudonym is not generated by TPA for extra communication cost. Anyone can request tasks with a random pseudonym.

Algorithm 1 Pseudonym and Certificate Generation

```
Input: $BP_i$, $t_s$, $t_e$, $n_j+1$, $T_j$, $\{pk_i^k\}_{k=1}^{n_j+1}$.
Output: Pseudonyms $\{p_i^k\}_{k=1}^{n_j+1}$ and certificates
        $\{c_i^k\}_{k=1}^{n_j+1}$.
1  // PA;
2  Choose two secrets $s_i^1, s_i^2$;
3  $\Delta t = \frac{t_e - t_s}{n_j}$;
4  for $k = 1$ to $n_j$ do
5      $S_{(i,1)}^k = H^k(s_i^1)$, $S_{(i,2)}^k = H^k(s_i^2)$;
6      $p_i^k = H(S_{(i,1)}^k|S_{(i,2)}^k|k)$;
7      $t_k = t_c + \Delta t \cdot k$;
8      $c_i^k = \{p_i^k|pk_i^k|t_k|T_j\}_{sk_{PA}}$;
9  $p_i^{n_j+1} = H(S_{(i,1)}^{n_j+1}|S_{(i,2)}^{n_j+1}|n_j+1)$;
10 $t_{n_j+1} = t_{n_j} + \Delta t$;
11 $c_i^{n_j+1} = \{p_i^{n_j+1}|pk_i^{n_j+1}|t_{n_j+1}|T_j\}_{sk_{PA}}$;
12 Keep secret record $\langle BP_i, s_i^1, s_i^2 \rangle$;
```

First, the PA chooses two secrets s_i^1, s_i^2, and equally divides the appropriate lifetime of T_j into n_j time intervals (Lines 2-3, each interval is Δt). Note that time division is to guarantee that each pseudonym/certificate is only valid in a time interval such that malicious users cannot launch Sybil attack and use expired pseudonym/certificate. For the k-th pseudonym, the PA derives its secrets using hash chain (Line 5). With hashed secrets, the k-th pseudonym of P_i (*i.e.*, p_i^k) is constructed as $p_i^k = H(S_{(i,1)}^k|S_{(i,2)}^k|k)$. After that, the PA generates the corresponding certificate by signature (pk_i^k is embedded in the certificate for data integrity verification). For the (n_j+1)-th pseudonym $p_i^{n_j+1}$, it is generated in a similar way and is used to submit receipts after task completion. Finally, the PA only needs to store one information record as $\langle BP_i, s_i^1, s_i^2 \rangle$, regardless of n_j. Therefore, EPTSense significantly saves the PA's storage cost, as PTISense needs to keep n_j+1 pseudonyms for each user.

In our scheme, only with valid pseudonyms/certificates issued by the PA, P_i can anonymously and successfully submit the sensing reports for T_j. Otherwise, P_i is regarded unauthorized if he/she uses randomly generated pseudonyms. Specifically, after receiving n_j+1 pseudonyms and certificates, P_i submits each report $\mathbb{R}_k (k = 1, 2, \ldots, n_j)$ to the server as

$$P_i \rightarrow server: \mathbb{R}_k = \langle p_i^k, c_i^k, T_j, \{T_j|L(P_i)\}_{sk_s}, D_i^k, \sigma_{sk_i^k} \rangle, \quad (4)$$

where D_i^k is the kth set of sensing data items, $\sigma_{sk_i^k}$ is P_i's signature on all the remaining report items and the corresponding verification public key is embedded in c_i^k. Besides data integrity, the signature also ensures that the pseudonym and certificate are in accord with the signer. Moreover, the ARC is necessarily included to assess the report's trust.

F. Trust Evaluation and Receipt Generation

For each sensing report $\mathbb{R}_k$ received, the server performs:

1) It checks the validity of p_i^k by verifying c_i^k and comparing p_i^k and T_j with those included in c_i^k. Moreover, the server checks if the current time is less than t_k to ensure the pseudonym validity.
2) It validates that the ARC has been signed by itself and the task ID included is consistent with T_j.

If both checks are passed, the server continues to assess the trust of each sensing report. In this paper, we adopt the trust assessment model in [34] which evaluates the sensory data in a comprehensive aspects, including the anonymous reputation level $L(P_i)$, some privacy-aware contextual factors (*e.g.*, time and location), and the similarity of data from multiple users. First, based on $L(P_i)$ and the contextual factors, the basic trust of each report $\mathbb{R}_k$ is computed. Then, the final trust, denoted by $T_F(\mathbb{R}_k)$, is derived by combining with the data similarity. Instead of using the single server only [34], we evaluate the report trust and user's reputation at the server and the GM, respectively. The advantage is that it eases the server-side heavy burden and reduces the user's information known by a single party. Additionally, in our multi-report scenario, reports submitted to T_j are further divided into n_j sets before data similarity computation. Due to space limitation, we omit the computation details and refer readers to [34] for detailed data similarity computation and trust evaluation.

After deriving the final trust, the server compares $T_F(\mathbb{R}_k)$ with the predefined trust threshold δ, if $T_F(\mathbb{R}_k) \geq \delta$, the server accepts it and computes a feedback level $l_f(\mathbb{R}_k)$ based on $T_F(\mathbb{R}_k) - L(P_i)$. The key principle is that a positive feedback level is set if $T_F(\mathbb{R}_k) \geq L(P_i)$ and a negative feedback is set otherwise. Additionally, we set the same feedback level for two reports with similar gaps, such that the server cannot link $l_f(\mathbb{R}_k)$ to the report when later submitting receipts. If $T_F(\mathbb{R}_k) < \delta$, $\mathbb{R}_k$ is considered invalid and will be rejected.

