Privacy-Preserving and Trustworthy Mobile Sensing with Fair Incentives

Haiqin Wu, Liangmin Wang, Guoliang Xue, Jian Tang and Dejun Yang

***Abstract*—Pervasive mobile devices and their advances in sensing and networking have led to an emerging mobile sensing paradigm. The diversity of mobile users and the openness of sensing systems raise several crucial concerns for users' privacy, data quantity, and quality. Although different aspects of these issues were addressed separately in existing researches, there is still a need to provide a holistic solution for secure and privacy-aware mobile sensing. In this paper, we propose a privacy-aware and trustworthy mobile sensing scheme with fair incentives. Leveraging group signature, (partial) blind signature, and limited number of pseudonyms technologies, our scheme enables well-behaved users to contribute their data anonymously, and prevents both greedy and malicious users from abusing the privacy protection. Moreover, we design a fair incentive scheme to stimulate users to contribute high-quality data, based on the data quality and the reputation feedback level. Security analysis demonstrates that our proposed scheme achieves the security goals. Extensive evaluation results are presented which demonstrate the effectiveness and efficiency of our scheme.**

***Index Terms*—mobile sensing, privacy preservation, trustworthiness, fair incentive.**

I. Introduction

Recent years have witnessed a rapid proliferation of mobile devices such as smartphones and tablets. With the advances of sensing and communication technologies, these mobile devices are generally equipped with various powerful embedded sensors (e.g., camera, GPS) and have enhanced communication capacities (e.g., WiFi, 4G, and Bluetooth). Due to these advancements, mobile sensing has emerged as a new sensing paradigm. Compared with the traditional static sensor-based wireless sensing, mobile sensing has exhibited numerous advantages such as lower deployment cost and better spatial-temporal coverage. With personal mobile devices, users can collect various sensing data from nearby environments, which fosters many promising applications, including environmental monitoring, assistive healthcare, and intelligent transportation.

However, we observe three crucial issues that might impede the large-scale deployment of these applications. First, privacy disclosure is a potential obstacle that prevents users from participating in sensing tasks, as their contributed data may reveal some sensitive information such as identity, location, or health status [2]. Therefore, there is an inherent necessity to provide users a privacy-aware and anonymous mobile sensing scheme. The second issue lies in the design of incentives in a fair manner to attract more user participation, which can provide sufficient sensing data and improve the quality of sensing service. However, a user would be reluctant to take sensing tasks unless desirable incentives are provided as compensation for their energy (e.g., battery) consumption. The third issue is the data reliability. In practical mobile sensing applications, due to the massive and open involvement of diverse participants, it is hard to guarantee that all participants would submit accurate and reliable sensing data.

There have been some efforts devoted to relevant researches such as the privacy protection technologies proposed for anonymous data collection ([6], [12]), incentive schemes [19], and some trust/reputation management systems ([8], [15]). However, these solutions only address these issues separately, rather than addressing all of them collectively. Although privacy-aware incentives and anonymous reputation systems were further studied, they still fail to consider these issues in a holistic perspective. It is nontrivial to address these issues simultaneously, as some combined issues may bring new challenges, such as the inherent conflict between user privacy and data trustworthiness, the potential abuse attack in privacy-aware incentives, and the fairness of incentives. In these cases, how to protect the privacy of benign users and prevent malicious users from breaking the data trustworthiness and fairness of incentives is much more challenging.

In this paper, we propose a practical integrated scheme providing privacy-preserving and trustworthy mobile sensing with fair incentives. Compared with previous researches ([10], [14], [17], [18]), our scheme is applicable to the multiple-report scenario[1] which is rarely considered except for [11], [16]. However, [11] requires a trusted authority for authentication, and large overhead is induced when generating anonymous report tokens. In contrast, we adopt group signature for anonymous user authentication, where the group manager is not fully trusted. A limited pseudonym-based approach is crafted to let participants anonymously submit their reports while prevent-

Wu and Wang are with School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. Xue is with Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85287. Tang is with Department of Electrical Engineering and Computer Science, Syracuse University Syracuse, NY 13244. Yang is with Department of Computer Science, Colorado School of Mines, Golden, CO 80401. This work was supported in part by NSF grants 1717197, 1421685, 1717315, and 1525920 , the National Natural Science Foundation of China under Grants U1736216, 61472001, and 61702233, the National Key Research and Development Program Grant 2017YFB1400703, and the Graduate student scientific research innovation projects of jiangsu province Grant KYCX17_1810. The information reported here does not reflect the position or the policy of the funding agencies. This work was done while Wu was visiting Arizona State University under the support of Chinese Scholarship Council (No. 201708320241).

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

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

ing malicious users from submitting more reports. Inspired by [15], we integrate an anonymous reputation management scheme with our new system model, enabling privacy-aware trust assessment and reputation update at different entities. Particularly, based on the data quality and reputation feedback, a fair payment allocation method is further developed to reward participants. Finally, we provide flexible revocation methods to evict participants from tasks or the whole system.