For each report $\mathbb{R}_k$ accepted, the server issues a receipt $R_{\mathbb{R}_k}$ to P_i, which can be later used by P_i to redeem rewards from the server. Particularly, to achieve the distinguishability and unlinkability of receipts, we adopt the PBS technology, in which the task ID is the common information shared by the user and the server. Different from PTISense, we do not embed the identity-relevant information (*i.e.*, h_i^1) into the receipt identifier (*i.e.*, ID) in this step, in case of possible privacy disclosure from h_i^1 when collusion exists. Instead, P_i chooses a nonce ρ and computes $\alpha_k = H(\rho|T_j|k)$, $k = 1, \ldots, n_j$ as the receipt ID, then obtains the PBS $\{\alpha_k, T_j\}_{sk_s}$ from the server. Meanwhile, the server sends $\{\{T_j|L(P_i)\}_{sk_s}|[l_f(\mathbb{R}_k)]_{pk_s}\}_{sk_s}$ to P_i. Based on this, the receipt $R_{\mathbb{R}_k}$ is as follows:

$$\begin{aligned} &server \rightarrow P_i: \\ &R_{\mathbb{R}_k} = \langle T_j, \{\alpha_k, T_j\}_{sk_s}, \{\{T_j|L(P_i)\}_{sk_s}|[l_f(\mathbb{R}_k)]_{pk_s}\}_{sk_s} \rangle. \end{aligned} \quad (5)$$

Note that the feedback level $l_f(\mathbb{R}_k)$ is encrypted with pk_s so that P_i cannot identify if it is negative or positive.

G. Participant Remuneration and Reputation Update

Upon reaching the end time of T_j (denoted as t_e'), each user can submit all receipts he/she obtained for reward redemption before a deadline. In this paper, we set the time interval for receipt submission to be the same as that between two consecutive report submissions. Specifically, P_i sends the following message with pseudonym $p_i^{n_j+1}$ before time $t'_e+\Delta t$.

$$\begin{aligned} &P_i \rightarrow server: \\ &\quad \langle p_i^{n_j+1}, c_i^{n_j+1}, \rho, (\alpha_k, R_{\mathbb{R}_k})_{k=1,\ldots,n_j}, \sigma_{sk_i^{n_j+1}} \rangle. \end{aligned} \tag{6}$$

Subsequently, the server does some verifications:

1) It verifies the validity of $p_i^{n_j+1}$ and $\{\alpha_k, T_j\}_{sk_s}$, which ensures that P_i is authorized and has submitted n_j reports for T_j.
2) It checks each $\alpha_k = H(\rho|T_j|k)$ to ensure the receipt is really issued to P_i. Anyone who steals other's receipts (without ρ) cannot pass the verification.

If both checks succeed, the server stores and invalidates α_k to avoid receipt reuse. Then, it decrypts $[l_f(\mathbb{R}_k)]_{pk_s} (k = 1, 2, \ldots, n_j)$ and gets $l_f(\mathbb{R}_k)$, based on which the average feedback value can be computed as $\overline{l_f} = \sum_{k=1}^{n_j} l_f(\mathbb{R}_k)/n_j$. Furthermore, with $L(P_i)$, the approximate average final trust of reports can be obtained by calculating $\overline{T_F} = \overline{l_f} + L(P_i)$.

For incentive fairness, in the premise of the same number of contributions, the basic principle is that users with higher trust values should earn more rewards than those with lower trust values. Moreover, it is more suitable to reward the positive-feedback and negative-feedback users with different allocation strategies. Let P be the set of all authorized users, and S_P, S_N be the set of users with positive and negative average feedback, respectively. Given the task budget B_j, the reward distributed to each user P_i is computed as follows:

$$r_i = \begin{cases} \dfrac{\overline{T_F}(P_i, T_j)}{\sum_{P_k \in P} \overline{T_F}(P_k, T_j)} \cdot B_j \cdot e^{\overline{l_f}(P_i, T_j) \cdot \psi}, & P_i \in S_N \\ \dfrac{\overline{T_F}(P_i, T_j)}{\sum_{P_k \in P} \overline{T_F}(P_k, T_j)} \cdot B_j, & P_i \in S_P \end{cases} \tag{7}$$

where ψ ($\psi > 1$) is an amplification factor to increase the effect of the negative feedback on the reward allocation. Essentially, the negative-feedback users would obtain fewer rewards than their real contributions (as punishment). As $\overline{T_F}(P_i, T_j)$ decreases, the reward reduces accordingly, indicating that users with higher negative feedback will get lower payments.

Note that, submitting insufficient receipts inadvertently indicates the existence of invalid reports with a high possibility, as no receipts are issued for invalid reports. In this case, the server computes the maximum reputation feedback level as $\delta - L(P_i)$ for each unreceived receipt. Intuitively, the addition of $\delta - L(P_i)$ for unreceived receipts would lower the average feedback of received receipts. If the average feedback of n_j receipts is positive, then P_i is considered a positive-feedback user. Otherwise, the server tags P_i with negative feedback. In both cases, the reward allocation also follows Eq. (7). For selfish users or malicious users who intentionally submit partial receipts with possible high feedback levels, those unreceived receipts are still assigned the same feedback level (negative with a high possibility) as we describe above. Hence, no matter behaving inadvertently or intentionally, users cannot benefit from submitting insufficient receipts.

Besides the rewards, the server also returns P_i a reputation update token U_{T_j} for T_j. Unlike PTISense which allows the server to link the update token with the user's identity-related information h_i^1 (the user-task relationship would be disclosed if collusion exists), we generate U_{T_j} in a privacy-aware manner. Specifically, the server first sends a message $\langle T_j, [\overline{l_f}]_{pk_{GM}}, \{[\overline{l_f}]_{pk_{GM}}\}_{sk_s} \rangle$ to P_i. To embed the identity-related information into U_{T_j} while keeping the token unlinkable, P_i computes $H(h_i^1|BT_j|[\overline{l_f}]_{pk_{GM}})$ and uses a blinding factor to hide its real value. Through another round of interaction with the server, P_i removes the blind factor and is able to obtain the reputation update token U_{T_j} as follows.

$$U_{T_j} = \{H(h_i^1|BT_j|[\overline{l_f}]_{pk_{GM}})\}_{sk_s}. \tag{8}$$

Note that, given U_{T_j}, the server is oblivious to which task/receipts the token issued for. Therefore, even though there is collusion between the GM and the server, P_i's identity-task relationship is still unrevealed.

No matter whether P_i is approved to perform tasks in the assignment phase, P_i needs to return feedback information to the GM for reputation update as long as he/she requests tasks. Specifically, if assigned, P_i submits $\langle U_{T_j}$, BT_j, $[\overline{l_f}]_{pk_{GM}}$, $\{[\overline{l_f}]_{pk_{GM}}\}_{sk_s} \rangle$ to the GM with the real identity. The GM verifies the server's signature and the authenticity of $[\overline{l_f}]_{pk_{GM}}$. Next, it computes $H(H(P_i| R(P_i)|BT_j)|BT_j|[\overline{l_f}]_{pk_{GM}})$ by retrieving $R(P_i)$. If the derived result equals $H(h_i^1|BT_j|[\overline{l_f}]_{pk_{GM}})$ in U_{T_j}, U_{T_j} is considered valid. After successful verification, the GM decrypts $[\overline{l_f}]_{pk_{GM}}$ and updates P_i's reputation. Moreover, it stores U_{T_j} and tags it as *used* to prevent token reuse.

On the contrary, if P_i is not authorized to perform T_j, he/she still needs to return a request feedback F_i to the GM.

$$P_i \rightarrow GM: F_i = \langle P_i, BT_j, \{h_i^1, 0\}_{sk_s} \rangle. \tag{9}$$

After receiving F_i, the GM verifies the signature $\{h_i^1, 0\}_{sk_s}$ and checks if the task request is indeed rejected by comparing $H(H(P_i|R(P_i)|BT_j))$ with h_i^1. In this case, the malicious participant cannot act as a new user (*i.e.*, with the initial reputation) once he/she has been assigned a task.

H. Participant Eviction

To deal with malicious users who may submit low-quality or even invalid reports to degrade the data trustworthiness, our scheme provides two effective countermeasures to evict participants from the specific tasks or the system.

Participants with Invalid Data. Some malicious participants may occasionally submit very low-quality data (lower than δ) for certain tasks due to their personal skill limitations. As mentioned in Section IV-F, these very low-quality data are considered invalid. To deal with these participants, we propose to evict them from the task once a report is detected invalid. In other words, the following data submitted for the task by the same participant is also considered invalid as long as a report is first evaluated as invalid. Take P_i who participates in task T_j as an example, P_i submits a data report with pseudonym p_i^k in T_j's time interval t_k, if the report is detected invalid, the server will send p_i^k and t_k to the PA. Accordingly, the PA scans each stored secret record $\langle BP_j, s_j^1, s_j^2 \rangle$) and compares the derived k-th pseudonym $H(S_{(j,1)}^k|S_{(j,2)}^k|k)$ with

Algorithm 2 Revocation of Participants From a Task

```
Input: $p_i^k, t_k, \langle P_j, s_j^1, s_j^2 \rangle$, $j = 1, 2, \ldots N$.
Output: A set of pseudonyms $\{p_i^l\}_{l=k+1}^{n_j}$ to be blacklisted.
1  // PA;
2  for $j = 1$ to $N$ do
3      $S_{(j,1)}^k \leftarrow H^k(s_j^1)$;
4      $S_{(j,2)}^k \leftarrow H^k(s_j^2)$;
5      if $H(S_{(j,1)}^k | S_{(j,2)}^k | k) == p_i^k$ then
6          Send $S_{(j,1)}^k$ and $S_{(j,2)}^k$ to the server;
7      else
8          Report $p_i^k$ is invalid to the server;
9  // Server;
10 for $l = k+1$ to $n_j$ do
11     Compute $H(S_{(j,1)}^{l-k})$, $H(S_{(j,2)}^{l-k})$;
12     $p_j^l = H(H(S_{(j,1)}^{l-k}) | H(S_{(j,2)}^{l-k}) | l))$;
13     Blacklist $p_j^l$;
```

p_i^k, as shown in Algorithm 2. If they are equal (*i.e.*, $j = i$), the PA sends $S_{(j,1)}^k$ and $S_{(j,2)}^k$ to the server. Next, the server derives the following pseudonyms $\{p_j^l\}_{l=k+1}^{n_j}$ to be used by P_i after t_k (Lines 10-13, Algorithm 2), which will be added to the blacklist and the reports submitted with these pseudonyms are considered invalid with no need of trust evaluation. In this case, it is observed that P_i is essentially evicted from T_j after t_k, since he/she cannot receive any receipt regarding T_j. Compared to eviction from the system, the real identity of P_i keeps hidden from the GM and the server, and even from the PA. Hence, even though the server colludes with the PA, they only know that a set of data are from the same misbehaving user but not knowing the real identity.

Participants with Very Low-Reputation. When performing tasks, if P_i frequently submits valid but low-quality data (*i.e.*, the data trust is higher than δ but may be lower than the average trust value), or frequently/occasionally submits invalid data (*i.e.*, the data trust is lower than δ), the reputation of P_i may be lower than the predefined threshold ϵ at some time. In these cases, we consider to evict P_i from the system. Specifically, after successful authentication and verification of task request token, the server extracts $L(P_i)$ and checks if $L(P_i) < \epsilon$. If it satisfies the condition, the server will deliver P_i's group signature $\{r\}_{msk_i}$ to the GM who can open the signature with gsk and reveal the identity of P_i. P_i will be added to the blacklist, and he/she cannot obtain any task request token from the GM, hence fails to take any task.

V. Security Analysis

In this section, we will show that EPTSense can achieve our defined requirements $R1 - R3$ on data trustworthiness, privacy, and incentive fairness.

A. Data Trustworthiness

Requirement 1: Unauthorized users cannot forge and intercept other's pseudonyms to submit falsified or redundant reports without being detected. Users are only able to honestly submit the reputation update information to the GM. Moreover, misbehaving users will be evicted from a specific task or the whole system, with their identities unrevealed or revealed.

Analysis. If a participant P_i is not authorized to join task T_j, he/she cannot obtain the approval message A_i, hence fails to request pseudonyms from the PA. Even if P_i intercepts an approval message which is sent to another authorized participant P_i', he/she cannot pass the verification of the PBS $\{h_{i'}^3, n_j + 1\}_{sk_s}$ and $H(H(BP_i|BT_j)|1) = h_{i'}^3$. Therefore, the user cannot obtain the valid pseudonyms. Since each pseudonym is signed by the PA, malicious users cannot forge a valid pseudonym, which ensures that all sensing reports received by the server are from the authorized users.