The remainder of this paper is organized as follows. Section II and Section III review some related work and preliminaries. We present our scheme in Section IV. Security analysis and performance evaluations are shown in Section V and Section VI, respectively. Finally, Section VII concludes this paper.

II. Related Work

For general privacy protection, Shin et al. [13] first proposed AnonySense for mobile sensing systems. It provides frameworks for anonymous tasking and reporting leveraging mix network and k-anonymity technology. However, this scheme lacks provable privacy guarantees. Considering the privacy and incentive issues simultaneously, Zhang et al. [18] first solved this problem with pseudonym, encryption and hash function. In [10], two privacy-aware incentive schemes were designed to reward participants with credits in single-report tasks. The first scheme relies on a **T**rusted **T**hird **P**arty (TTP) while the second adopts blind signature and commitment techniques to preserve privacy. These two schemes were further improved in [11] which supports both single-report and multiple-report tasks. Son et al. [14] realized privacy-preserving mobile incentives with efficient pseudonym verification. Particularly, duplicate data with different pseudonyms can be detected by revealing the user's private key. Besides these, privacy-aware auction [9] is also studied as incentive mechanism. However, none of these solutions consider the trustworthiness of sensed data.

To improve the quality of sensed data without compromising user's privacy, [8] assigned each participant multiple pseudonyms and relied on a TTP to transform the reputation between multiple pseudonyms of the same participant. A similar solution IncogniSense [5] was proposed by using blind signatures and cloaking techniques. As an improvement, Wang et al. [15] proposed ARTSense, which contains a privacy-aware trust assessment and an anonymous reputation protocol without the existence of TTP. Nevertheless, no incentives are provided in these solutions. Although Gisdakis et al. [7] proposed a secure and accountable mobile sensing system that preserves the user privacy and provides incentives based on the Shapley value. However, it lacks a detailed reputation evaluation method suitable for the multi-report scenario.

III. Preliminaries

A. System Architecture

In this paper, we consider our mobile sensing system consisting of the following entities.

1) Data collectors (DCs): organizations or individuals who create sensing tasks by specifying some task requirements, such as the specific sensing area/time, the reward budget, and other requirements.
2) Sensing servers (or servers): entities receiving tasks from the DCs and publishing tasks to the users. After a task finishes, servers will give rewards and feedback to the users, based on the evaluation of sensing data.
3) Participants: mobile users[2] collect sensing data with their mobile devices for requested tasks.
4) Group manager (GM): entity acting for user registration, reputation management, and request tokens issuance.
5) Trusted pseudonym authority (TPA): an authority who issues valid pseudonyms to authorized participants for their reports and receipts submission.

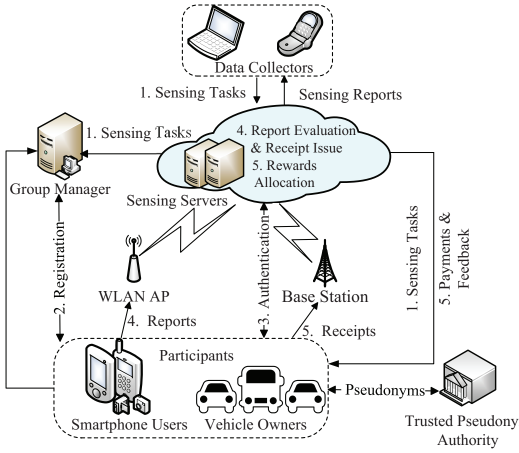

Fig. 1. System architecture

The system architecture is illustrated in Fig. 1. First, the DCs create sensing tasks and then forward them to the servers. Subsequently, the servers publish these tasks to the group manager and mobile users (Step 1). In this paper, we consider the DCs and the servers the same party for simplicity, as DCs' privacy is not under our consideration.

If a mobile user wants to join a task, he/she must register with the group manager and obtain relevant task request tokens (Step 2). After being authenticated (Step 3), the user can request corresponding pseudonyms from the TPA, with which the sensing reports can be later submitted to the server anonymously. For each report received, the server evaluates its reliability and issues a receipt to the participant (Step 4). After completing a task, the participant submits all his/her receipts to the server and gets corresponding rewards. Meanwhile, a reputation feedback is also returned (Step 5). Finally, the participant submits the feedback to the group manager using his/her real identity for reputation update (Step 6).

B. Threat model and assumptions

Threats to privacy. Curious group manager may want to know which tasks the user is interested in. The server may be curious about the real identity of participants and whether two tasks/reports are taken/submitted by the same participant.

[2] In this paper, we use participant and user interchangeably.

Threats to incentives. Greedy participants may try to earn more rewards by submitting extra reports. Some malicious participants may try to use tokens obtained from different tasks interchangeably, use a token twice or usurp others' tokens.

Threats to trustworthiness. Unauthenticated users may contribute forged data to the server. For legitimate participants, they may exhibit malicious behaviors, including submitting false sensing data randomly for certain purposes or collusively send the same false data to disrupt the sensing applications.