During the reputation update phase, if P_i is authorized, he/she must submit $\langle U_{T_j}, BT_j, [\overline{l_f}]_{pk_{GM}}, \{[\overline{l_f}]_{pk_{GM}}\}_{sk_s} \rangle$ to the GM, in which $[\overline{l_f}]_{pk_{GM}}$ cannot be modified by P_i, otherwise the signature verification of $\{[\overline{l_f}]_{pk_{GM}}\}_{sk_s}$ will not be passed. If P_i uses the encrypted average feedback which corresponds to another task or another user, he/she may be updated with a lower reputation as P_i is oblivious to the exact value of $\overline{l_f}$. Even if P_i steals other's entire reputation update certificate including the reputation update token, the encrypted feedback, and the signature, his/her reputation cannot be successfully updated due to the hash check in U_{T_j}. In other words, P_i can only use his/her own reputation update token to update the reputation at the GM after a specific task. On the other hand, the verification of the PBS $\{h_i^1, 0\}_{sk_s}$ prevents P_i from not submitting the reputation update information and always holding the initial reputation when requesting tasks.

As described in Section IV-H, with the cooperation of the GM, misbehaving users can be identified and removed from the system if their reputation is less than ϵ. Additionally, with the cooperation of the PA, users with invalid data will be detected and removed from the corresponding task by the server. In our scheme, all pseudonyms would be revealed to the server once the user maliciously submits a report whose trust is lower than δ. However, the real identities of these malicious users keep unknown to the server and the PA. Even though the server maliciously requests all pseudonyms of a certain user, it cannot infer the real identity of these pseudonyms and cannot correlate the same user joining different tasks. Overall, our two eviction methods ensure that all malicious users and low-quality reports are detected and removed.

B. Privacy Preservation

Requirement 2: The adversaries can neither link a user's identity to his/her requested tasks (or submitted reports) nor link multiple tasks (or reports) requested (or contributed) by the same user, as long as the user is well behaved and the GM, the server, and the PA do not collude simultaneously.

Analysis: In the registration phase, the user's real identity is included in the TTR, but the requested task ID is blinded with b. Given two different tasks, the GM cannot identify if the two tasks are requested by the same user. In the task assignment phase, the user can get authenticated anonymously via group signature. Although the task ID is disclosed to

TABLE II
COMPUTATION, COMMUNICATION AND STORAGE COST FOR EACH USER PER TASK

	Schemes	Comp.	Comm.	Storage
Participant	PTISense	nH+(2n+2)M.E.+(n+1)M.M.+GS/SIG	$O(n)$	$O(n)$
	EPTSense	(n+2)H+(2n+12)M.E.+(n+11)M.M.+GS/SIG		
Server	PTISense	(n+1)H+(7n+10)M.E.+M.M.+GS/VER	$O(n)$	$O(n)$
	EPTSense	nH+(7n+9)M.E.+GS/VER		
GM	PTISense	4(H+M.E.)	$O(1)$	$O(1)$
	EPTSense	4H+6M.E.		
PA	PTISense	$O(n)$	$O(n)$	$O(n)$
	EPTSense	(n+3)H+(n+2)M.E.		$O(1)$

the server, it is impossible to link tasks to the user's real identity or link multiple tasks requested by the same user. Even though the GM and the server colludes with each other, they cannot infer the identity-task relationship through the task request token, as the BS makes the GM oblivious to which task request token issued to a certain user.

In the pseudonym request phase, only BP_i and BT_j are revealed to the PA, hence the identity-task relationship keeps hidden, even though the PA colludes with the GM. When submitting reports, since different users may have the same reputation level, it is hard for the server to deduce any linkage between reports from the same $L(P_i)$. Although the PA and the server may collude with each other to infer if two reports are sent by the same user, they have to perform extra computations to derive the pseudonyms issued to the same user, which is only considered when malicious users exist (*i.e.*, eviction) and it conflicts with our *honest-but-curious* model. When issuing receipts, the server only knows which tasks the receipts issued for, but cannot link a user to the contributed reports due to the PBS. Finally, using the BS to generate the update token provides stronger privacy protection, regardless of the possible collusions between the GM and the server.

C. Fair and Privacy-Aware Incentives

Requirement 3: A user cannot earn more rewards by using the receipts of different tasks interchangeably or the receipts of a task multiple times. If the user is not assigned any task or did not submit any report, he/she cannot earn any reward.

Analysis: In EPTSense, the receipts issued to a user contain the identifier committed to a specific task via PBS. Therefore, malicious users cannot use the receipts inconsistent with task T_i to redeem rewards from task T_i. For each receipt $R_{\mathbb{R}_k}$ received, the server would invalidate its identifier α_k, so the user can only use these receipts for one time.

If the task request is rejected, the user will not receive $\{h_i^3, n_j + 1\}_{sk_s}$, hence fails to request pseudonyms from the PA. In other words, the user cannot submit reports without valid pseudonyms (or the participant is regarded unauthorized if using randomly generated pseudonyms). On the other hand, if the user is assigned a task but does not contribute reports, he/she would get no receipt from the server. Accordingly, no reward is allocated to the user.

Requirement 4: A user cannot earn more rewards by forging receipts or stealing other receipts. The higher-quality data a user submits, the more rewards he/she will earn. When redeeming rewards, it is impossible for the server to correlate the rewards with the reports submitted before.

Analysis: In EPTSense, since each receipt is signed by the server, others would fail to forge a valid receipt. Malicious users may steal receipts to earn more rewards. However, without extra pseudonyms, they cannot submit the stolen receipts. Although some may want to replace their low-feedback receipts with that having higher feedback level, they are faced with the risk of getting fewer rewards or being detected by the server. The reason is that $l_f(\mathbb{R}_k)$ is encrypted by pk_s, others can neither distinguish the positive feedback from the negative feedback, nor tell which receipt has a higher feedback. In the worst case, if the higher-feedback receipts were usurped, the possible inconsistency of reputation level in the submitted receipts will reveal his/her malicious behavior. As shown in (7), a user will earn more rewards if he/she submits reports with higher-quality. Leveraging the PBS, it is infeasible to associate the receipts a user receives with the reports the user has submitted. Since the rewards depend on the receipts, the server cannot link rewards with the submitted reports.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of EPTSense. First, we give a brief complexity analysis. Then, we further demonstrate the performance through simulation experiments.

A. Complexity Analysis

As summarized in Table II, we compare EPTSense with PTISense in terms of computation, communication, and storage cost. Particularly, we give a detailed analysis of the computation cost which is mainly induced by the cryptographic primitives such as modular multiplication (M.M.), modular exponentiation (M.E.), hash function (H), and operations for group signature generation and verification (denoted as GS/SIG and GS/VER, respectively). Note that we set $n_j = n$ for simplicity and use RSA algorithm for digital signature and encryption, which essentially involves M.E. operations.

1) Computation Cost: We observe that PTISense and EPTSense show comparable computation cost, with more computations performed at the user and the GM for EPTSense but more computations performed at the server for PTISense. The reason is that the user needs to cooperate with the server to generate the task request and reputation update tokens (see Sections IV-C and IV-G). For request token generation, in EPTSense, the participant needs to compute h_i^3 locally and performs extra operations to obtain the server's blind signatures on h_i^1 and h_i^3, whereas h_i^3 is computed by the server and there is no need to blind two hashes in PTISense. For reputation update token generation, the user in EPTSense also needs to generate partial information locally and obtains the server's blind signature. Therefore, EPTSense induces less computation cost on the server side at the cost of the user's overhead.

As to the GM, additional cost comes from the process of blinding the user's identity and signing on $L(P_i)$ in EPTSense, which incurs more computation cost than PTISense. As to the PA, since how pseudonyms are generated is not specified in PTISense, we only give the asymptotic computation complexity $O(n)$. In contrast, EPTSense conducts $(n+3)$H+ $(n+2)$M.E. operations for each user at the PA, including verifying $\{h_i^3, n+1\}_{sk_s}$, checking hash value, and generating $n+1$ pseudonyms/certificates.

Communication Cost. As shown in Table II, the participant, the server, and the PA show the same communication complexity $O(n)$ for each task in both schemes, as n reports/receipts are transmitted and $n+1$ pseudonyms are issued to each user per task. In contrast, the GM generates a task request token for each user requesting a task, and receives a reputation update token after completing the task. Therefore, $O(1)$ communication cost is induced at the GM.

Storage Cost. For an authorized user, besides key pairs, n receipts and a reputation update (task request, respectively) token are required to be stored for a short time. Therefore, the user-side storage cost is $O(n)$. Correspondingly, in order to prevent double-spending problem, the server needs to store a request token and n receipt identifiers of a user per task. As to the GM, besides gsk and vk, he/she only stores each user's real identity and the current reputation in two schemes. For each authorized user, the PA stores $n+1$ pseudonyms in PTISense while only storing $\langle BP_i, s_i^1, s_i^2\rangle$ in EPTSense, incurring $O(n)$ and $O(1)$ storage cost in two schemes, respectively.

B. Implementation

1) Simulation Setup: In our scheme, we used the same trust and reputation evaluation model, in which we refer to [34] for the parameter setting, such as user's initial reputation (0.5), maximum sensing distance, and time gap, *etc*. We assumed that there were 100 users, out of which 10 malicious users was set in default. For benign and malicious users, they send similar and opposite data for the same task, respectively. For simplicity, we considered that similar reports had the maximum similarity 1 while opposite reports had the minimum similarity -1. To demonstrate the accuracy of our trust/reputation model, we varied the trust threshold δ from 0.3 to 0.7 with the increment of 0.1. Moreover, the number of malicious participants (η) was varied from 20 to 60 with the increment of 10, in order to show the system's robustness against malicious users. For the efficiency performance, we focused on testing the computation cost at each entity, in which the number of reports n was varied from 5 to 25 with the increment of 5 (the default value is 10) to show its impacts on the computation overhead. Additionally, we compared EPTSense with our prior scheme PTISense and a closest work from Li and Cao [21] (TTP-free scheme) which also adopts some similar cryptographic techniques for privacy protection, without considering the data trustworthiness and incentive fairness. For simplicity, the detailed pseudonym generation method in PTISense is assumed to be instantiated with the similar method to that in EPTSense with some minor adjustment. All programs were implemented in Java on Andriod smartphone (Snapdragon 820, 1.8GHz CPU and 3GB RAM, as the participant) and a laptop (AMD Athlon M320, 2.1GHz CPU and 4GB RAM, running as other entities).

2) Simulation Results: We first evaluate the accuracy of our trust/reputation evaluation model. Recall that we consider a report correct/valid if its trust value is higher than the predefined threshold δ, however, it is possible that a report is actually correct but the trust value we calculated is lower than δ, which is called false positive FP. Conversely, false negative FN means that the derived trust of a false report is higher than δ. Fig. 3 shows the rates of FP and FN with different δ as the number of tasks M increases, in which the results

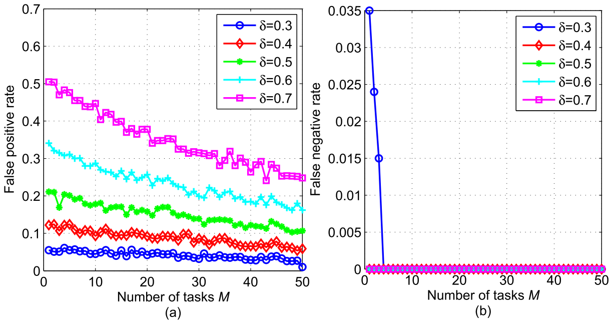


Fig. 3. The FP and FN rates with different δ.

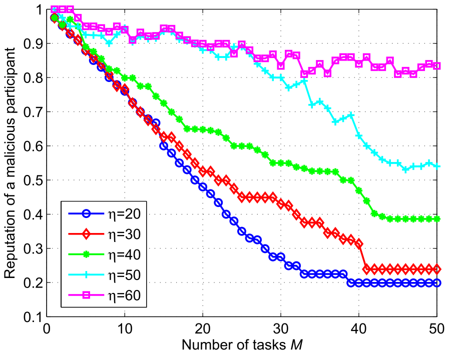


Fig. 4. Reputation of a malicious participant.