Assumptions. Servers and the group manager are "*honest but curious*", indicating that they will follow the designated protocols, but are curious to infer user's privacy. Moreover, the communications between users and servers are anonymized by Mix Networks or IP and MAC address recycling techniques.

C. Design goals

The following are our design goals:

G1 Privacy-preserving participation: Group manager and servers cannot infer if a given participant has requested/accepted a specific task, or whether two or more tasks have been accepted by the same participant.

G2 Fair and privacy-aware incentives: Participants should be rewarded fairly based on their data quality in a privacy-aware manner. Malicious users cannot increase their rewards by abusing pseudonyms, reusing, or stealing tokens.

G3 Data trustworthiness. Participants should be authenticated before task assignment. Additionally, an anonymous reputation assessment scheme should be built to mitigate data trustworthiness threats.

D. Cryptography and Reputation Primitives

Group signature [3]. A group signature scheme allows a member of a group to anonymously sign messages on behalf of the group. Specifically, the key generation algorithm KeyGen() outputs a public verification key vk and a group secret key gsk. A new member i will obtain a member secret key msk_i after joining the group. Any group member can sign a message m with msk_i and others is able to verify the signature with vk. If necessary, the group manager can identify, trace, and revoke the signer with gsk. Group signature has two properties: anonymity (except for the group manager) and traceability (only for the group manager), which captures the security requirements in our system.

Blind signature and partial blind signature. Blind signature [4] enables a user to obtain a signature from a signer without knowing the message m to be signed. Specifically, the user chooses a blinding factor b relatively prime to the signer's public modulo Q, and computes $m' = m \cdot b^e mod\ Q$ (blind RSA signature), where e is the signer's public key. The signer signs on m' with k and sends the signature $\{m'\}_k$ to the user. By computing $\{m\}_k = (\{m'\}_k \cdot b^{-1}) mod\ Q$, the user can obtain the real signature. Besides blindness and unlinkability, the user cannot forge a valid signature from $\{m'\}_k$ for another different message m. In contrast, partially blind signature [1] enables the signer to add some public information in the signature, while others are similar to the blind signature.

Trust and reputation. Following the definition in [15]. We use "trust" and "reputation" to assess the sensing reports and the participants, respectively. Particularly, "reputation level" is employed for privacy protection, which is a discrete approximation deduced from the participant's reputation.

IV. The PTISense scheme

In this section, we present our scheme PTISense, an integrated scheme achieving the goals on "**P**rivacy Preservation", "Data **T**rustworthiness" and "Fair **I**ncentives" for mobile **S**ensing. The key challenge to be addressed is how to protect the privacy of well-behaved participants while preventing misbehaving users from launching abuse attacks. We adopt the idea of limited number of pseudonyms to avoid users submitting more reports. In the whole process, blind signature and partial blind signature are employed to delink the correlation between data. Moreover, a fair incentive is designed to reward users in different degrees. To tackle malicious participants, we provide two revocation methods based on the anonymous reputation evaluation. Our entire scheme consists of seven phases for each task group, and the detailed interactions between our system entities are as follows.

A. Initialization

In this phase, a certificate authority first delivers a key pair to the server and the group manager, respectively. Moreover, the group manager performs KeyGen() to generate a group public key vk and a group secret key gsk.

The server groups all tasks (e.g., indexed as $1, 2, \ldots, M$ in the order of their reception time) received from the DCs. Then, the server publishes M tasks to the mobile users.

B. Participant registration

If a participant P_i wants to take task T_j for the first time, he/she must register with the group manager and obtain msk_i. Then, P_i needs to send some private information and acquire the corresponding task request token. Using task request token is to let the server anonymously authenticate the legitimacy of users and determine whether to authorize users their requested tasks. In this paper, a task request token is constructed by binding the user's identity, reputation $R(P_i)$, reputation level $L(P_i)$, and the blinded task ID. Specifically, P_i first computes the blinded task ID $BT_j = T_j \cdot b^{pk_{GM}} mod\ Q$ using the group manager's public key pk_{GM} and then sends a task token request (TTR) to the group manager with his/her real identity.

The group manager maintains a reputation table for users with a preset initial reputation. After receiving TTR, the group manager first derives $h_i^1 = H(P_i|R(P_i)|BT_j|\rho)$, $h_i^2 = H(P_i|BT_j)$, where H is a one-way hash function and ρ is a nonce. Then, based on its blind signature on BT_j (sk_{GM} is the signing/private key), P_i's task request token for T_j is constructed as $\tau_i^j = \{h_i^1, h_i^2, \{BT_j\}_{sk_{GM}}, L(P_i)\}_{sk_{GM}}$.

C. Participant authentication and task assignment

To prevent malicious users from using tokens inconsistent with the requested task, the actual task ID should be revealed to the server in this phase. Specifically, P_i generates a group signature $\{b\}_{msk_i}$, which is sent along with b, τ_i^j, and T_j. Specifically, P_i generates a random pseudonym p_i^0 and sends a anonymous task request $\Re_i = \langle p_i^0, T_j, \{b\}_{msk_i}, b, \tau_i^j \rangle$ to the server. Upon receiving $\Re_i$, based on b and vk, the server can verify $\{b\}_{msk_i}$ anonymously. If it succeeds, P_i is considered legitimate. To further verify τ_i^j, the server performs:

1) It verifies the authenticity of τ_i^j by checking the signature of the group manager with his/her public key pk_{GM}.
2) It extracts $\{BT_j\}_{sk_{GM}}$ from the token and obtains $\{T_j\}_{sk_{GM}}$ by removing the blinding factor b.
3) It verifies $\{T_j\}_{sk_{GM}}$ and ensures the correctness of τ_i^j.

If all steps succeed and τ_i^j has not been used, the token is considered authentic and correct. Next, the server extracts $L(P_i)$ from τ_i^j and decides whether $L(P_i)$ satisfies the task requirement. If it satisfies, τ_i^j is stored and tagged as *approved* to prevent token reuse. Meanwhile, the server computes $h_i^3 = H(h_i^2|n_{T_j} + 1|1)$ and returns an approval message $A_i = \langle \{h_i^1\}_{sk_{ss}}, \{h_i^3\}_{sk_{ss}}, \{T_j|L(P_i)\}_{sk_{ss}} \rangle$, where n_{T_j} is the number of reports required by T_j, and h_i^3 is used to request pseudonyms. Conversely, $\{h_i^1|0\}_{sk_{ss}}$ is returned to P_i.

D. Report submission and trust evaluation

Before submitting reports, P_i needs to get $n_{T_j} + 1$ pseudonyms from the TPA. Specifically, P_i sends a pseudonym request $\langle P_i, BT_j, n_{T_j}{+}1, \{h_i^3\}_{sk_{ss}} \rangle$ to the TPA. After verifying $\{h_i^3\}_{sk_{ss}}$, the TPA returns pseudonyms $p_i^1, p_i^2, \ldots, p_i^{n_{Tj+1}}$.

With the obtained pseudonyms, P_i can submit reports for T_j. Particularly, each report is submitted anonymously as $\mathbb{R}_k = \langle p_i^k, T_j, \{T_j|L(P_i)\}_{sk_{ss}}, D_i^k \rangle, (k = 1, 2, \ldots, n_{T_j})$, where D_i^k is the kth data. $\{T_j|L(P_i)\}_{sk_{ss}}$ is included for later report trust evaluation. For each sensing report received, the server first verifies the validity of the pseudonym p_i^k, and then validates $\{T_j|L(P_i)\}_{sk_{ss}}$ and ensures that the task ID in it is T_j.

If both checks are passed, the server then assesses the trust of each report. Regarding the trust assessment approach, we resort to [15] which evaluates the report in a comprehensive perspective. First, the basic trust of a report $\mathbb{R}_k$ is computed based on the reputation level $L(P_i)$ and some contextual factors (e.g., time/location). Then, its final trust can be derived by further considering the similarity of data submitted by different participants. Unlike [15], our trust assessment and reputation update are performed at two different entities instead of the single server. Moreover, in our multiple-report scenario, reports for a particular task T_j are further divided into n_{T_j} collections, and the data similarity for a certain report is computed based on one of these collections.

Let $T_F(\mathbb{R}_k)$ denote the final trust of a report $\mathbb{R}_k$. After deriving $T_F(\mathbb{R}_k)$, the server can obtain a feedback level $l_f(\mathbb{R}_k)$ by comparing $T_F(\mathbb{R}_k)$ with $L(P_i)$. Generally, a positive feedback is set when $T_F(\mathbb{R}_k) > L(P_i)$ and a negative feedback otherwise. Moreover, two reports with similar gaps would have the same feedback level, such that the server cannot associate $l_f(\mathbb{R}_k)$ with the related report when later submitting receipts.

After receiving $\mathbb{R}_k$, the server issues a receipt $R_{\mathbb{R}_k}$ to P_i, which can be used to redeem rewards later. Particularly, to achieve the distinguishability and unlinkability of receipts, we adopt partial blind signature, in which T_j is the common information shared by the user and the server. Specifically, P_i computes $\alpha_k = H(h_i^1|T_j|k)$ as the receipt identifier and obtains the partial blind signature $\{\alpha_k, T_j\}_{sk_{ss}}$ from the server. Meanwhile, the server sends $\{\{T_j|L(P_i)\}_{sk_{ss}}|[l_f(\mathbb{R}_k)]_{pk_{ss}}\}_{sk_{ss}}$ to P_i. Based on this, the receipt $R_{\mathbb{R}_k}$ is as follows:

$$R_{\mathbb{R}_k} = \langle T_j, \{\alpha_k, T_j\}_{sk_{ss}}, \{\{T_j|L(P_i)\}_{sk_{ss}}|[l_f(\mathbb{R}_k)]_{pk_{ss}}\}_{sk_{ss}} \rangle. \tag{1}$$