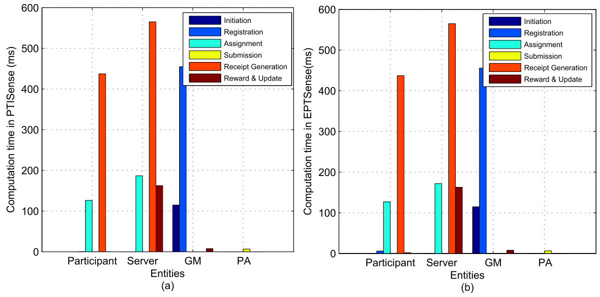
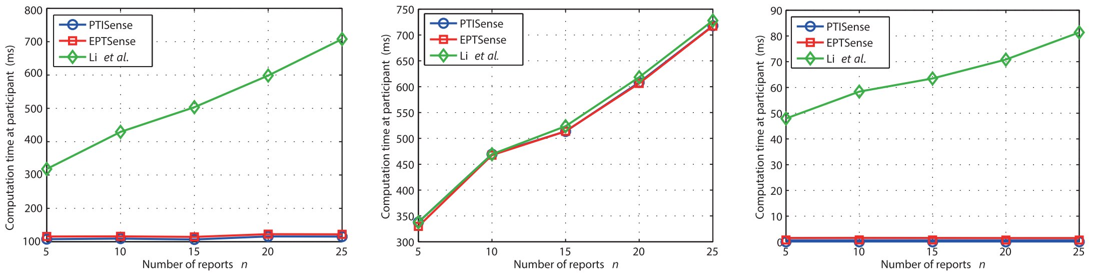


Fig. 5. The average running time of performing a task.

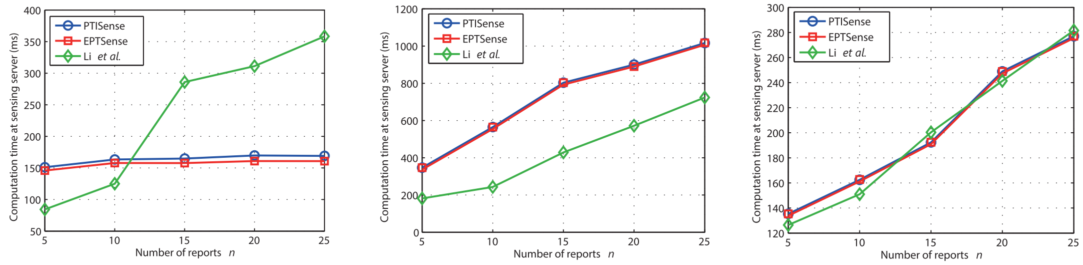


(a) Task allocation phase

(b) Receipt generation phase

(c) Reward and update phase

Fig. 6. User's computation cost in different phases.



(a) Task allocation phase

(b) Receipt generation phase

(c) Reward and update phase

Fig. 7. Server's computation cost in different phases.

are similar to PTISense [39] due to the same trust/reputation assessment model. As we can see, the FP and FN rates are both approximately 0 when δ is small. As δ increases, the FP rate grows while the FN rate keeps stable and finally remains 0. The reason is that it is more possible that a correct report is evaluated as invalid when a large δ is set. On the contrary, there is negligible possibility that a false report whose trust is more than a large δ. Additionally, we observe that with more tasks, the FP rates under different δ values all decrease. The reason is that the reputation of each user is updated after each task and it is more accurate to evaluate the report trust based on the user's newest reputation.

To further show the resilience of our reputation evaluation against malicious participants, Fig. 4 reports how a malicious user's reputation is changed with the task quantity M under different number of malicious users (η), in which the initial reputation of a malicious user is set to 1 in the worst case. As shown in the figure, with the increase of task quantity, the reputation of a malicious user drops down quickly, which finally remains stable and is close to 0 when there are a few malicious users. This is reasonable because the reports submitted by the malicious users conflict with those contributed by the majority well-behaved users. Therefore, there is high possibility that malicious users will get very low report trust and a negative feedback. As more malicious users are involved, the reputation drops more slowly, since more untrustworthy reports support each other. When there are more than 50% malicious users, the evaluation results may be dominated by the most untrustworthy reports, which results in that malicious users get high report trust and maintain a high reputation. As a result, our scheme is robust against malicious users as long as more benign users exist in the system.

To study the practicality of our proposed schemes, we mainly measure the computation cost in different phases at four entities. Fig. 5 compares each entity's running time at different phases of a task ($n = 10$) in PTISense and EPTSense. We find that both schemes show comparable computation cost, with negligibly more computations performed by the user while less computation cost induced at the server for EPTSense. This result is consistent with the above complexity analysis (see Section VI-A). Moreover, it is observed that the receipt generation accounts for the most running time for the user and server in both schemes. The reason is that n receipts are generated based on n PBS, which needs considerable computation cost for both user and server. When $n = 10$, it takes about 450ms for two schemes to generate receipts, which is considered low in our holistic trustworthy and privacy-aware sensing system. In contrast, the GM consumes more computations in the user registration phase due to the time-consuming request token generation. As to the PA, the least computation cost is incurred for both schemes as only efficient hash operations are required for pseudonym generation.

Furthermore, for fair comparison with [21], we mainly compare the computation cost at the user and the server in common phases (*i.e.*, task allocation, receipt generation, and reward allocation). The results are shown in Fig. 6 and Fig. 7, respectively. First, in the assignment phase (Fig. 6(a) and Fig. 7(a)), we observe that the computation time of EPTSense and PTISense keeps stable with n for both user and server, while that of Li *et al.* increases as n grows. This is due to the fact that a certain number of operations (*i.e.*, group signature generation and verification) are conducted at both entities in our two schemes, independent of the scale of n. Nevertheless, Li *et al.* require n PBS to generate

n report tokens generation, leading to more computation cost. As shown in Fig. 6(b), during the submission phase, EPTSense and PTISense require comparable time with Li *et al*. at the user due to the similar cryptographic operations. However, we observe that in Fig. 7(b), more server-side time is consumed due to the additional encryption of reputation feedback level, which is the cost of privacy-aware reputation management. In the reward redemption and reputation update phase, Fig. 6(c) clearly shows the superiority of EPTSense and PTISense, only with negligible cryptographic overhead at the user (1M.E. operation in PTISense and 3M.E.+2M.M.+H in EPTSense). In contrast, large overhead is incurred in [21] due to the fact that multiple blinding and unblinding operations are conducted at the user when redeeming rewards. For the server, as presented in Fig. 7(c), the three schemes have similar cost. Overall, EPTSense can achieve trustworthy and enhanced privacy-aware crowdsensing, which has comparable computation cost with PTISense and far less cost than the TTP-free scheme of Li *et al*., especially on the user side.