E. Receipt submission and user remuneration

When the server announces the completion of task T_j, each participant can submit all receipts he/she obtained. Specifically, P_i sends $\langle p_i^{n_{T_j}+1}, h_i^1, (\alpha_k, R_{\mathbb{R}_k})_{k=1,\ldots,n_{T_j}} \rangle$ to the server. Subsequently, the server does some verifications:

1) It verifies the validity of $p_i^{n_{T_j}+1}$ and $\{\alpha_k, T_j\}_{sk_{ss}}$, ensuring that P_i is authorized and has submitted n_{T_j} reports for task T_j.
2) It checks each $\alpha_k = H(h_i^1|T_j|k)$ to ensure that these receipts are really issued to P_i. Anyone who steals other's receipts (without h_i^1) cannot pass the verification.

If both checks succeed, the server stores and invalidates α_k to avoid receipt reuse. Then, it decrypts $[l_f(\mathbb{R}_k)]_{pk_{ss}}, (k = 1, \ldots, n_{T_j})$ and gets $l_f(\mathbb{R}_k)$, based on which the average feedback $\overline{l_f}$ can be computed. Eventually, the server obtains the average report trust of P_i by calculating $\overline{T_F} = \overline{l_f} + L(P_i)$.

To realize fair incentives, we enable participants with higher report trust to earn more rewards. Moreover, different strategies are adopted to reward positive-feedback participants S_P and negative-feedback participants S_N, respectively. Given the task budget B_{T_j}, the reward distributed to each participant $P_i \in S_N$ is

$$r_i = \frac{\overline{T_F}(P_i, T_j)}{\sum_{P_k \in P} \overline{T_F}(P_k, T_j)} \cdot B_{T_j} \cdot e^{\overline{l_f}(P_i, T_j) \cdot \psi}, \tag{2}$$

where ψ is an amplification factor to increase the effect of the negative feedback on the reward allocation. Obviously, the reward paid to negative-feedback participant is less than their real contribution, which can be regarded as a punishment.

For each participant $P_{i'} \in S_P$, the reward paid is

$$r_{i'} = \frac{\overline{T_F}(P_{i'}, T_j)}{\sum_{P_k \in S_P} \overline{T_F}(P_k, T_j)} \cdot (B_{T_j} - \sum_{P_i \in S_N} r_i). \tag{3}$$

Additionally, the server also returns P_i a reputation update token $U_{T_j} = \langle T_j, \{H(h_i^1|\overline{l_f})\}_{sk_{ss}}, \{[\overline{l_f}]_{pk_{GM}}\}_{sk_{ss}} \rangle$ for task T_j.

F. Reputation update

In this phase, P_i needs to return feedback information to the group manager for reputation update as long as he/she requested tasks. Specifically, upon receiving U_{T_j}, P_i sends a "blinded" reputation token $U_{BT_j} = \langle BT_j, \{H(h_i^1|\overline{l_f})\}_{sk_{ss}}, \{[\overline{l_f}]_{pk_{GM}}\}_{sk_{ss}} \rangle$ with his/her real identity. The group manager verifies the signature of the server, and obtains $\overline{l_f}$ after decryption. Subsequently, it verifies that $H(P_i|R(P_i)|BT_j|\rho|\overline{l_f}) = H(h_i^1|\overline{l_f})$, which prevents users from stealing update tokens. After successful verification, the group manager updates P_i's reputation based on $\overline{l_f}$. Moreover, it stores U_{BT_j} and tags it as *used* to prevent token reuse.

Conversely, if P_i is not authorized to T_j, he/she also needs to return a request feedback $F_i = \langle P_i, BT_j, \{h_i^1|0\}_{sk_{ss}} \rangle$ to the group manager. Upon receiving F_i, the group manager verifies $\{h_i^1|0\}_{sk_{ss}}$ and checks if the task request is indeed rejected by comparing $H(P_i|R(P_i)|BT_j|\rho)|0$ with $h_i^1|0$. 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.

G. Participant eviction

To further improve the data trustworthiness, PTISense provides countermeasures to tackle users with low reputations.

In the initialization phase, the system sets a reputation level threshold, below which the participant is considered unreliable and should not be assigned any task. Specifically, after successful verification of τ_i^j, the server extracts $L(P_i)$ and checks if $L(P_i)$ is less than the threshold. If it satisfies, the server will deliver $\{b\}_{msk_i}$ to the group manager who can open the signature with gsk and reveal the identity of P_i. Finally, the participant will be added to the blacklist and cannot obtain any task request token from the group manager.

However, some reputation-qualified participants may submit low-quality data occasionally due to certain motivations. To mitigate this, PTISense can identify these participants with the cooperation of TPA. Specifically, once the trust of a report is detected lower than the preset trust threshold ε, the server will send its pseudonym to the TPA. TPA retrieves other pseudonyms and sends them to the server. Hence, the participant utilizing these pseudonyms will not get any receipt.

V. Security analysis

In this section, we show that PTISense can achieve our defined goals $G1 - G3$.

Requirement 1. *The group manager and the server can neither associate the participant ID with his/her requested tasks (submitted reports) nor link multiple tasks (reports) accepted (contributed) by the same participant, as long as the two entities do not collude with each other.*

Analytical validation. Although the participant's real ID is included in TTR, the exact task ID is blinded with b. Given two different tasks, the group manager cannot identify if they are requested by the same participant. In the task assignment phase, a participant can get authenticated anonymously via group signature. Although the task ID is disclosed to the server, it is impossible to link tasks to the user's real identity or link multiple tasks requested by the same participant, as long as there is no collusion between group manager and server.

In the report submission phase, although $L(P_i)$ is included in the report, the server cannot deduce any linkage between reports with the same $L(P_i)$, as different participants may have the same reputation level. Moreover, the server only knows which tasks the receipts are issued for, but cannot link a user to the contributed reports due to the partial blind signature.

Requirement 2. *A participant can neither use the receipts of task T_i to earn rewards from another task T_j nor use the receipts of a task multiple times to earn more rewards.*

Analytical validation. Recall that the receipts issued to a participant contain the identifiers committed to a specific task via partial blind signatures. Therefore, malicious participants cannot use the receipts obtained from another tasks T_j to earn rewards for task T_i. For each receipt $R_{\mathbb{R}_k}$ received, the server would invalidate its identifier α_k, so the participant can earn rewards from the task using his/her receipts only once.

Requirement 3. *A participant cannot forge receipts or steal other receipts to earn more rewards. The higher-quality data a participant submits, the more rewards he/she will earn. Meanwhile, The server cannot correlate the rewards with the reports submitted before.*

Analytical validation. Since each receipt issued is signed by the server, it is infeasible to forge a valid receipt. Some malicious participants may steal receipts to earn more rewards. However, they cannot submit the stolen receipts without extra pseudonyms. Although some may want to steal receipts with higher feedback level to replace their low-feedback receipts, they are faced with the risk of getting fewer rewards or being detected by the server. This is because the feedback is encrypted by pk_{ss}, other entities can neither distinguish the positive feedback from the negative feedback, nor tell which receipt has higher feedback. Even though higher-feedback receipts were usurped, the possible inconsistency of reputation level in submitted receipts will reveal his/her malicious behavior.

As shown in Eq. (2) and Eq. (3), a participant will earn more rewards if he/she submits higher-quality reports. Due to the adoption of partial blind signature, it is infeasible to associate the receipts with the corresponding data reports. Although the server can link rewards with the receipts but it cannot link them with the reports submitted before.

Requirement 4. *Unauthorized participants cannot forge and intercept other's pseudonyms to report data without being detected. Moreover, malicious participants can be evicted from a specific task or the whole system.*

Analytical validation. If an unauthorized participant P_i intercepts an approval message which is sent to another authorized participant $P_{i'}$, he/she cannot pass the check on $H(P_i|BT_j|n_{T_j}+1) = h_{i'}^3$, hence cannot obtain the valid pseudonyms. Since each pseudonym is signed by the TPA, malicious participants cannot forge a valid pseudonym. Therefore, we can ensure that all sensing reports received by the server are from the authorized participants.

As described in Section IV-G, with the cooperation of the TPA, malicious participants can be identified if he/she contributed low-quality data occasionally for a task. In this case, all pseudonyms of this malicious participant are revealed to the server, but the real identity keeps hidden since the only entity (i.e., TPA) who knows the relationship between participant pseudonyms and the real identity do not collude with the server. For the same reason, even though the server maliciously requests all pseudonyms of a certain user, it cannot infer the real identity of these pseudonyms, and cannot correlate pseudonyms used in different tasks. On the other hand, for very low-reputation participants, they will be evicted from the system with their identity revealed.

VI. Performance evaluation

A. Complexity analysis

Let M.M. and M.E. denote modular multiplication and modular exponentiation in $\mathbb{Z}_N$, respectively. GS/SIG and GS/VER denote the operations for group signature generation and verification, respectively. H is the hash computation. Hence, the user's computation cost due to task blinding and group signature generation is M.M.+M.E.+GS/SIG. If P_i is assigned a task with n reports requirement, he/she needs to obtain n partial blind signatures for receipts, which induces n(H+2M.M.+2M.E.) overhead. Therefore, for each authorized participant, the total computation cost incurred in our scheme is nH+(2n+1)(M.M.+M.E.)+GS/SIG per task. In comparison, [11] needs more computations due to the extra n partial blind signatures for report token generation.