VII. Conclusions

In this paper, we proposed EPTSense, a multi-entity-assisted mobile crowdsensing system with enhanced security, which simultaneously addresses the data trustworthiness, user privacy, and incentive fairness. In our scheme, legitimate users are able to join tasks anonymously and obtain the corresponding rewards while malicious users cannot abuse anonymity to earn more rewards or improve their reputations. Otherwise, misbehavior would be detected with users revoked from the system or a task. Compared to our previous work PTISense, we further remove the assumption of a trust PA and is resistant against possible collusion attacks. Security analysis demonstrates that EPTSense achieves the same security and privacy goals as PTISense under a TTP-free and two-entity collusive model. Our prototype implementation further shows the efficiency and practicality via comparisons.

Acknowledgment

The information reported here does not reflect the the policy of the funding agencies.

References

[1] M. Abe and E. Fujisaki, “How to date blind signatures,” in *Advances in Cryptology—ASIACRYPT*. Berlin, Germany: Springer, 1996, pp. 244–251.

[2] M. A. Alsheikh, Y. Jiao, D. Niyato, P. Wang, D. Leong, and Z. Han, “The accuracy-privacy trade-off of mobile crowdsensing,” *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 132–139, Jun. 2017.

[3] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, “A secure privacy-preserving data aggregation scheme based on bilinear ElGamal cryptosystem for remote health monitoring systems,” *IEEE Access*, vol. 5, pp. 12601–12617, 2017.

[4] J. Camenisch and J. Groth, “Group signatures: Better efficiency and new theoretical aspects,” in *Proc. Int. Conf. Secur. Commun. Netw.*, vol. 3352, 2004, pp. 120–133.

[5] D. Chaum, “Blind signature system,” in *Advances in Cryptology*, 1984.

[6] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, “IncogniSense: An anonymity-preserving reputation framework for participatory sensing applications,” *Pervas. Mobile Comput.*, vol. 9, no. 3, pp. 353–371, Jun. 2013.

[7] C. Cornelius *et al.*, “Anonysense: Privacy-aware people-centric sensing,” in *Proc. MobiSys*, Jun. 2008, pp. 211–224.

[8] E. De Cristofaro and C. Soriente, “Extended capabilities for a privacy-enhanced participatory sensing infrastructure (PEPSI),” *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 12, pp. 2021–2033, Dec. 2013.

[9] L. Duan *et al.*, “Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing,” in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1701–1709.

[10] X. Fan, J. Liu, Z. Wang, Y. Jiang, and X. Liu, “Crowdsourced road navigation: Concept, design, and implementation,” *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 126–128, Jun. 2017.

[11] D. Förster, F. Kargl, and H. Löhr, “PUCA: A pseudonym scheme with strong privacy guarantees for vehicular ad-hoc networks,” *Ad Hoc Netw.*, vol. 37, pp. 122–132, Feb. 2016.

[12] H. Gao *et al.*, “Online quality-aware incentive mechanism for mobile crowd sensing with extra bonus,” *IEEE Trans. Mobile Comput.*, to be published. doi: 10.1109/TMC.2018.2877459.

[13] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, “Security, privacy, and incentive provision for mobile crowd sensing systems,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 839–853, Oct. 2016.

[14] K. Huang, S. S. Kanhere, and W. Hu, “Are you contributing trustworthy data?: The case for a reputation system in participatory sensing,” in *Proc. MSWIM*, Oct. 2010, pp. 14–22.

[15] K. L. Huang, S. S. Kanhere, and W. Hu, “A privacy-preserving reputation system for participatory sensing,” in *Proc. LCN*, Oct. 2012, pp. 10–18.

[16] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, “Quality of information aware incentive mechanisms for mobile crowd sensing systems,” in *Proc. MobiHoc*, Jun. 2015, pp. 167–176.

[17] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, “Enabling privacy-preserving incentives for mobile crowd sensing systems,” in *Proc. ICDCS*, Jun. 2016, pp. 344–353.

[18] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, “Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2019–2032, Oct. 2018.

[19] R. F. El Khatib, N. Zorba, and H. S. Hassanein, “A fair reputation-based incentive mechanism for cooperative crowd sensing,” in *Proc. GlobeCom*, Oct. 2018, pp. 1–6.

[20] M. Li, J. Lin, D. Yang, G. Xue, and J. Tang, “QUAC: Quality-aware contract-based incentive mechanisms for crowdsensing,” in *Proc. MASS*, Oct. 2017, pp. 1–9.

[21] Q. Li and G. Cao, “Providing privacy-aware incentives in mobile sensing systems,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1485–1498, Jun. 2016.

[22] Q. Li and G. Cao, “Providing privacy-aware incentives for mobile sensing,” in *Proc. PerCom*, Mar. 2013, pp. 76–84.

[23] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, “Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 8, pp. 1851–1864, Aug. 2018.

[24] Z. Liu, S. Jiang, P. Zhou, and M. Li, “A participatory urban traffic monitoring system: The power of bus riders,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2851–2864, Oct. 2017.

[25] Z. Luo and X. Huang, “A personalized k-anonymity with fake position generation for location privacy protection,” in *Proc. Internet Conf. China*, Apr. 2014, pp. 46–55.

[26] L. Ma, X. Liu, Q. Pei, and Y. Xiang, “Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing,” *IEEE Trans. Services Comput.*, to be published. doi: 10.1109/TSC.2018.2825986.

[27] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. S. Shen, “Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, to be published. doi: 10.1109/TMC.2019.2908638.

[28] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, “SACRM: Social aware crowdsourcing with reputation management in mobile sensing,” *Comput. Commun. J.*, vol. 65, pp. 55–65, Jul. 2015.

[29] X. Sheng, J. Tang, X. Xiao, and G. Xue, “Sensing as a service: Challenges, solutions and future directions,” *IEEE Sensors J.*, vol. 13, no. 10, pp. 3733–3741, Oct. 2013.

[30] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, “AnonySense: A system for anonymous opportunistic sensing,” *Pervasive Mobile Comput.*, vol. 7, no. 1, pp. 16–30, Feb. 2011.

[31] J. Son *et al.*, “Privacy aware incentive mechanism to collect mobile data while preventing duplication,” in *Proc. MILCOM*, Oct. 2015, pp. 1242–1247.

[32] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, “Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing,” in *Proc. ICDCS*, Jun. 2016, pp. 1–10.

[33] Q. Wang *et al.*, “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy,” *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 591–606, Jul./Aug. 2018.

[34] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, “Enabling reputation and trust in privacy-preserving mobile sensing,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2777–2790, Dec. 2014.

[35] Z. Wang *et al.*, “Personalized privacy-preserving task allocation for mobile crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.