At the server, it needs 1GS/VER computation, 2M.E. for RSA signature verification and M.M.+M.E. cost for blinding factor removal, respectively. If the server approves the task request, it needs to perform one hash operation and two RSA signatures (i.e., H+2M.E.) for the approval message. Next, nM.E. and 3nM.E. computation cost are incurred to verify $\{T_j|L(P_i)\}_{sk_{ss}}$ and derive n receipts. Correspondingly, the verification and decryption cost is n(2M.E.+H). Moreover, 3M.E. is required to derive U_{T_j}. Therefore, the server's computation cost is GS/VER+M.M.+(6n+8)M.E.+(n+1)H for an assigned task. In contract, more computations are required in [11] to verify the report token.

For the group manager, 2(H+M.E.) cost is required to derive τ_i^j. In the reputation update phase, the group manager needs to verify the signature and decrypt the feedback for authorized participants (2M.E.+H).

B. Implementation

Simulation setup. For the trust and reputation model, the same parameter setting as in [15] is used to assess the report trust. We assume that there are 100 participants, out of which there are 10 malicious participants in default. For simplicity, we consider similar reports have the maximum similarity 1 while opposite reports have the minimum similarity -1. We varied the reputation/trust threshold from 0.2 to 0.8, and the number of malicious participants from 10 to 60, in order to demonstrate the accuracy and robustness of trust and reputation model. In addition, we varied the number of reports n from 5 to 25 to show its impacts on the computation cost. All programs are implemented in Java on Andriod smartphone (Snapdragon 820) and a laptop (AMD Athlon M320).

Simulation results. To show the accuracy of our trust assessment and the impact of ε, we tested the rates of false positive FP and false negative FN with different thresholds, and the corresponding results are shown in Fig. 2. As we can see, when ε is small, the FP and FN rates are very low (approximately 0). As ε increases, the FP rate grows while the FN rate remains 0. This is reasonable as there is a higher possibility that a report is actually correct but its trust is less than a large ε. On the contrary, it is hardly possible that the trust of a false report is more than the large ε. When $\varepsilon = 0.5$, the FP rate is only 0.1 after users take 50 tasks and the corresponding FN rate is 0.

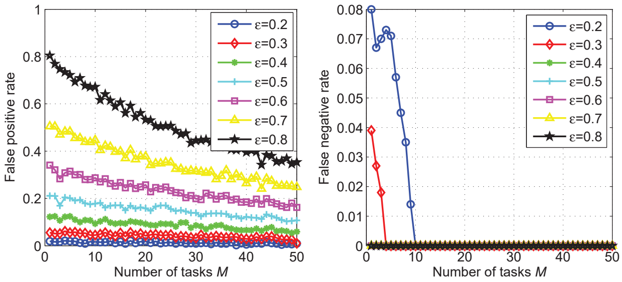

Fig. 2. The rates of false positive and false negative

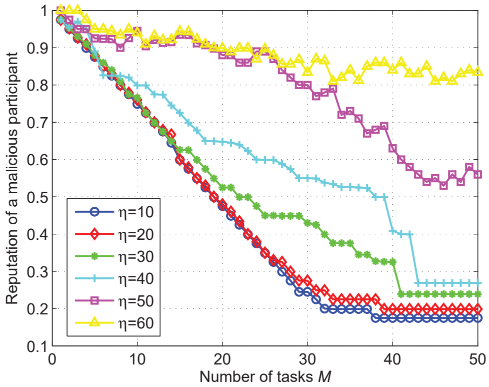

Fig. 3. Reputation of a malicious participant

Fig. 3 reports how a malicious user's reputation is changed with task quantity under different number of malicious users (η). Apparently, as more tasks are taken, the reputation of a malicious user drops down quickly and finally remains stable (close to 0) when a few malicious users exist. The reason is that the reports submitted by malicious users conflict with those from majority benign users. Correspondingly, it is highly possible that malicious users will get low report trust and negative feedback. With more malicious users, the reputation decreases more slowly, since more untrustworthy reports support each other. When more than 50% malicious users exist, untrustworthy reports may dominate and it results in that malicious users get high report trust and maintain a high reputation. Therefore, our scheme is robust to malicious participants as long as more benign participants are involved.

To study the practicality of our proposed scheme, Fig. 4 measures the computation cost in different phases at three entities. We find that most computations are performed at the server. The reason is that n partial blind signatures, n RSA signatures and encryptions are required by the server for a task. For participants, the time generating the blinded task ID is negligible in the registration phase. In contrast, the user's major computation time also focuses on the report submission phase, taking only about 440ms for 10 reports.

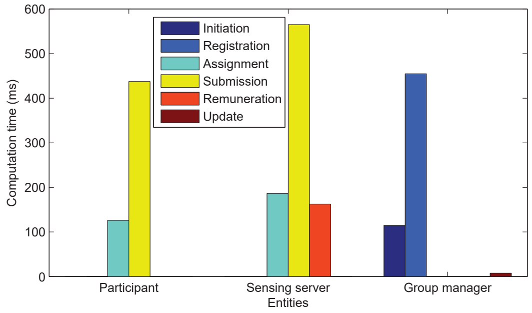

Fig. 4. The average running time of performing a task

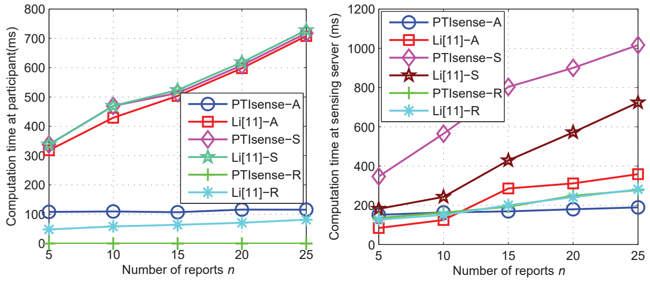

Fig. 5. The comparison of running time with varying n

To study the computation performance of our proposed scheme, we mainly compare PTISense with a state-of-art solution Li et. al [11]. The result is shown in Fig. 5 (A denotes assignment, S denotes report submission, and R denotes remuneration). In the assignment phase, we observe that the computation time of PTISense keeps stable with n for both entities, while that of [11] increases as n grows. This is because a certain number of group signature generations/verifications are conducted at both entities in PTISense, independent of n. Nevertheless, n partial blind signatures are generated for n report tokens. When submitting reports, PTISense requires comparable running time at user due to the similar cryptographic operations. However, our scheme takes the server more time due to the extra encryption of reputation feedback level, which is the cost of anonymous reputation management. In the remuneration phase, it clearly shows the superiority of PTISense at the participant (no cryptographic cost), while comparable cost is induced at the server for both schemes. Overall, PTISense can achieve privacy-aware sensing and incentive with less computation cost, especially at users.

VII. Conclusions

In this paper, we proposed PTISense to achieve privacy-aware and trustworthy mobile sensing with fair incentives. Based on the group signature, (partial) blind signature, we enable legitimate users to join tasks, contribute data, and earn rewards without any data linkability. Additionally, by limiting the number of pseudonyms issued by the TPA, greedy users are prevented from abusing the privacy-aware system. To further improve the data trustworthiness, we integrate the anonymous reputation management into the entire system, based on which a fair incentive scheme is elaborated to motivate user's reliable participation. Security analysis and prototype implementation demonstrate the security and efficiency of PTISense.

References

[1] M. Abe and E. Fujisaki, "How to date blind signatures," in *Advances in Cryptology-ASIACRYPT'96*. Springer, 1996, pp. 244–251.

[2] A. Ara, M. Al-Rodhaan, T. Yuan, 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. 12 601–12 617, 2017.

[3] J. Camenisch and J. Groth, "Group signatures: Better efficiency and new theoretical aspects." in *International Conference on Security in Communication Networks*, vol. 3352. Springer, 2004, pp. 120–133.

[4] D. Chaum, "Blind signature system," in *Advances in cryptology*. Springer, 1984, pp. 153–153.

[5] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," *Pervasive and mobile Computing*, vol. 9, no. 3, pp. 353–371, 2013.

[6] D. Förster, F. Kargl, and H. Löhr, "Puca: A pseudonym scheme with strong privacy guarantees for vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 37, pp. 122–132, 2016.

[7] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "Security, privacy, and incentive provision for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839–853, 2016.

[8] K. L. Huang, S. S. Kanhere, and W. Hu, "A privacy-preserving reputation system for participatory sensing," in *IEEE 37th Conference on Local Computer Networks (LCN)*. IEEE, 2012, pp. 10–18.

[9] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2016, pp. 344–353.

[10] Q. Li and G. Cao, "Providing privacy-aware incentives for mobile sensing," in *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2013, pp. 76–84.

[11] ——, "Providing privacy-aware incentives in mobile sensing systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1485–1498, 2016.

[12] Z. Luo and X. Huang, "A personalized k-anonymity with fake position generation for location privacy protection," in *Internet Conference of China*. Springer, 2014, pp. 46–55.

[13] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "Anonysense: A system for anonymous opportunistic sensing," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, 2011.

[14] J. Son, D. Kim, R. Hussain, A. Tokuta, S.-S. Kwon, and J.-T. Seo, "Privacy aware incentive mechanism to collect mobile data while preventing duplication," in *Military Communications Conference*. IEEE, 2015, pp. 1242–1247.

[15] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2777–2790, 2014.

[16] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *MOBICOM'2012: International Conference on Mobile Computing and Networking*, 2012, pp. 173–184.

[17] J. Zhang, L. He, Q. Zhang, and Y. Gan, "Pseudonym-based privacy protection scheme for participatory sensing with incentives." *KSII Transactions on Internet & Information Systems*, vol. 10, no. 11, 2016.

[18] J. Zhang, J. Ma, W. Wang, and Y. Liu, "A novel privacy protection scheme for participatory sensing with incentives," in *IEEE 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS)*, vol. 3. IEEE, 2012, pp. 1017–1021.

[19] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2830–2838.