[36] Z. Wang *et al.*, “Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform,” in *Proc. INFOCOM*, Apr./May 2019, pp. 2053–2061.

[37] Z. Wang *et al.*, “Privacy-preserving crowd-sourced statistical data publishing with an untrusted server,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1356–1367, Jun. 2019.

[38] H.-Q. Wu, L. Wang, and G. Xue, “Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing,” *IEEE Trans. Netw. Sci. Eng.*, to be published. doi: 10.1109/TNSE.2019.2892583.

[39] H. Wu, L. Wang, G. Xue, J. Tang, and D. Yang, “Privacy-preserving and trustworthy mobile sensing with fair incentives,” in *Proc. ICC*, May 2019, pp. 1–7.

[40] D. Yang, G. Xue, X. Fang, and J. Tang, “Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing,” in *Proc. MobiCom*, Aug. 2012, pp. 173–184.

[41] D. Yang, G. Xue, X. Fang, and J. Tang, “Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

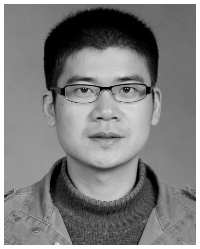
[42] J. Zhang, L. He, Q. Zhang, and Y. Gan, “Pseudonym-based privacy protection scheme for participatory sensing with incentives,” *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 11, pp. 5654–5673, Nov. 2016.

[43] J. Zhang, J. Ma, W. Wang, and Y. Liu, “A novel privacy protection scheme for participatory sensing with incentives,” in *Proc. Int. Conf. Cloud Comput. Intell. Syst.*, vol. 3, Oct./Nov. 2012, pp. 1017–1021.

[44] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, “Countermeasures against false-name attacks on truthful incentive mechanisms for crowdsourcing,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 478–485, Feb. 2017.

[45] Y. Zheng, H. Duan, and C. Wang, “Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing,” *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2475–2489, Oct. 2018.

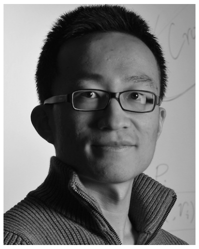
Haiqin Wu received the bachelor’s degree in computer science from Jiangsu University in June 2014, where she is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering. She was a visiting Ph.D. student at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. Her research interests include data security and privacy, mobile crowdsensing, data query processing, and blockchain-based applications.



Liangmin Wang received the B.S. degree in computational mathematics from Jilin University, Changchun, China, in 1999, and the Ph.D. degree in cryptology from Xidian University, Xi’an, China, in 2007. He is currently a Professor of computer science and communication engineering with Jiangsu University, Zhenjiang, China. He has published over 70 technical papers at premium international journals and conferences, such as TITS, TNSE, TVT, IoT Journal, GLOBECOM, and so on. His research interests include data security and privacy. He has served as a TPC member for many IEEE conferences, such as the IEEE ICC, IEEE HPCC, and IEEE TrustCom. He has been honored as a “Wan-Jiang Scholar” of Anhui Province since November 2013. He is currently an Associate Editor of the *Security and Communication Networks*, a member of the ACM, and a Senior Member of the Chinese Computer Federation.

Guoliang Xue (F’11) received the Ph.D. degree in computer science from the University of Minnesota in 1991. He is currently a Professor of computer science and engineering with Arizona State University. His research interests span the areas of quality of service provisioning, network security and privacy, crowdsourcing and network economics, RFID systems and Internet of Things, smart city, and smart grids. He has published over 280 papers in these areas, many in top conferences such as INFOCOM, MOBICOM, NDSS, and top journals such as IEEE/ACM TON, IEEE JSAC, and IEEE TMC. He has received the IEEE Communications Society William R. Bennett Prize in 2019 (Best Paper Award for IEEE/ACM TON and IEEE TNSM in the previous three years). He was a keynote speaker at the IEEE LCN’2011 and ICNC’2014. He was the TPC Co-Chair of the IEEE INFOCOM’2010 and the General Co-Chair of the IEEE CNS’2014. He has served on the TPC of many conferences, including ACM CCS, ACM MOBIHOC, IEEE ICNP, and IEEE INFOCOM. He has served on the Editorial Board of the IEEE/ACM TRANSACTIONS ON NETWORKING. He serves as the Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, overseeing 13 editors in the Wireless Networking Area.

Jian Tang (F’19) received the Ph.D. degree in computer science from Arizona State University in 2006. His research interests lie in the areas of machine learning, IoT, wireless networking, big data systems, and cloud computing. He has published over 140 papers in premier journals and conferences. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Syracuse University. He received an NSF CAREER Award in 2009. He also received several Best Paper Awards, including the 2019 William R. Bennett Prize and the 2019 TCBD (Technical Committee on Big Data) Best Journal Paper Award from the IEEE ComSoc, the 2016 Best Vehicular Electronics Paper Award from the IEEE Vehicular Technology Society (VTS), and the Best Paper Awards from the 2014 IEEE ICC and the 2015 IEEE Globecom, respectively. He has served as an Editor for several IEEE journals, including the IEEE TBD, IEEE TMC, and so on. In addition, he has served as the TPC Co-Chair for a few international conferences, including the IEEE/ACM IWQoS2019, MobiQuitous2018, and IEEE iThings2015; as the TPC Vice Chair for the INFOCOM2019; and as an Area TPC Chair for INFOCOM 2017–2018. He is also an IEEE VTS Distinguished Lecturer, and the Vice Chair of the Communications Switching and Routing Committee of the IEEE ComSoc.



Dejun Yang (M’13–SM’19) received the B.S. degree in computer science from Peking University, Beijing, China, in 2007, and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. He is currently an Associate Professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing, with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security, and privacy problems. He has received the IEEE Communications Society William R. Bennett Prize in 2019 (Best Paper Award for IEEE/ACM TON and IEEE TNSM in the previous three years), and the Best Paper Awards at the IEEE GLOBECOM (2015), the IEEE MASS (2011), and the IEEE ICC (2011 and 2012), as well as the Best Paper Award Runner-up at the IEEE ICNP (2010). He is the TPC Vice Chair of information systems for the IEEE INFOCOM 2020, a Student Travel Grant Co-Chair for INFOCOM 2018–2019, and was the Symposium Co-Chair for the International Conference on Computing, Networking and Communications (ICNC) 2016. He currently serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL.