CROWDFA: A Privacy-Preserving Mobile Crowdsensing Paradigm via Federated Analytics

Bowen Zhao, *Member, IEEE*, Xiaoguo Li, Ximeng Liu, *Senior Member, IEEE*, Qingqi Pei, *Senior Member, IEEE*, Yingjiu Li, *Member, IEEE*, and Robert H. Deng, *Fellow, IEEE*

***Abstract*— Mobile crowdsensing (MCS) systems typically struggle to address the challenge of data aggregation, incentive design, and privacy protection, simultaneously. However, existing solutions usually focus on one or, at most, two of these issues. To this end, this paper presents CROWDFA, a novel paradigm for privacy-preserving MCS through federated analytics (FA), which aims to achieve a well-rounded solution encompassing data aggregation, incentive design, and privacy protection. Specifically, inspired by FA, CROWDFA initiates an MCS computing paradigm that enables data aggregation and incentive design. Participants can perform aggregation operations on their local data, facilitated by CROWDFA, which supports various common data aggregation operations and bidding incentives. To address privacy concerns, CROWDFA relies solely on an efficient cryptographic primitive known as additive secret sharing to simultaneously achieve privacy-preserving data aggregation and privacy-preserving incentive. To instantiate CROWDFA, this paper presents a privacy-preserving data aggregation scheme (PRADA) based on CROWDFA, capable of supporting a range of data aggregation operations. Additionally, a CROWDFA-based privacy-preserving incentive mechanism (PRAED) is designed to ensure truthful and fair incentives for each participant, while maximizing their individual rewards. Theoretical analysis and experimental evaluations demonstrate that CROWDFA protects participants' data and bid privacy while effectively aggregating sensing data. Notably, CROWDFA outperforms state-of-the-art approaches by achieving up to 22 times faster computation time.**

***Index Terms*— Crowdsensing, privacy protection, data aggregation, reward distribution, federated analytics.**

Manuscript received 19 December 2022; revised 18 July 2023; accepted 20 August 2023. Date of publication 25 August 2023; date of current version 5 September 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3102700; in part by the National Natural Science Foundation of China under Grant 62202358, Grant 62072109, Grant U1804263, Grant 61702105, and Grant 61632013; in part by the Key Research and Development Programs of Shaanxi under Grant 2021ZDLGY06-03; and in part by the China Post-Doctoral Science Foundation under Grant 2023TQ0258. The work of Yingjiu Li was supported in part by the Ripple University Blockchain Research Initiative. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Grigorios Loukides. *(Corresponding authors: Ximeng Liu; Bowen Zhao.)*

Bowen Zhao is with the Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China (e-mail: bwinzhao@gmail.com).

Xiaoguo Li and Robert H. Deng are with the School of Computing and Information Systems, Singapore Management University, Singapore 188065 (e-mail: xiaoguoli@smu.edu.sg; robertdeng@smu.edu.sg).

Ximeng Liu is with the College of Computer and Data Science, Fuzhou University, Fujian 350025, China (e-mail: snbnix@gmail.com).

Qingqi Pei is with the State Key Laboratory of Integrated Service Networks, the Shaanxi Key Laboratory of Blockchain and Secure Computing, and the Shaanxi Engineering Research Center of Trusted Digital Economy, Xidian University, Shaanxi, Xi'an 710071, China (e-mail: qqpei@mail.xidian.edu.cn).

Yingjiu Li is with the Department of Computer and Information Science, University of Oregon, Eugene, OR 97403 USA.

Digital Object Identifier 10.1109/TIFS.2023.3308714

I. INTRODUCTION

MOBILE crowdsensing (MCS) systems [1], [2] outsource sensing tasks to participants instead of deploying dedicated sensors, and then aggregate participants' collected sensing data. MCS systems enjoy low deployment costs and enhanced flexibility, and take full advantage of the potential of crowds. These systems have found widespread applications in various fields such as healthcare, environment protection, indoor localization, refined urban management, event sensing, and more [3], [4], [5].

Incentive design, data aggregation, and privacy protection are critical requirements for MCS systems. First, incentive design motivates whether participants take part in crowdsensing or not [6], [7]. Firstly, incentive design plays a crucial role in motivating participants to take part in crowdsensing activities [6], [7]. As participants collect sensing data and contribute to data aggregation, they consume valuable resources such as computation and communication. It is essential to provide incentives that compensate for participants' resource consumption [8]. Secondly, the primary purpose of MCS systems is to derive meaningful conclusions from the collected sensing data. Therefore, common data aggregation (e.g., sum, mean, and variance) becomes necessary for generating reasoning results based on participants' sensing data [5], [9]. Lastly, privacy concerns act as obstacles that impede participants' willingness to engage in MCS [10], [11], [12], [13]. Even if participants are offered incentives, they may decline to participate due to apprehensions about their privacy being compromised, such as the exposure of sensitive information like location, bidding activities, and healthcare status.

No existing solutions have simultaneously addressed the issues of incentive design, data aggregation, and privacy protection. Common incentive mechanisms include auction/bidding [14], [15] and posting awards [16], [17]. Typical privacy-preserving incentive mechanisms are either encrypted-based, or differential privacy (DP)-based [18], [19], [20], [21]. Encryption-based privacy-preserving incentive mechanisms [20], [21] rely on a trusted sensing platform (SP), while DP-based privacy-preserving incentive mechanisms [18], [19] assume an honest-but-curious SP. However, existing incentive mechanisms, regardless of whether encryption-based or DP-based, fail to enable common data aggregation operations including sum, mean, and variance.

On the other hand, data aggregation and privacy concerns are addressed in privacy-preserving data aggregation, where

SP is considered as honest-but-curious. Such solutions typically enable privacy-preserving data aggregation [15], [22], [23], [24], [25], [26]; however, it remains challenging to select a winner in bidding when bids are encrypted in these solutions. Previous solutions [15], [26] support privacy-preserving data aggregation for crowdsensing; however, the data aggregation operations are restricted to sum only, and they fail to protect bid privacy [27]. No previous solutions for MCS meet all requirements on incentive design, data aggregation, and privacy protection simultaneously due to various technical challenges.

The first challenge is to design a unified framework or paradigm that achieves privacy-preserving data aggregation and privacy-preserving incentive design, simultaneously. Combining a privacy-preserving incentive mechanism and a privacy-preserving data aggregation solution in an MCS system may result in both high system complexity and an increased risk of privacy leakage. *The second challenge is to enable accurate and efficient privacy-preserving data aggregation*. DP-based data aggregation introduces noise to aggregated results, which can lead to erroneous aggregation [28] and restrict aggregation operations. Homomorphic encryption-based data aggregation schemes suffer from high computation costs and increased communication traffic overheads due to expensive homomorphic operations and expanded ciphertext sizes [23]. *The third challenge is to achieve truthfulness and fairness in privacy-preserving incentive design*. Truthfulness, which prevents participants from submitting deviated bids to improve rewards without knowledge of others' bids [14], and fairness, where each participant's reward is positively related to their contributions [17], are critical objectives for an incentive mechanism. DP-based incentive mechanisms only guarantee degraded truthfulness rather than full truthfulness [18], [19], while encryption-based incentive mechanisms are weak in achieving truthfulness [20], [21]. Moreover, auction-based incentive mechanisms usually disregard fairness.

To tackle the aforementioned challenges, this paper introduces a novel approach called CRWODFA,[1] a privacy-preserving mobile crowdsensing paradigm through federated analytics (FA). FA, a collaborative computing framework, enables the extraction of insights from raw data distributed across multiple decentralized devices [29]. By integrating FA into MCS, CROWDFA empowers participants to perform operations on their local data and submit only the aggregated results, minimizing communication traffic and reducing the burden of data aggregation on SP. To ensure the confidentiality of raw sensing data, aggregated results, and bids, CROWDFA exclusively employs an efficient cryptographic primitive known as additive secret sharing. This primitive supports computations over encrypted data without introducing any noise. In summary, our contributions are three main aspects.

- We propose CROWDFA, a novel MCS paradigm based on FA. To the best of our knowledge, CROWDFA is the first work seamlessly integrating FA and MCS, achieving privacy-preserving data aggregation, and privacy-preserving incentive design.
- We present PRADA, a CROWDFA-based privacy-preserving data aggregation scheme that facilitates a range of privacy-preserving data aggregation operations, such as sum, mean, and variance. PRADA enables participants to perform local aggregation operations while safeguarding the privacy of their raw sensing data. Additionally, PRADA supports aggregation operations like p-order moment, skewness, and kurtosis. Compared to prior solutions, PRADA significantly reduces computing time, requiring up to 22 times less computation time.
- We design PRAED, a CROWDFA-based privacy-preserving incentive mechanism that aims to protect the privacy of each participant's bids. PRAED not only ensures truthfulness and fairness but also maximizes each participant's rewards.

[1] CROWDFA comes from privacy-preserving **Crowd**sensing paradigm via **F**ederated **A**nlaytics.

TABLE I
COMPARISONS BETWEEN CROWDFAAND RELATED WORK

Schemes	Incentive	Data aggregation/Local†	Privacy	Noise-free‡
[26]	●	S./○	◐	○
[18]	●	○/○	◐	○
[30]	●	S./○	●	○
[25]	○	S., M., V./○	●	●
CROWDFA	●	S., M., V., etc./●	●	●

Note. ●=sayisfaction, ◐=Partial sayisfaction, ○=Dissayisfaction; S., M., and V., denote sum, mean, and variance, respectively; † indicates a participant performs aggregation operation locally, while a sensing platform only aggregates local aggregation results; ‡ signifies that there is no noise present in the aggregation result or bids

The rest of this paper is organized as follows. In Section II, we describe the related work. In Section III, we formulate CROWDFA with its system model and threat mode. Section IV and Section V elaborate on the design of PRADA and PRAED, respectively. Results of theoretical analysis and experimental evaluation are given in Section VI. Finally, we conclude this paper in Section VII.

II. RELATED WORK

A. Privacy-Preserving Data Aggregation

Li et al. [24] put forward to privacy-preserving data aggregation in mobile sensing and designed a sum aggregation protocol based on additive homomorphic encryption. To tackle the privacy concerns of data aggregation in sensing systems, Zhang et al. [31] presented a novel peer-to-peer framework to enable privacy-preserving data aggregation, including sum, average, variance, etc. Zhuo et al. [23] proposed a cloud-assisted three-party architecture and adopted BGV homomorphic encryption to support privacy-preserving sum, average, and variance computations. Vakilinia et al. [22] proposed a solution that allows an untrusted server to aggregate the sums of participants' sensing data in a privacy-preserving manner based on linear transformation and

homomorphic encryption. Wu et al. [32] introduced a fog-assisted architecture and two-trapdoor Pailliery cryptosystem to enable privacy-preserving aggregations, such as sum, mean, and variance. Yan et al. [25] utilized additive secret sharing to protect the privacy of sensing data and support sum, mean, and variance operations on sensing data. To the best of our knowledge, almost all existing schemes require participants to encrypt or obfuscate sensing data item by item and transmit encrypted or obfuscated sensing data to one or multiple servers for performing aggregation operations. Consequently, these schemes fail to take advantage of participants' computation capability over raw sensing data to speed up the aggregation.

B. Privacy-Preserving Incentive Mechanism

To protect bid privacy, Sun and Ma [20] proposed a solution that utilizes homomorphic encryption to enable participants to encrypt their bids. This approach allows auctions to be conducted over encrypted bids. Similarly, Wang et al. [21] also presented a solution aimed at safeguarding bid privacy and auctions through the use of encrypted bids. In contrast to previous encryption methods, Jin et al. [18] employed differential privacy to ensure bid privacy while enabling approximate truthfulness in incentive design. Subsequently, Wang et al. [19] developed an honest-but-curious sensing platform, where they designed a privacy-preserving and approximately truthful incentive mechanism using differential privacy. Zhang et al. [30] also proposed a privacy-preserving and truthful incentive mechanism based on differential privacy. However, the majority of prior research on privacy-preserving incentive mechanisms primarily focuses on protecting bid privacy, and largely neglects the purpose of data aggregation for crowdsensing. The question remains as to how to achieve privacy-preserving data aggregations effectively using such incentive mechanisms.

As shown in Table I, CROWDFA achieves a balanced solution among data aggregation, incentive design, and privacy protection, setting it apart from existing solutions. Additionally, CROWDFA empowers participants to perform aggregation operations directly on their locally raw sensing data, resulting in accelerated data aggregation and reduced overhead. Moreover, CROWDFA ensures aggregation results and bids remain unaffected by any noise.

III. Problem Formulation

A. System Model

As shown in Fig. 1, scCrowdFA comprises a sensing platform (SP) and multiple participants.

- **SP**: SP recruits multiple participants and collaborates with them to produce aggregation results. Additionally, SP implements an incentive mechanism that relies on participants' bids to allocate rewards to each participant.
- **Participant**: Each participant collects sensing data and performs aggregation operations with locally raw sensing data. In particular, each participant adopts a privacy-preserving mechanism (e.g., additive secret sharing) to protect data privacy, such as sensing data and bids.

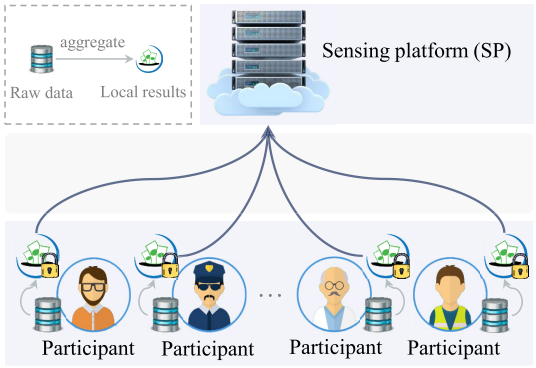


Fig. 1. System model of CrowdFA.

Also, CROWDFA allows SP to dynamically selects some participants as leader assisting in data aggregation.

Noted. Certain participants being regarded as leaders is a common assumption made in privacy-preserving data aggregation [33], [34], [35] and MCS [36], [37], [38]. In these scenarios, the leading participant plays a crucial role by assisting a central server (e.g., SP) in aggregating other participants' data.

B. Problem Statement

CROWDFA is essentially a secure multi-party computation problem [39]. Multiple participants and SP take as input their own data and jointly output results, where each participant's input is private data. Formally, CROWDFA $f(x_1, \cdots, x_n, \eta)$ is defined as follows.

$$f(x_1, \cdots, x_n, \eta) \triangleq \mathcal{G}\left(\mathcal{L}_\eta(x_1), \mathcal{L}_\eta(x_2), \cdots, \mathcal{L}_\eta(x_n)\right), \quad (1)$$

where η represents the input of SP, x_i denotes the participant i's input, $\mathcal{L}_\eta(\cdot)$ is a local operation performed by each participant, and $\mathcal{G}(\cdot)$ is a global operation based on the outputs of all participants. In fact, the key point of CROWDFA solving an MCS task lies in the design of both the local operation and the global operation. This paper considers the following two types of MCS tasks: data aggregation and incentive design.

1) Data Aggregation: In terms of MCS systems, data aggregation plays a critical role in drawing conclusions from participants' sensing data [22], [30]. Thus, data aggregation is regarded as a fundamental task in MCS systems. Without loss of generality, let $\Gamma(d)$ represent an aggregation function that takes a dataset d as input. In the context of data aggregation, both local and global operations aim to execute the Γ function. During this process, SP does not provide any input (namely $\eta = \perp$), while each participant contributes his dataset d_i (also denoted as x_i in Eq. (1)). Specifically, 1) each participant i employs Γ (also referred to as $\mathcal{L}_\eta$ in Eq. (1)) to generate a local aggregation result, denoted as $\Gamma(d_i)$; and 2) then a privacy-preserving interactive protocol is performed between SP and participants, utilizing the Γ function that takes as input all participants' local aggregation results. Formally, a CROWDFA-based data aggregation $f(d_1, \cdots, d_n)$ is formulated as follows

$$f(d_1, \cdots, d_n) \triangleq \Gamma\left(\Gamma(d_1), \Gamma(d_2), \cdots, \Gamma(d_n)\right). \quad (2)$$

In this case, *the challenge is how to design a privacy-preserving and efficient interactive protocol*.

2) Incentive Design: Incentive design plays another fundamental role in MCS systems. In terms of MCS systems, incentive mechanisms offer an elegant and practical method to promote the collection of sensing data [18], [26]. In general, the incentive mechanism requires each participant to submit a bid b_i, while SP prepares a reward budget B. Consequently, designing a privacy-preserving incentive mechanism for MCS systems essentially becomes a secure multi-party computation problem. Each participant and SP input their private bids and the budget, respectively, resulting in each participant receiving a reward. Although privacy-preserving incentive design is significantly different from privacy-preserving data aggregation, we also formulate it based on the CROWDFA framework. To be specific, a CROWDFA-based incentive mechanism $f(b_1, \cdots, b_n, B)$ is formulated as follows

$$f(b_1, \cdots, b_n, B) \triangleq \mathcal{G}\left(\mathcal{L}_{B,\sigma}(b_1), \mathcal{L}_{B,\sigma}(b_2), \cdots, \mathcal{L}_{B,\sigma}(b_n)\right), \tag{3}$$

where $\mathcal{L}_{B,\sigma}$ represents a local operation that calculates the reward u_i for each participant. The function $\mathcal{G}$ maps these rewards to an output vector, denoted as $\mathcal{G}(u_1, u_2, \cdots, u_n) = [u_1, u_2, \cdots, u_n]$. Formally, $\mathcal{L}_{B,\sigma}$ can be denoted by

$$\mathcal{L}_{B,\sigma}(b) \triangleq \sigma(i) \cdot v + b \cdot u, \tag{4}$$

where (u, v) represents a reward base, while $\sigma(i)$ is a sign function related to each participant i ($i \in \{1, \cdots, n\}$). In this case, *the challenge lies in designing a privacy-preserving and efficient interactive protocol that can generate parameters (including u, v, and $\sigma(i)$) within the scope of $\mathcal{L}_{B,\sigma}$*. Given these parameters, each participant easily calculates her reward as $u_i = \mathcal{L}_{B,\sigma}(b_i)$.

Note that we use d to represent a sensing dataset, while $\mathtt{d}$ is used to denote the size of d. Furthermore, b indicates a bid. Each participant is numbered by an integer i ($1 \leq i \leq n$), where n represents the total number of participants.

C. Threat Model

In our system, both SP and any participant (including the leading participants) are regarded as *semi-honest* (a.k.a *honest-but-curious*). This means that SP and each participant honestly perform protocols but intend to learn others' data. Consequently, there are two types of adversaries: SP and the participant. SP attempts to obtain the participant's raw sensing data and bids, while each participant attempts to acquire SP's aggregation results or other participants' bids. It is assumed no collusion between SP and any participant (including the leading participants). This assumption is commonly made in a multiparty computation paradigm [39], [40] and MCS systems [18], [19], [23], [24], [41]. The reasoning behind this assumption is that either SP or the leading participant lacks motivation to collude with each other. The main objectives of CROWDFA are two-fold: 1) prevent SP from learning any raw sensing data and bids, and 2) prevent any participant (including the leading participants) from learning any aggregation results and other participants' raw sensing data and bids.

To capture the threat model mentioned above, we formally define a security model for CROWDFA. Let us commence with the following notations. $\{f_i\}_{i=1}^n$ and f represent probabilistic functionalities. π denotes a CROWDFA protocol utilized for computing $f(d_1, \cdots, d_n, \eta)$. During the execution of π on $f(d_1, \cdots, d_n, \eta)$, $\mathcal{V}_i^\pi$ and $\mathcal{V}_{\text{SP}}^\pi$ refers the participant i's view and SP's view, respectively. Additionally, the output of the i-th participant is denoted by $\mathsf{output}_i^\pi(d_1, \cdots, d_n, \eta)$, while the output of SP is denoted by $\mathsf{output}_{\text{SP}}^\pi(d_1, \cdots, d_n)$. We denote the joint output of all involved parties as

$$\mathsf{output}^\pi(d_1, \cdots, d_n, \eta) \triangleq \{\mathsf{output}_{\text{SP}}^\pi; \mathsf{output}_1^\pi, \cdots, \mathsf{output}_n^\pi\}.$$

Let $y = f(d_1, \cdots, d_n, \eta)$. We will now define the formal security for CROWDFA as follows.

Definition 1 (Security for CROWDFA*): We say that π securely computing f if there exist probabilistic polynomial time (P.P.T.) simulators $\mathcal{S}$ and $\mathcal{S}_i$ such that for any dataset $(d_1, \cdots, d_n, \eta)$, the following conditions*

$$\{\mathcal{S}(y), y\} \overset{c}{\equiv} \{\mathcal{V}_{SP}^\pi, \mathit{output}^\pi(d_1, \cdots, d_n, \eta)\} \tag{5}$$

and

$$\{\mathcal{S}_i(d_i, f_i(d_i)), y\} \overset{c}{\equiv} \{\mathcal{V}_i^\pi, \mathit{output}^\pi(d_1, \cdots, d_n, \eta)\} \tag{6}$$

hold. Here, $\overset{c}{\equiv}$ represents computationally indistinguishable. Intuitively, Eq. (5) captures the limitation of a semi-honest SP in learning any participant's data, while Eq. (6) implies any semi-honest participant cannot learn other participant's data.

However, the security model defined in Definition 1 only considers one party being corrupted in CROWDFA. To account for more powerful attacks, we introduce another security model, namely t-collusion resistance, for CROWDFA. In this model, we accommodate multiple participants colluding with each other.

Definition 2 (t-Collusion Resistance for CROWDFA*): We say that π is t-collusion if there exists a P.P.T. simulator $\mathcal{S}_I$ such that for any dataset $(d_1, \cdots, d_n, \eta)$, $I \subset \{1, \cdots, n\}$ and $|I| \leq t$, the following condition holds*:

$$\begin{aligned}&\{\mathcal{S}_I(\{d_i\}_{i\in I}, \{f_i(d_i)\}_{i\in I}), f(d_1, \cdots, d_n, \eta)\}\\ &\overset{c}{\equiv} \{\{\mathcal{V}_i^\pi\}_{i\in I}, \mathit{output}^\pi(d_1, \cdots, d_n, \eta)\}.\end{aligned} \tag{7}$$

Intuitively, Eq. (7) implies that collusion among t participants cannot yield additional advantages for corrupting other participants' data. This paper introduces PRADA, which can withstand collusion attacks involving up to $n-1$ participants.

IV. PRADA DESIGN

In this section, we first provide a brief description of additive secret sharing. After that, we elaborate on PRADA,[2] which is a privacy-preserving data aggregation scheme based on CROWDFA. PRADA offers support for common privacy-preserving data aggregation operations, including sum, mean, and variance.

[2]PRADA: PRivAcy-preserving Data Aggregation.

A. Additive Secret Sharing

An additive secret sharing scheme consists of a sharing function S and a recovering function R. S takes as input an origin secret x and outputs two secret shares. $\mathsf{S}(x)$ is formulated as

$$\mathsf{S}(x) \rightarrow ([x]_1, [x]_2), \tag{8}$$

where $[\cdot]_j$ denotes one secret share ($j = 1$ or 2). R takes as input two secret shares and outputs the origin secret. $\mathsf{R}([x]_1, [x]_2)$ is formulated as

$$\mathsf{R}([x]_1, [x]_2) \rightarrow x. \tag{9}$$

In practice, given the origin secret x, $\mathsf{S}(x)$ usually sets $[x]_2 = r$ and $[x]_1 = x - r \mod N$, where r is randomly chosen from $\mathbb{Z}_N = \{0, 1, \cdots, N-1\}$, and N is usually a larger integer, such as $N = 2^\lambda$. Given two secret shares $[x]_1$ and $[x]_2$, R recovers the origin secret by computing $[x]_1 + [x]_2 \mod N$.

Considering practical MCS applications, sensing data may be floating-point numbers. One common approach is to convert floating-point numbers to integers [32], [41]. This paper converts the floating-point number into an integer by using the formula $\lceil x \rfloor \leftarrow x \cdot 2^\ell$, where 2^ℓ is chosen to be sufficiently large to ensure that $x \cdot 2^\ell$ is an integer, and $2^\ell \ll N$. Particularly, this paper sets $\ell = 32$ and assumes the bit length of decimal places in a floating point number is less than 32 bits. For simplicity, we will henceforth uses x instead of $\lceil x \rfloor$ when x refers to a floating-point number.

B. Formulation of PRADA

Technically, PRADA instantiates both $\mathcal{G}$ and $\mathcal{L}_\eta$ in Eq. (1) as Γ-function and Γ-representable function supporting data aggregation operations. Γ-function and Γ-representable function are defined as follows. Γ-function is over $d \subset \mathcal{D}$, where d and $\mathcal{D}$ are the function's input and range, respectively.

Definition 3 (Γ-Function): A aggregation function is called Γ-function if for any $d_1, d_2 \subset \mathcal{D}$, we have that

$$\Gamma(d_1, d_2) = \Gamma(\Gamma(d_1), \Gamma(d_2)) \tag{10}$$

holds, in which d_1, d_2 denotes the concatenation between d_1 and d_2.

Let $d_1 = \{3, 1, 7\}$ and $d_2 = \{3, 5\}$ be two pieces of raw sensing data from two parties, and the concatenation between d_1 and d_2 means that $(d_1, d_2) = \{3, 1, 7, 3, 5\}$. Then we explain Γ-function with a few simple examples.

Sum is a Γ-function. *Sum* is defined as $\Gamma(d) \triangleq \sum_{j=1}^{\mathtt{d}} x_j$, where $\mathtt{d}$ denotes the count of $d = \{x_1, \cdots, x_\mathtt{d}\}$. In the above example, $\Gamma(d_1) = 11$, $\Gamma(d_2) = 8$, and $\Gamma(d_1, d_2) = \Gamma(\Gamma(d_1), \Gamma(d_2)) = 19$. Therefore, *Sum* is a Γ-function for any $d_1, d_2 \subset \mathbb{Z}$.

Product is a Γ-function. *Product* is defined as $\Gamma(d) \triangleq \Pi_{j=1}^{\mathtt{d}} x_j$, where $\mathtt{d}$ denotes the count of $d = \{x_1, \cdots, x_\mathtt{d}\}$. In the above example, $\Gamma(d_1) = 21$, $\Gamma(d_2) = 15$, and $\Gamma(d_1, d_2) = \Gamma(\Gamma(d_1), \Gamma(d_2)) = 315$. Therefore, *Product* is a Γ-function for any $d_1, d_2 \subset \mathbb{Z}$.

Sum on a transformation function $\phi(\cdot)$ is a Γ-function. In this ca-se, the aggregation function is defined as $\Gamma(d) \triangleq \sum_{j=1}^{\mathtt{d}} \phi(x_j)$, where $d = \{x_1, \cdots, x_\mathtt{d}\}$. If $\phi(x) = x^2$, then we have $\Gamma(d) \triangleq \sum_{j=1}^{\mathtt{d}} (x_j)^2$. In the above example, we have $\Gamma(d_1) = 59$, $\Gamma(d_2) = 34$, $\Gamma(d_1, d_2) = \Gamma(\Gamma(d_1), \Gamma(d_2)) = 93$. Also, when $\phi(\cdot)$ is defined as a count function, i.e., $\phi(d) = \mathtt{d}$, we have *Count* is a Γ-function. Arguably, $\phi(x)$ can be any function about the only variable x.

Mean is *not* a Γ-function. In this case, the aggregation function is defined as $\Gamma(d) \triangleq \sum_{j=1}^{\mathtt{d}} \frac{x_j}{\mathtt{d}}$, where $d = \{x_1, \cdots, x_\mathtt{d}\}$. In the above example, $\Gamma(d_1) = 11/3$, $\Gamma(d_2) = 4$, and $\Gamma(d_1, d_2) \neq \Gamma(\Gamma(d_1), \Gamma(d_2))$. One intuitive explanation is $\frac{x_j}{\mathtt{d}}$ involves two variables.

It is also easy to verify that *Maximum* are also Γ functions. In the above example, $\max(\max(3, 1, 7), \max(3, 5)) = 7$) and $\min(\min(3, 1, 7), \min(3, 5)) = 1$. To capture more practical aggregation functions (e.g. mean, and variance), we introduce a family for $\Gamma function$, called Γ-representable function.

Definition 4 (Γ-Representable Function): A aggregation function Θ is called Γ-representable function if there exists a finite set $\{\Gamma_1, \Gamma_2, \cdots, \Gamma_m\}$ such that

- *for any $i \in \{1, 2, \cdots, m\}$, Γ_i is a Γ-function;*
- *Θ is an arithmetic combination from these Γ-functions.*

From Definition 4, it is evident that the mean can be formulated using two Γ-functions: the Γ-function of *sum* and the Γ-function of *count*. On the other hand, the variance can be formulated as three Γ-funcitons: the Γ-function of *sum*, the Γ-function of *count*, and the Γ-function of *sum* on $\phi(x) = x^2$. Additionally, various other aggregation operations such as standard deviation, p-order moment, skewness, and kurtosis can be achieved through Γ functions. One of the key innovations of this paper is the integration of multiple types of data aggregation operations within a unified computing paradigm.

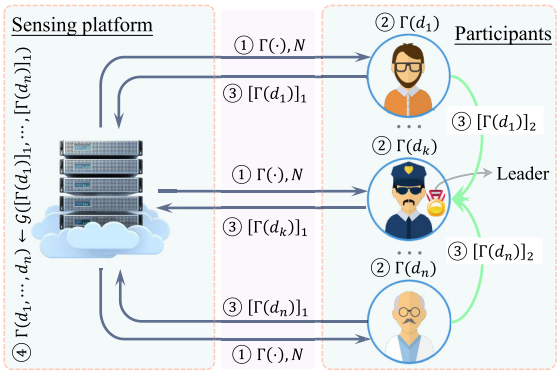


Fig. 2. Workflow of PARADA.

Fig. 2 shows a workflow for PRADA, which comprises four steps: *initialization*, *federated computation*, *transmission*, and *aggregation*. Assume that SP recruits n participants to collect sensing data. Each participant is indexed by i, where $i \in [1..n]$. For simplicity, the rest of this paper uses $[1..n]$ to represent the set $\{1, 2, \cdots, n\}$. The detailed workflow of PRADA is presented as follows.

- **Initialization**: SP initializes PRADA. In this process, SP recruits n participants and randomly selects one participant (e.g., the participant k) as a leader from n participants. To begin, SP initializes the parameter N

(e.g., $N = 2^{62}$) for additive secret sharing, along with a Γ-function. Afterward, SP broadcasts $\Gamma(\cdot)$ and N to all participants.

- **Federated computation**: Each participant collects raw sensing data and locally performs the Γ-function on their own raw sensing data. Specifically, each participant i computes $\Gamma(d_i)$ ($i \in [1..n]$). d_i represents the participant i's local sensing dataset.
- **Transmission**: Except for the participant k, each participant i ($i \in [1..n] \setminus k$) secretly shares (or say encrypts) their local aggregation result $\Gamma(d_i)$. Formally, the participant i invokes $\mathsf{S}(\cdot)$ to secretly share $\Gamma_i(d_i)$, i.e.,

$$\mathsf{S}(\Gamma(d_i)) \rightarrow ([\Gamma(d_i)]_1, [\Gamma(d_i)]_2), \tag{11}$$

and transmits $[\Gamma(d_i)]_1$ and $[\Gamma(d_i)]_2$ to SP and the participant k (i.e., the leading participant), respectively. After receiving other participants' shares, the participant k computes

$$\begin{cases} [\Gamma(d_k)]_2 \leftarrow \sum_{i=1, i\neq k}^{n} [\Gamma(d_i)]_2 \mod N, \\ [\Gamma(d_k)]_1 \leftarrow \Gamma_k(d_k) + [\Gamma(d_k)]_2 \mod N. \end{cases} \tag{12}$$

Next, the participant k transmits $[\Gamma(d_k)]_1$ to SP.

- **Aggregation**: SP computes the final aggregation result by aggregating all participants' secret shares of local aggregation results, i.e.,

$$\Gamma(d_1, \cdots, d_n) \leftarrow \Gamma([\Gamma(d_1)]_1, \cdots, [\Gamma(d_n)]_1). \tag{13}$$

As shown in Fig. 2, we observe that each participant generates a local aggregation result (see **Federated computation**), while SP only aggregates all participants' local results (see **Aggregation**). Therefore, PRADA significantly reduces the communication and computing overhead for both the participant and SP. Compared to existing schemes [25], [26], [32] that require SP to directly aggregate all participants' data instead of their local aggregation results, PRADA outperforms them in terms of communication and computation costs. Notably, PRADA adopts additive secret sharing to to compute $\Gamma(\Gamma(d_1), \cdots, \Gamma(d_n))$, enabling a privacy-preserving and efficient interactive protocol among participants and SP.

C. Implementation of PRADA

This section provides three privacy-preserving data aggregation protocols of PRADA, using sum, mean, and variance as examples. Note that other aggregation operations of PRADA can be implemented by following the design of privacy-preserving sum aggregation (`SumAgg`), privacy-preserving mean aggregation (`MenAgg`), and privacy-preserving variance aggregation (`VarAgg`).

Assume that the participant i ($i \in [1..n]$) collects a sensing dataset $d_i = \{x_{i,1}, \cdots, x_{i,\mathtt{d}_i}\}$, where $\mathtt{d}_i$ represents the size of d_i, and $x_{i,j}$ denotes the j-th raw sensing data item of the participant i. Without loss of generality, the participant k is selected as the leader. $\xleftarrow{\$}$ signifies a random selection operation. `SumAgg`, `MenAgg`, and `VarAgg` are detailed as follows.

Algorithm 1 $\mathtt{SumAgg}(d_1, \cdots, d_n) \rightarrow \sum_{i=1}^{n} \sum_{j=1}^{\mathtt{d}_i} x_{i,j}$

Input: n participants have $\{d_1, \cdots, d_n\}$.
Output: SP obtains $\sum_{i=1}^{n} \sum_{j=1}^{\mathtt{d}_i} x_{i,j}$.
Procedure:
Initialization (@SP):
- Recruit n participants and randomly select the participant k as a leader;
- Initialize $\Gamma(\cdot)$ as the sum aggregation operation and N;
- Broadcast $\Gamma(\cdot)$ and N to all participants.

Federated computation & Secret sharing (@Participant):
- The participant i ($i \in [1..n] \setminus k$) computes

$$\begin{cases} \Gamma(d_i) \leftarrow \sum_{j=1}^{\mathtt{d}_i} x_{i,j}, \\ [\Gamma(d_i)]_2 \xleftarrow{\$} \mathbb{Z}_N, \\ [\Gamma(d_i)]_1 \leftarrow \Gamma(d_i) - [\Gamma(d_i)]_2 \mod N, \end{cases}$$

and then transmits $[\Gamma(d_i)]_1$ and $[\Gamma(d_i)]_2$ to SP and the leader, respectively;
- The leader firstly computes $\Gamma(d_k) \leftarrow \sum_{j=1}^{\mathtt{d}_k} x_{k,j}$, and then performs Eq. (12) and transmits $[\Gamma(d_k)]_1$ to SP.

Aggregation (@SP):
- Compute $\sum_{i=0}^{n} [\Gamma(d_i)]_1 \mod N$ to output the aggregation result.

*1) **SumAgg**:* Given $\{d_1, \cdots, d_n\}$, `SumAgg` outputs $\sum_{i=1}^{n} \sum_{j=1}^{\mathtt{d}_i} x_{i,j}$. As the sum function is a Γ-function, `SumAgg` follows the workflow of PRADA. Technically, `SumAgg` is formulated as $\Gamma(d_i) \triangleq \sum_{j=1}^{\mathtt{d}_i} x_{i,j}$, and then $\mathtt{SumAgg}(d_1, \cdots, d_n)$ is denoted by

$$\mathtt{SumAgg}(d_1, \cdots, d_n) \equiv \sum_{i=0}^{n} [\Gamma(d_i)]_1 \mod N. \tag{14}$$

As depicted in Fig. 1, `SumAgg` compromises four steps, where the second and the third steps are performed by participants. SP is responsible for initializing the system and aggregating all participants' locally encrypted results in the first step and the fourth step, respectively. Note that if `SumAgg` takes $\{\mathtt{d}_1, \cdots, \mathtt{d}_n\}$ as input, $\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d})$ outputs $\sum_{i=1}^{n} \mathtt{d}_i$. The key idea of `SumAgg` is that each participant performs aggregation operations on locally raw sensing data, while SP aggregates local aggregation results via a privacy-preserving manner.

*2) **MenAgg**:* Given $\{d_1, \cdots, d_n\}$, `MenAgg` outputs $\mu = \frac{\sum_{i=1}^{n} \sum_{j=1}^{\mathtt{d}_i} x_{i,j}}{\sum_{j=1}^{n} \mathtt{d}_j}$. Technically, `MenAgg` can be formulated as $\Gamma(D) \triangleq \frac{\Gamma_1(D)}{\Gamma_2(D)}$, where $D = \{d_1, \cdots, d_n\}$. $\Gamma_1(D)$ and $\Gamma_2(D)$ are a sum Γ-function and a count Γ-function, respectively. Thus, the mean function is a Γ-representable function. `MenAgg` is easily derived from PRADA. Formally, $\mathtt{MenAgg}(d_1, \cdots, d_n)$ is denoted by

$$\mathtt{MenAgg}(d_1, \cdots, d_n) \triangleq \frac{\mathtt{SumAgg}(d_1, \cdots, d_n)}{\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d}_n)}. \tag{15}$$

If SP obtains the sums of all participants' raw sensing data (i.e., $\sum_{i=1}^{n} \sum_{j=1}^{\mathtt{d}_i} x_{i,j}$) and the total amount of raw sensing

data ($\sum_{j=1}^{n} \mathtt{d}_j$), he can compute the mean of all participants' raw sensing data μ.

As depicted in Algorithm 2, MenAgg calls SumAgg twice to obtain $\Gamma_1(d_1, \cdots, d_n)$ and $\Gamma_2(\mathtt{d}_1, \cdots, \mathtt{d}_n)$, i.e., $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}$ and $\sum_{i=1}^{n} \mathtt{d}_i$, respectively. Next, SP outputs μ by computing $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}}{\sum_{i=1}^{n} \mathtt{d}_i}$.

Algorithm 2 $\mathtt{MenAgg}(d_1, \cdots, d_n) \rightarrow \mu$

Input: n participants have $\{d_1, \cdots, d_n\}$.

Output: SP obtains $\mu = \frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}}{\sum_{j=1}^{n} \mathtt{d}_j}$.

Procedure:

Sum aggregation (@SP & @Participants):

- SP defines Γ_1 as a sum function and Γ_2 as a count function;
- SP and participants jointly perform $\mathtt{SumAgg}(d_1, \cdots, d_n)$ and $\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d}_n)$ to obtain $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}$ and $\sum_{i=1}^{n} \mathtt{d}_i$, respectively.

Aggregation (@SP):

- Compute $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}}{\sum_{i=1}^{n} \mathtt{d}_i}$ and output the aggregation result.

From Algorithm 2, we see that if an aggregation operation (e.g., mean) is a Γ-respresentable function, it can be implemented by calling Γ-function multiple times. Following the same idea, we also construct VarAgg. In fact, the advantage of CROWDFA is to achieve multiple types of data aggregation operations by the same framework.

3) ***VarAgg****:* Given $\{d_1, \cdots, d_n\}$, VarAgg outputs σ^2, where $\sigma^2 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j}-\mu)^2}{\sum_{j=1}^{n} \mathtt{d}_j}$. According to the Definition of variance $\frac{\sum_{i=1}^{m}(x_i-\mu)^2}{m}$, VarAgg can be formulated as $\Gamma(D) \triangleq \frac{\Gamma_3(D)}{\Gamma_2(D)} - \frac{\Gamma_1(D)^2}{\Gamma_2(D)^2}$, where $D = \{d_1, \cdots, d_n\}$. $\Gamma_1(D)$ and $\Gamma_2(D)$ are a sum Γ-function and a count Γ-function, respectively. Γ_3 is denoted by $\Gamma_3(d) = \sum_{j=1}^{\mathtt{d}} \phi(x_j)$, where $\phi(x_j) = (x_j)^2$. Thus, the variance function is a Γ-representable function. VarAgg is easily derived from PRADA. Formally, $\mathtt{VarAgg}(d_1, \cdots, d_n)$ is denoted by

$$\begin{aligned}\mathtt{VarAgg}(d_1, \cdots, d_n) &\triangleq \frac{\mathtt{SumAgg}(\phi(d_1), \cdots, \phi(d_n))}{\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d}_n)} \\ &\quad - \frac{(\mathtt{SumAgg}(d_1, \cdots, d_n))^2}{(\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d}_n))^2}. \end{aligned} \tag{16}$$

If SP obtains $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2$, $\sum_{j=1}^{n} \mathtt{d}_j$, and $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}$, he can compute the variance of all participants' raw sensing data σ^2.

As depicted in Algorithm 3, VarAgg calls SumAgg thrice to obtain $\Gamma_1(d_1, \cdots, d_n)$, $\Gamma_2(\mathtt{d}_1, \cdots, \mathtt{d}_n)$, and $\Gamma_3(\phi(d_1), \cdots, \phi(d_n))$, i.e., $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}$, $\sum_{i=1}^{n} \mathtt{d}_i$, and $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2$. Next, SP outputs σ^2 by computing $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2}{\sum_{i=1}^{n} \mathtt{d}_i} - \frac{(\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j})^2}{(\sum_{i=1}^{n} \mathtt{d}_i)^2}$.

Algorithm 3 $\mathtt{VarAgg}(d_1, \cdots, d_n) \rightarrow \sigma^2$

Input: n participants have $\{d_1, \cdots, d_n\}$.

Output: SP obtains $\sigma^2 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j}-\mu)^2}{\sum_{j=1}^{n} \mathtt{d}_j}$.

Procedure:

Sum aggregation (@SP & @Participants):

- SP defines Γ_2 as a sum function under a transformation function $\phi(x) = x^2$, Γ_1 as a sum function, and Γ_2 as a count function, respectively;
- SP and participants jointly perform $\Gamma_1(d_1, \cdots, d_2)$, $\Gamma_2(\mathtt{d}_1, \cdots, \mathtt{d}_n)$, and $\Gamma_3(\phi(d_1), \cdots, \phi(d_n))$ to obtain $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j}$, $\sum_{i=1}^{n} \mathtt{d}_i$, and $\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2$, respectively.

Aggregation (@SP):

- Compute $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2}{\sum_{i=1}^{n} \mathtt{d}_i} - \frac{(\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j})^2}{(\sum_{i=1}^{n} \mathtt{d}_i)^2}$ and output the aggregation result.

In the proposed VarAgg, we transform $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j}-\mu)^2}{\sum_{j=1}^{n} \mathtt{d}_j}$ into $\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2}{\sum_{i=1}^{n} \mathtt{d}_i} - \frac{(\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j})^2}{(\sum_{i=1}^{n} \mathtt{d}_i)^2}$. It is easy to verify that

$$\begin{aligned}\frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j}-\mu)^2}{\sum_{j=1}^{n} \mathtt{d}_j} &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i}(x_{i,j})^2}{\sum_{i=1}^{n} \mathtt{d}_i} \\ &\quad - \frac{(\sum_{i=1}^{n}\sum_{j=1}^{\mathtt{d}_i} x_{i,j})^2}{(\sum_{i=1}^{n} \mathtt{d}_i)^2}. \end{aligned} \tag{17}$$

On the other hand, as the product function is also a Γ-function, VarAgg can also be denoted by

$$\mathtt{VarAgg}(d_1, \cdots, d_n) \triangleq \frac{\mathtt{MulAgg}(\phi'(d_1), \cdots, \phi'(d_n))}{\mathtt{SumAgg}(\mathtt{d}_1, \cdots, \mathtt{d}_n)}, \tag{18}$$

where MulAgg means a product Γ-function, and $\phi'(d_i) = \Pi_{j=0}^{\mathtt{d}_i}(x_{i,j} - \mu)^2$. Following the above idea, PRADA can support other privacy-preserving data aggregation operations, such as p-order moment, skewness, and kurtosis. Technically, the p-order moment is defined as $v_p = \frac{\sum_{i=1}^{i=n}(x_i-\mu)^p}{N}$, where μ represents the mean. Skewness is defined as $s = \frac{v_3}{\sigma^3}$, while kurtosis is defined as $\kappa = \frac{v_4}{\sigma^4}$, where σ indicates the standard deviation. It is easy for PRADA to compute $\sum_{i=1}^{i=n}(x_i - \mu)^p$ by using binomial theorem. PRADA easily calculates $\sum_{i=1}^{i=n}(x_i - \mu)^p$ using the binomial theorem. Moreover, since PRADA can obtain σ, it can compute the p-order moment, skewness, and kurtosis efficiently. Additionally, if $\Gamma(\cdot)$ is defined as a maximum-value function or a minimum-value function, CROWDFA can output the minimum or the maximum of all participants' raw sensing data. Without loss of generality, $\Gamma(d)$ is defined as $\min\{x_1, \cdots, x_{\mathtt{d}}\}$, and Γ is also defined as min operation. Thus, we have $\Gamma(d_1, \cdots, d_n) \triangleq \min\{\Gamma(d_1), \cdots, \Gamma(d_n)\}$. In the next section, we instantiate $\mathcal{L}_{B,\sigma}$ in Eq. (4) as a reward distribution function.

V. PRAED Design

Mobile crowdsensing platforms, such as Gigwalk,[3] TaskRabbit,[4] Waze,[5] McSense [42], and CrowdOS [26], adopt incentive mechanisms to engage and reward participants. Monetary incentives for mobile crowdsensing are typically implemented by allowing each participant to bid, and rewards are distributed based on their respective bids [6], [15], [26]. there are at least two issues with this approach. Firstly, participants are likely to dishonestly bid, which compromises the truthfulness of the incentive mechanism. Secondly, a participant's bid does not necessarily reflect their actual contribution, thereby undermining the fairness of the incentive mechanism. Truthfulness and fairness are indeed two essential requirements for an effective incentive mechanism [6]. To tackle these issues, a practical design called `HyInc` has been proposed in the work [41] to ensure both truthfulness and fairness in the incentive mechanism. Unfortunately, `HyInc` fails to protect bid privacy [27].

To the aforementioned issues, this section elaborates on PRAED,[6] a privacy-preserving incentive design based on CrowdFA. While drawing inspiration from `HyInc`, PRAED distinguishes `HyInc` by offering two significant improvements and at least two noteworthy novelties. Regarding the improvements, firstly, PRAED ensures the protection of bid privacy. Secondly, it maximizes the reward for each participant while adhering to a given reward budget. In terms of novelties, PRAED introduces a shared paradigm/framework that unifies both privacy-preserving incentive design and privacy-preserving data aggregation. Additionally, PRAED maintains its truthfulness and fairness even while safeguarding bid privacy. Bids are typically associated with participants' private information, such as their location. The disclosure of bids can result in a breach of location privacy [18], [43], [44], [45]. Protecting the privacy of participants' bids is crucial as revealing them could potentially incur threats to participants' location [45]. Participants' bids may contain sensitive information such as location [43].

A. Formulation of PRAED

Technically, PRAED instantiates $\mathcal{L}_{B,\sigma}$ in Eq. (4) as a reward distributed function. Specifically, PRAED formulates $\mathcal{L}_{B,\sigma}(b_i)$ as

$$\mathcal{L}_{B,\sigma}(b_i) = \begin{cases} \sigma(i) + u \cdot b_i, & \text{for } b_i \le \min_{j\neq i} b_j, \\ u \cdot b_i, & \text{others.} \end{cases} \tag{19}$$

$$\text{s.t.} \quad \sigma(i) = \alpha(\min_{j\neq i} b_j - b_i) \text{ for } b_i \le \min_{j\neq i} b_j,$$

$$u = \frac{\beta}{\sum_{j=1}^{n} b_j},$$

$$\sum_{i=1}^{n} \mathcal{L}_{B,\sigma}(b_i) = B, \tag{20}$$

where b_i represents the participant i's bid, and $\min_{j\neq i} b_j$ denotes the minimum value among $\{b_1, \cdots, b_n\} \setminus b_i$. α and β serve as two control parameters. In PRAED, the winner selection process is formulated as $\sigma(i)$. To calculate (u, v) and perform $\sigma(i)$, PRAED employs additive secret sharing to design an interactive protocol among participants and SP, ensuring both privacy and efficiency. Following the design of `HyInc` [41], each participant's sensing data quantity is regarded as considered their cost and contribution in PRAED. Particularly, when the participant i sets $b_i = \mathtt{d}_i$, she is considered as honest bidding.

Eq. (19) demonstrates that if each participant transmits their bids to SP directly, bids are leaked, thereby compromising the privacy of participants' bid privacy [18], [27]. To address this issue, PRAED mandates that each participant only submits one secret share of their bids to SP. Consequently, PRAED faces three key challenges: 1) *How can* $\min_{j\neq i} b_j - b_i$ *of Eq. (20) be computed without any knowledge of the participant's bids*; 2) *How can the participant with the minimum bid be identified without revealing any participant's bid*; and 3) *How How can maximum rewards for each participant be achieved while ensuring truthfulness and fairness.*

B. Implementation of PRAED

Observation. Given $\{x_1, \cdots, x_n\}$ $(x_i \in \mathbb{Z}_{2^\ell})$, if x_k is the minimum among $\{x_1, \cdots, x_n\}$, $x_k - y \mod N$ is also the minimum among $\{x_1 - y \mod N, \cdots, x_n - y \mod N\}$, where $2^\ell < N$ and $\max\{x_1, \cdots, x_n\} < y < N$. Furthermore, given $\{x_1 - y \mod N, \cdots, x_n - y \mod N\}$, it is easy to output $\min_{i\neq k} x_j - x_k$ by sorting $\{x_1 - y \mod N, \cdots, x_n - y \mod N\}$.

Following the design of PRADA and the above observation, if PRAED also designates one participant (e.g., the participant k) as a leader responsible for computing (u, v) and $\sigma(i)$ in Eq. (20), it becomes a challenging to prevent the leader from selecting himself as the winner, i.e., the leader sets $i = k$. To overcome this issue, PRAED employs a two-stage approach in the winner selection operation and selects two distinct leaders. In the first stage, the first leader selects two potential winners except for himself. Then, in the second stage, the second leader (excluding the potential winners) identifies the final winner among two potential winners and the first leader.

As shown in Algorithm 4, PRAED consists of three steps. In the first step, each participant offers secret bidding protected by additive secret sharing. In the second step, PRAED selects the winner who bids the lowest price without leaking any participant's bid. Specifically, SP and two leaders jointly compute $\min_{j\neq i} b_j - b_i$ for $b_i \le \min_{j\neq i} b_j$ and locate i. In the last step, PRAED determines each participant's reward based on Eq. (19). Specifically, each participant cooperates with SP to compute her reward. Note that PRAED assumes all bidding is less than 2^ℓ and $2^\ell < N$, where N is the parameter of additive secret sharing, and ℓ can be 32. The participants k and k' are the first leader and the second leader, respectively.

According to Algorithm 4, the values of α and β in Eq. (20) are quantized as $\frac{B}{\sum_{i=1}^{n} b_i + \Delta}$ and $\frac{B \cdot \sum_{i=1}^{n} b_i}{\sum_{i=1}^{n} b_i + \Delta}$, respectively. Note

[3] https://www.gigwalk.com/
[4] https://www.taskrabbit.com/
[5] https://www.waze.com/
[6] PRAED: PRivAcy-preserving incEntive Design.

Algorithm 4 Privacy-Preserving Incentive Design

Input: n participants have $\{b_1, \cdots, b_n\}$, and SP has a reward budget B.
Output: Each participant i obtains a reward u_i ($i \in [1..]$).
Procedure:
Secret bidding (@Participants):

• Each participant takes b_i ($i \in [1..n]$) as bidding, and secretly shares b_i into $[b_i]_1$ and $[b_i]_2$ by calling $\mathsf{S}(b_i)$. Next, each participant sends $[b_i]_1$ to SP.

Winner selection (@SP & @Participants):

• SP randomly selects $k \xleftarrow{\$} [1..n]$ and $r \xleftarrow{\$} \{2^\ell + 1, \cdots, N\}$. Next, SP computes $\{[b_i]_1 - r \mod N\}$ for $i \in [1..n] \setminus k$ and transmits them to the participant k;

• The participant i sends $[b_i]_2$ to the participant k, where $i \in [1..n] \setminus k$. After that, the participant k computes $\{[b_i]_1 - r + [b_i]_2 \mod N\}$ for $i \in [1..n] \setminus k$, and then sorts them and transmits the index p, q of the smallest two items to SP, where $p, q \in [1..n] \setminus k$;

• SP randomly selects $k' \xleftarrow{\$} [1..n] \setminus \{p, q, k\}$ and $r' \xleftarrow{\$} \{2^\ell + 1, \cdots, N\}$, and then computes $\{[b_j]_1 - r' \mod N\}$ for $j \in \{p, q, k\}$ and transmits them to the participant k';

• Participants p, q and k send $[b_p]_2$, $[b_q]_2$, and $[b_k]_2$ to the participant k', respectively. Next, the participant k' computes $\{[b_j]_1 - r' + [b_j]_2 \mod N, \}$ for $j \in \{p, q, k\}$, and then outputs the index p', q' of the smallest two items ($p', q' \in \{p, q, k\}$). After that, the participant k' calculates

$$\Delta \leftarrow ([b_{p'}]_1 - r' + [b_{p'}]_2) - ([b_{q'}]_1 - r' + [b_{q'}]_2) \mod N$$

and sends Δ and q' to SP. Without loss of generality, let $[b_{p'}]_1 - r' + [b_{p'}]_2 \mod N \geq [b_{q'}]_1 - r' + [b_{q'}]_2 \bmod N$. Lastly, SP sets the participant q' as the winner.

Rewarding (@SP & @Participants):

• SP computes $v \leftarrow \frac{B \cdot \Delta}{\sum_{i=1}^n b_i + \Delta}$ and $u \leftarrow \frac{B}{\sum_{i=1}^n b_i + \Delta}$ and broadcasts them to all participants;

• The participant i ($i \in [1..n] \setminus q'$) computes $u_i \leftarrow u \cdot b_i$, while the participant q' computes $u_{q'} \leftarrow v + u \cdot b_{q'}$.

that PRAED calls $\texttt{SumAgg}(b_1, \cdots, b_n)$ to calculate $\sum_{i=1}^n b_i$ without revealing any participant's bid. Furthermore, it can be easily verified that $\sum_{i=0} u \cdot b_i + v = B$, indicating that all participants earn the entire reward budget allocated by SP. Since each participant's dominant strategy is to honestly bid (see Section VI), which means each participant maximizes their rewards by setting $b_i = \mathtt{d}_i$ ($i \in [1..n]$). Therefore, PRADED effectively maximizes each participant's reward.

VI. Analysis and Evaluation

A. Theoretical Analyses

In this section, we demonstrate the proposed `SumAgg`, `MenAgg`, and `VarAgg` do not leak any participant's raw sensing data to SP and aggregation results to any participant. We also present evidence confirming that PRAED does not reveal any participant's bid to SP. Furthermore, we give proof sketches that establish the the *truthfulness* and *fairness* attributes of PRAED.

Theorem 1: For any Γ-function, PRADA prevents 1) SP from learning participants' raw sensing data and 2) any participant from learning other participants' raw sensing data.

Proof: We prove Theorem 1 from two-aspect: 1) consider SP is corrupted, and 2) consider one participant is corrupted. We prove them by following two lemmas.

Lemma 1: If SP is corrupted, there exists a simulator $\mathcal{S}$ that takes as input received messages by SP to generate a view that is computationally indistinguishable from the one of SP.

Proof: In PRADA, SP receives multiple shares $[\Gamma(d_i)]_1$ from each participant and then outputs the final aggregation result $\mathsf{output}_{SP}^{\pi} = \sum_{i=1}^n \Gamma(d_i)$. Besides, other participants in PRADA receive no aggregation result, namely $\mathsf{output}_i^{\pi} = \perp$. Therefore, we have that $\mathsf{output}^{\pi}(d_1, \cdots, d_n) \stackrel{c}{\equiv} \{\mathsf{output}_{\mathrm{SP}}^{\pi}\}$, which is denoted by y for simplicity. In other words, SP's view in the real execution of PRADA is as follows.

$$\mathcal{V}_{SP} = \{[\Gamma(d_1)]_1, [\Gamma(d_2)]_1, \cdots, [\Gamma(d_n)]_1\},$$

We assume the participant $k = n$ is selected as the leader. We construct a simulator $\mathcal{S}$ that only takes y as input and works as follows.

1) $\mathcal{S}$ selects $n-1$ random numbers (say $s_1, s_2, \cdots, s_{n-1}$) from $\mathbb{Z}_N$;
2) $\mathcal{S}$ outputs $\{s_1, s_2, \cdots, s_{n-1}, y - \sum s_i\}$.

Note that in the real execution of PRADA, for any $i \neq n$, $[\Gamma(d_i)]_1$ is obtained from additive secret sharing and set by $[\Gamma(d_i)]_1 = \Gamma(d_i) - r_i \mod N$, where r_i is a random number in $\mathbb{Z}_N$. In case $i = n$, $[\Gamma(d_i)]_1$ (according to Eq. (12)) is calculated by $[\Gamma(d_n)]_1 = \Gamma(d_n) + \sum_{i \neq n} r_i \mod N$. Thus we conclude that for any dataset $(d_1, d_2, \cdots, d_n)$, it is not hard to verify the two distributions

$$\{s_1, s_2, \cdots, s_{n-1}, y - \sum s_i, y\}$$

and

$$\{\Gamma(d_1) - r_1, \cdots, \Gamma(d_{n-1}) - r_{n-1}, \Gamma(d_n) + \sum_{i \neq n} r_i, \sum \Gamma(d_i)\}$$

are computationally indistinguishable, which completes the proof.

Lemma 2: If participant i is corrupted, there exists a simulator $\mathcal{S}_i$ that takes as input received messages by the participant i to generate a view that is computationally indistinguishable from participant i's view.

Proof: In PRADA, any non-leader participants do not receive any messages from other participantss. Consequently, the view of a non-leader participant is empty, denoted as $\mathcal{V}_i^{\pi} = \perp$. Thus, for any non-leader participant i, the simulator $\mathcal{S}_i$ simply outputs $\perp$, indicating no no information. In other words, a non-leader participant is corrupted, they gain no knowledge about the raw sensing data of other participants. However, we remains to demonstrate that a corrupted leader participant cannot access other participants' raw sensing data. Without loss of generality, we assume the participant k is selected as the leader. As depicted in PRADA, the view of the participant k is as follows.

$$\mathcal{V}_{Leader} = \{[\Gamma(d_1)]_2, \cdots, [\Gamma(d_{k-1})]_2, [\Gamma(d_{k+1})]_2, \cdots, [\Gamma(d_n)]_2\}.$$

We construct a simulator $\mathcal{S}_k$ that only takes as inputs $[\Gamma(d_i)]_2$ ($i \in [1..n] \setminus k$) and works as follows. $\mathcal{S}_k$ selects $n-1$ random numbers (say $s_1, \cdots, s_{k-1}, s_{k+1}, \cdots, s_n$) from $\mathbb{Z}_N$

and outputs the set directly. For any dataset $(d_1, d_2, \cdots, d_n)$, it is not hard to verify the following two distributions

$$\{s_1, \cdots, s_{k-1}, s_{k+1}, \cdots, s_n\}$$

and

$$\{[\Gamma(d_1)]_2, \cdots, [\Gamma(d_{k-1})]_2, [\Gamma(d_{k+1})]_2, \cdots, [\Gamma(d_n)]_2\}$$

are computationally indistinguishable, which completes the proof.

Since Lemma 1 and Lemma 2 hold, Theorem 1 holds.

Theorem 2: For any Γ-function, PRADA achieves $(n-1)$-collusion resistance.

Proof: Suppose $\mathcal{A}$ is a P.P.T. adversary that corrupts at most t participants in PRADA. Without loss of generality, let $I \subset [1..n]$ be a set that $\mathcal{A}$ corrupts. We define $I = \{i_1, i_2, \cdots, i_m\}$, where $m \leq n-1$. Obviously, $\mathcal{A}$'s view in real execution is the joint of all corrupted participants. Specifically, $\mathcal{A}$'s view is $\{\mathcal{V}_{i_1}^{\pi}, \mathcal{V}_{i_2}^{\pi}, \cdots, \mathcal{V}_{i_m}^{\pi}\}$.

Without loss of generality, we still assume the participant k is selected as the leader. Since any non-leader participant, we have that $\mathcal{V}_i^{\pi} = \perp$. Therefore, we have the following two cases.

$$\mathcal{V}_{\mathcal{A}}^{\pi} = \begin{cases} \perp, & k \notin I, \\ \mathcal{V}_{Leader}, & k \in I \end{cases}$$

We construct a simulator $\mathcal{S}_I$ that only takes the $\{d_i, \Gamma(d_i)\}_{i \in I}$ as inputs and works as follows. If $k \notin I$, it outputs $\perp$ directly; otherwise, it invokes $\mathcal{S}_k$ (presented in Lemma 2) as a copy and returns the output of $\mathcal{S}_k$. It is not hard to verify $\mathcal{A}$'s view in our simulation is computationally indistinguishable from its view in real execution. Therefore, Eq. (7) holds, which completes the proof.

Corollary 1: In `SumAgg`, SP fails to learn any participant's raw sensing data, while any participant fails to learn the sums of all participants' raw sensing data.

Proof: Since the sum is a Γ-function, this corollary is derived directly from Theorem 1.

Corollary 2: In `MenAgg`, SP fails to learn any participant's raw sensing data, while any participant fails to learn the mean of all participants' raw sensing data.

Proof: The mean is a Γ-representable function based on the sum Γ-function, and $\texttt{MenAgg}(d_1, \cdots, d_n) \triangleq \frac{\texttt{SumAgg}(d_1, \cdots, d_n)}{\texttt{SumAgg}(1, \cdots, 1)}$. In other words, `MenAgg` only performs `SumAgg` twice. Thus, as long as Corollary 1 holds, the corollary also holds.

Corollary 3: In `VarAgg`, SP fails to learn any participant's raw sensing data, while any participant fails to learn the variance of all participants' raw sensing data.

Proof: This proof is similar to COROLLARY 2, and we omit the detailed proof because of limited pages.

Theorem 3: PRAED prevents 1) SP from learning participants' bids and 2) any participant from learning other participants' bids.

Proof: We prove Theorem 3 from two-aspect: 1) consider SP is corrupted, and 2) consider one participant is corrupted. We prove them by following two lemmas.

Lemma 3: If SP is corrupted, there exists a simulator $\mathcal{S}$ that takes as input received messages by SP to generate a view that is computationally indistinguishable from the one of SP.

Proof: In PRAED, SP's communications consist of six parts. Specifically,

1) Receiving bidding shares $[b_i]_1$ from each participant i for $i \in [1..n]$;
2) Sending masked shares $[b_i]_1 - r \mod N$ to a randomly selected participant k, for $i \in [1..n] \backslash k$;
3) Receiving two candidates p, q from the participant k;
4) Sending masked shares $[b_j]_1 - r' \mod N$ to another randomly selected participant k', for $j \in \{p, q, k\}$;
5) Receiving a final candidate q' from the participant k'; and
6) Outputting a reward base (u, v).

Note that the participant's output is either $u \cdot b_i$ or $v + u \cdot b_i$ $(i \in [1..n])$. Without loss of generality, we assume the 1-st participant bids the smallest bidding. Then, we have

$$\mathsf{output}^{\pi}(b_1, \cdots, b_n) \stackrel{c}{\equiv} \{v + u \cdot b_1, u \cdot b_2, \cdots, u \cdot b_n\}.$$

and SP's view in the real execution of PRAED is as follows.

$$\mathcal{V}_{SP} = \{[b_1]_1, [b_2]_1, \cdots, [b_n]_1, (p, q), q'\}.$$

where $q' \in \{p, q, k\}$. We first define a leakage function $L_{SP} = \{B, p, q, q', \Delta\}$, which implies a legal information leakage. We construct a simulator $\mathcal{S}(L_{SP}, u, v)$ that takes the legal leakage L_{SP} and SP's output (u, v) as inputs and works as follows.

1) $\mathcal{S}$ selects n random numbers $r_1, r_2, \cdots, r_n$ from $\mathbb{Z}_N$ and then sends them to SP;
2) Receiving k and $\{[b_i]_1 - r \mod N\}$ for $i \in [1..n] \backslash k$ from SP, $\mathcal{S}$ forwards (p, q) to SP;
3) Receiving k' and $\{[b_i]_1 - r' \mod N\}$ for $i \in \{p, q, k\}$ from SP, $\mathcal{S}$ forwards q' to SP;
4) $\mathcal{S}$ selects n random bidding $b_1', b_2', \cdots, b_n'$ and then outputs the final rewards $\{v + u \cdot b_1', u \cdot b_2', \cdots, u \cdot b_n'\}$.

It is not hard to verify that SP's view in real execution and $\mathcal{S}$'s view are computationally indistinguishable. Because $[b_i]_1$ is generated from additive secret sharing, which is also a random number in $\mathcal{A}$'s view.

Lemma 4: If participant i is corrupted, there exists a simulator $\mathcal{S}_i$ that takes as input received messages by the participant i to generate a view that is computationally indistinguishable from the participant i's view.

Proof: In PRAED, for any participant i, if $i \in \{k, k'\}$, it receives no message from other participants; therefore, the views of these participants are empty, namely $\mathcal{V}_i^{\pi} = \perp$. For participant $i \notin \{k, k'\}$, we just let simulator $\mathcal{S}_i$ output $\perp$. However, it remains to prove that the corrupted participant k or k' cannot obtain other participants' bids. As depicted in PRAED, the view $\mathcal{V}_k^{\pi}$ of participant k is denoted by $\{[b_i]_1 - r \mod N\}_{i=1, i \neq k}^{n}$ and $\{[b_i]_2\}_{i=1, i \neq k}^{n}$, where the former is the view from SP and the latter is from other participants. We construct a simulator $\mathcal{S}_k$ that only takes b_k as input and works as follows. $\mathcal{S}_k$ selects $2(n-1)$ random numbers (say s_1, $\cdots$, s_{k-1}, s_{k+1}, $\cdots$, s_n and $r_1, \cdots, r_{k-1}, r_{k+1}, \cdots, r_n$) from $\mathbb{Z}_N$ and outputs the set directly. Note that in the real execution of PRAED, for any $i \neq k$, $[b_i]_1$ and $[b_i]_2$ are obtained from

additive secret sharing; Therefore, the participant k's view in real execution and in simulation execution are computationally indistinguishable.

As for simulator $\mathcal{S}_{k'}$ for participant k', we can build it in a similar manner as $\mathcal{S}_k$. Because the communication of participant k and participant k' is similar, and it is a more simplified case. Therefore, we omit the construction of $\mathcal{S}_{k'}$. Taken together, Lemma 4 holds.

Since Lemma 3 and Lemma 4 hold, Theorem 3 hold.

Theorem 4: PRAED is truthful, i.e., any participant cannot improve his rewards by submitting a deviated bid without knowing others' bids.

Proof: If any participant maximizing rewards is to bid honestly, i.e., $b_i = \mathtt{d}_i$, they fail to improve rewards by submitting a deviated bid. We consider two cases: (1) $\min_{j\neq i} b_j \geq \mathtt{d}_i$, and (2) $\min_{j\neq i} b_j < \mathtt{d}_i$. Here, b_i represents the participant i's bid ($i \in [1..n]$), and $\min_{j\neq i} b_j$ is the minimum among $\{b_1, \cdots, b_j\} \setminus b_i$. We first prove that the participant i ($i \in [1..n] \setminus k$) can maximize their reward by setting $b_i = \mathtt{d}_i$ in both cases, where the k-th participant is the first leader. For simplicity, we assume $b \leftarrow \min_{j\neq i} b_j$.

- **Case 1:** $b \geq \mathtt{d}_i$. When $b_i \leq b$, the participant i is a winner and gets a reward $\alpha(b - \mathtt{d}_i) + \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$. Also, if $b_i > b$, the participant i gets a reward $\beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$, where α, β are two control parameters and $\alpha, \beta > 0$. In this case, as $\alpha(b - \mathtt{d}_i) + \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j} \geq \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$ always holds, the participant i maximizes rewards by submitting a bid $b_i \leq \mathtt{d}_i$.
- **Case 2:** $b < \mathtt{d}_i$. When $b_i \leq b$, the participant i is the winner and gets a reward $\alpha(b - \mathtt{d}_i) + \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$. Also, if $b_i > b$, the participant i gets a reward $\beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$. In this case, as $\alpha(b - \mathtt{d}_i) + \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j} < \beta \cdot \frac{b_i}{\sum_{j=1}^{n} b_j}$ always holds, the participant i maximizes rewards by submitting a bid $b_i \geq \mathtt{d}_i$.

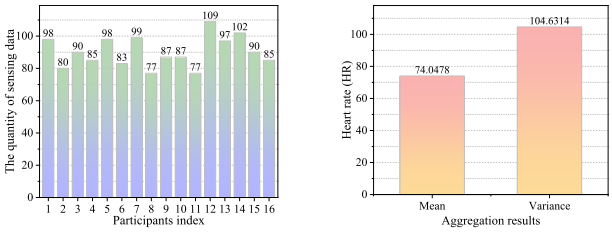
Thus, the participant i ($i \in [1..n] \setminus k$) maximizes his rewards by honestly biding, i.e., $b_i = \mathtt{d}_i$.

Now, we consider the strategy of the first leader who is aware that the participant p or q could potentially win. Without loss of generality, we assume $b = \min\{b_p, b_q\}$. From the perspective of the first leader, there are two possible scenarios: either $b_k \geq \mathtt{d}_K$ or $b_k < \mathtt{d}_k$. As proven above, it is evident that the dominant strategy for the first leader is to set $b_k = \mathtt{d}_k$. Likewise, the dominant strategies for participants p and q are to set $b_p = \mathtt{d}_p$ and $b_q = \mathtt{d}_q$, respectively. On the other hand, the second leader cannot modify their strategy once the final winner is being determined. Consequently, their dominant strategy would be to set $b_{k'} = \mathtt{d}_{k'}$.

In summary, participants in PRAED fail to enhance their rewards by submitting deviated bids without knowledge of others' bids. Consequently, PRAED can be considered as a truthful mechanism.

Theorem 5: PRAED is fair, i.e., any participant's reward is positively related to her amount of sensing data.

Proof: Without loss of generality, assume $b_i > b_j$, where $i, j \in [1..n]$. There are two cases: 1) the participant j is the winner; 2) the participant j is not the winner.



(a) The size of sensing data (b) Aggregation results

Fig. 3. Experimental data and aggregation results.

- **Case 1**: The reward of the participant j is $u_j = v + u \cdot b_j$, and that of the participant i is $u_i = u \cdot b_i$. As the participant i and the participant j set $b_i = \mathtt{d}_i$ and $b_j = \mathtt{d}_j$, respectively to obtain truthful rewards, $u_i \geq u_j$ always holds when $\mathtt{d}_i > \mathtt{d}_j$.
- **Case 2**: The reward of the participant j is $u_j = u \cdot b_j$, and that of the participant i is $u_i = u \cdot b_i$, where $b_i = \mathtt{d}_i$ and $b_j = \mathtt{d}_j$. If $\mathtt{d}_i > \mathtt{d}_j$, $u_i > u_j$ always holds.

When considering all the factors, it becomes evident that the reward of any participant is directly correlated to the quantity of sensing data they contribute. Therefore, the fairness of PRAED can be concluded.

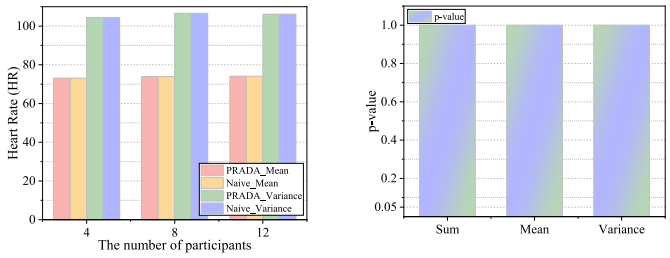
B. Experimental Evaluations

In this section, we implement PRADA and PRAED in Java to evaluate their feasibility and efficiency using a real-world sensing dataset.[7] We adopt the heart rate from the dataset as the sensing data, denoted by an integer. Note that the proposed solution supports floating-point numbers by converting them into integers. The experimental settings are as follows: We use a personal computer equipped with Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz and 32 GB RAM as the sensing platform, simulating 16 participants. The value of N for the additive secret sharing is set to $N = 2^{62}$. Additionally, we allocate a reward budget $B = \$28.88$. Furthermore, we we compare our approach with the state-of-the-art PAGE [25] and a naive solution (refereed to as "Naive" throughout this paper). Both PAGE and Naive utilizes the additive secret sharing for privacy-preserving data aggregation, while SP in Naive aggregates participants' raw sensing data directly. All experimental results presented here are averaged over 200 independent experiments.

According to the real-world sensing dataset, we assume that 16 participants take part in collecting sensing data. Each participant contributes an amount of sensing data shown in Fig. 3(a). In total, 16 participants contribute 1444 sensing data from Fig. 3(a). The mean and variance of 1444 sensing data are 74.0478 and 104.6314 depicted in Fig. 3(b), respectively. It is evident that the sum of the 1444 sensing data points amounts to 106925.

To demonstrate the feasibility of PRADA, we compare the aggregation results between PRADA and Naive. Additionally, we employ the Wilcoxon rank-sum test at a significance level of 0.05 to perform statistical tests. The p-value of

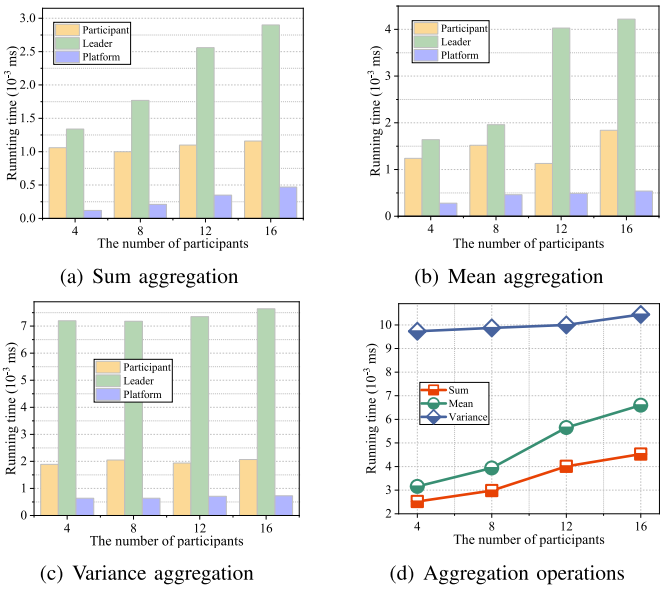
[7]https://www.kaggle.com/datasets/saurav9786/heart-rate-prediction?



(a) Aggregation results between PRADA and Naive

(b) Statistical tests between PRADA and Naive

Fig. 4. Feasibility evaluations.



(a) Sum aggregation

(b) Mean aggregation

(c) Variance aggregation

(d) Aggregation operations

Fig. 5. The running time of entities for different aggregation operations.

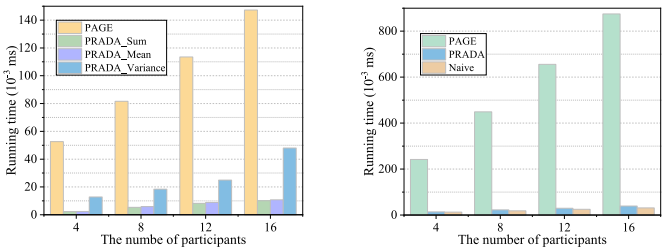
the Wilcoxon rank-sum test between two datasets indicates significant differences when it is less than 0.05. Conversely, if the p-value exceeds 0.05, there is no significant difference between the two datasets according to the statistical tests. The experimental results depicted in Figure 4(a) show that PRADA outputs the same mean and variance as Naive. Consequently, PRADA generates the same sum as Naive when both algorithms aggregate an identical amount of sensing data. Since $[x]_1 + [x]_2 \mod N = x$ for $x \in \mathbb{Z}_N$, we can deduce that $\sum_{i=1}^{n}[x_i]_1 + \sum_{i=1}^{n}[x_i]_2 \mod N = \sum_{i=1}^{n} x_i$. In other words, additive secret sharing consistently yields correct sum aggregations, ensuring that PRADA maintains the same sum of sensing data as Naive. Given that PRADA always aggregates sums accurately, it consistently produces the same aggregation results as Naive. As a result, PRADA introduces no noise into the aggregation results.

In terms of statistical tests, we examine the p-value between PRADA and Naive, as depicted in Fig. 4. Taking into account the sum, mean, and variance, the calculated p-value between PRADA and Naive is greater than 0.05. Consequently, we conclude that there is no significant difference between PRADA and Naive when aggregating participants' raw sensing data directly. Thus, PRADA demonstrates its feasibility as a solution for aggregating sensing data based on the proposed CrowdFA paradigm.

PRADA comprises three types of entities, namely participant, leader, and sensing platform. Each type of entity carries out distinct operations to generate aggregation results. Additionally, different aggregation operations necessitate specific operations from each entity. Consequently, we assess the average running time of each entity for various aggregation operations. The experimental results depicted in Fig. 5 indicate that leaders consume more time compared to other entities when generating results for different aggregation operations. One possible explanation for this disparity is that leaders perform a greater number of modular arithmetic operations than other entities. Since modular arithmetic requires more computational resources than arithmetic operations, leaders take longer to complete their tasks. On the other hand, exhibit the shortest running time among the three types of entities across different aggregation operations. In contrast to leaders, SP receives a set of values denoted as $\{[\Gamma(d_1)]_1, \cdots, [\Gamma(d_n)]_1\}$ and computes the sum $\sum_{i=1}^{n}[\Gamma(d_i)]_1$. Conversely, the leader receives a set denoted as $\{[\Gamma(d_1)]_2, \cdots, [\Gamma(d_n)]_2\} \backslash [\Gamma(d_k)]_2$ and performs operations on them. As $[\Gamma(d_i)]_1 = \Gamma(d_i) - [\Gamma(d_i)]_1 \mod N$ ($i \in [1..n] \setminus k$), $[\Gamma(d_i)]_1$ represents a random number in $\mathbb{Z}_N$, and $\Gamma(d_i) \ll [\Gamma(d_i)]_1$, the leader must engage in more modular arithmetic compared to SP. Hence, the leader consumes a greater amount of running time than SP. Furthermore, in comparison to common participants, SP performs fewer operations, resulting in lower running time requirements. The running time consumption of both SP and the leader is positively correlated with the number of participants. In fact, as the number of participants increases, both SP and the leader require more time. Notably, the aggregation operation of variance consumes more running time than the other two operations due to its higher computational demands.

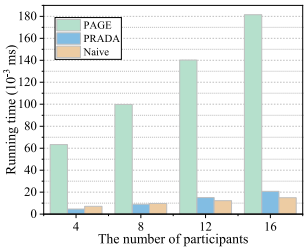
On the whole, in our experimental settings, the privacy-preserving aggregation protocols proposed within PRADA demonstrate exceptional efficiency, requiring only a few microseconds to output aggregation results. Thus, PRADA stands out as a highly efficient and privacy-preserving data aggregation scheme for MCS systems.

To further demonstrate the efficiency of the proposed PRADA, we execute comparative experiments with Naive and PAGE [25]. Considering fairness, instead of fully homomorphic encryption adopted by PAGE, we utilized Beaver triples [46] to enable secure multiplication for additive secret sharing. Specifically, we evaluated the encryption time between PAGE and PRADA for different aggregation operations. The results in Fig. 6(a) show that PRADA significantly reduces running time compared to PAGE, achieving more than an order of magnitude savings for sum and mean aggregations. Although PAGE transmits secret shares to only two sensing platforms and does not require participants to perform aggregation operations, it still needs to share shares of each sensing data based on additive secret sharing. In contrast, our novel approach, CrowdFA-based PRADA, allows participants to perform aggregation operations locally and submit local aggregation results only

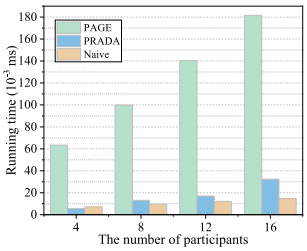


(a) Encryption

(b) Sum aggregation



(c) Mean aggregation



(d) Variance aggregation

Fig. 6. Comparisons of computation costs among PAGE, Naive, and PRADA.

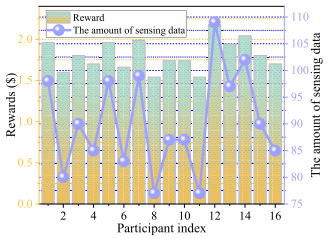
TABLE II
COMPARISONS OF COMMUNICATION COST AMONG PAGE, NAIVE, AND PRADA

	Sum		Mean		Variance	
	Ptcpt	Pltfm	Ptcpt	Pltfm	Ptcpt	Pltfm
PAGE	$2m\|N\|$	$2mn\|N\|$	$2m\|N\|$	$2mn\|N\|$	$2m\|N\|$	$16mn\|N\|$
Naive	$m\|x\|$	$mn\|x\|$	$m\|x\|$	$mn\|x\|$	$m\|x\|$	$mn\|x\|$
PRADA	$2\|N\|$	$n\|N\|$	$4\|N\|$	$2n\|N\|$	$6\|N\|$	$6n\|N\|$

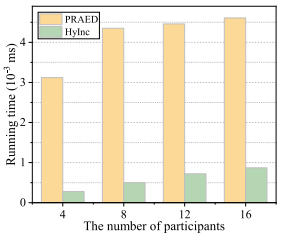
Note. Ptcpt: Participant; Pltfm: Platform; m represents a participant's amount of sensing data; n represents the number of participants; x indicates one sensing data; $|\cdot|$ means the length in bits.

to SP. Therefore, PRADA only requires secret sharing of the local aggregation results rather than all sensing data. This approach becomes particularly advantageous when participants possess multiple sensing data, resulting in fewer secret sharing operations and reduced computation costs. Furthermore, although PRADA requires participants to perform additional local operations for variance aggregation, note that the computation cost of PAGE is also $3-4$ times higher than that of PRADA. Hence, we can conclude that PRADA significantly reduces each participant's encryption cost by enabling them to perform aggregation operations locally.

Also, we carefully compare the running time among PAGE, Naive, and PRADA for sum aggregation, mean aggregation, and variance aggregation. As depicted in Fig. 6(b)-(d), despite PAGE adopts the more efficient Beaver triples to implement secure multiplication, its running time is still $5-22$ times that of PRADA. One possible explanation is that PRADA, based on CROWDFA, distributes aggregation operations to participants, thereby reducing the computation burden on SP. Additionally, each participant conducts aggregation operations directly on raw sensing data, resulting in higher efficiency. Experimental results shown in Fig. 6 also demonstrate that PRADA's running time is comparable to Naive. There are



(a) Reward



(b) Computation costs

Fig. 7. Performance of the proposed PRAED.

two potential reasons for this. Firstly, all participants in PRADA perform aggregation operations locally, mirroring SP's direct aggregation of participants' sensing data. Secondly, PRADA's use of additive secret sharing proves highly efficient in generating sums of secret shares.

In particular, we conduct an analysis of communication costs among PAGE, Naive, and PRAED, which are then presented in Table II. The table clearly illustrates that the communication costs for entities in PRADA remain unaffected by the participant's amount of sensing data. However, in the case of PAGE and Naive, these costs exhibit a positive correlation with the participant's amount of sensing data. When participants are required to collect multiple sensing data for an MCS task (e.g., a continue sensing task), PRAED outperforms PAGE in terms of communication costs. Therefore, PRADA represents an optimization of communication performance for privacy-preserving data aggregation in MCS systems.

Finally, we evaluate the performance of the proposed PRAED. When $B = \$28.88$, the relationship between a participant's rewards and their amount of sensing data is shown in Fig. 7(a). It is evident that participants with more sensing data receive higher rewards, thus demonstrating the fairness of the proposed PRAED. The experimental results depicted in Fig. 7(a) further demonstrate the effective winner localization capability of PRAED.

In terms of efficiency, `HyInc` [41] outperforms PRAED, as shown in Fig. 7(b). However, `HyInc` fails to protect bid privacy [18], [27], as it computes each participant's reward using their respective bids. In contrast, PRAED protects each participant's bid and computes rewards using secret shares. Consequently, PRAED incurs higher computational costs for distributing rewards to participants. Nevertheless, PRAED still achieves a running time of several microseconds, affirming its status as a privacy-preserving and efficient incentive design.

VII. CONCLUSION

In this paper, we proposed CROWDFA to address the challenges of incentive design, data aggregation, and privacy concerns simultaneously in mobile crowdsensing. CROWDFA achieves privacy-preserving data aggregation and privacy-preserving incentives based on additive secret sharing only within a unified framework. We formulated CROWDFA as a multiparty computing paradigm. In particular, we proposed a c-based privacy-preserving data aggregation scheme (PRADA) supporting three privacy-preserving aggregation protocols

(SumAgg, MenAgg, and VarAgg) within PRADA. Additionally, we designed a privacy-preserving incentive mechanism (PRAED) as another main building block of CROWDFA to achieve truthful and fair incentives for mobile crowdsensing. Experimental evaluations demonstrated the high efficiency of CROWDFA. For future work, we will explore non-linear privacy-preserving data aggregation protocols to extend the applicability of CROWDFA.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the reviewers for their valuable comments and suggestions.

REFERENCES

[1] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, 3rd Quart., 2019.

[2] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "iTAM: Bilateral privacy-preserving task assignment for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3351–3366, Dec. 2021.

[3] F.-J. Wu and G. Solmaz, "CrowdEstimator: Approximating crowd sizes with multi-modal data for Internet-of-Things services," in *Proc. 16th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2018, pp. 337–349.

[4] Z. Yu, H. Ma, B. Guo, and Z. Yang, "Crowdsensing 2.0," *Commun. ACM*, vol. 64, no. 11, pp. 76–80, 2021.

[5] A. U. Nambi, I. Mehta, A. Ghosh, V. Lingam, and V. N. Padmanabhan, "ALT: Towards automating driver license testing using smartphones," in *Proc. 17th Conf. Embedded Networked Sensor Syst.*, Nov. 2019, pp. 29–42.

[6] X. Zhang et al., "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.

[7] R. Zhou, R. Zhang, Y. Wang, H. Tan, and K. He, "Online incentive mechanism for task offloading with privacy-preserving in UAV-assisted mobile edge computing," in *Proc. 23rd Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Oct. 2022, pp. 211–220.

[8] M. Karaliopoulos, I. Koutsopoulos, and L. Spiliopoulos, "Optimal user choice engineering in mobile crowdsensing with bounded rational users," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1054–1062.

[9] J. V. Jeyakumar, L. Lai, N. Suda, and M. Srivastava, "SenseHAR: A robust virtual activity sensor for smartphones and wearables," in *Proc. 17th Conf. Embedded Networked Sensor Syst.*, Nov. 2019, pp. 15–28.

[10] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2735–2749, 2020.

[11] Z. Wang et al., "When mobile crowdsensing meets privacy," *IEEE Commun. Mag.*, vol. 57, no. 9, pp. 72–78, Sep. 2019.

[12] C. Zhang, M. Zhao, L. Zhu, T. Wu, and X. Liu, "Enabling efficient and strong privacy-preserving truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3569–3581, 2022.

[13] Y. Ren et al., "Towards privacy-preserving spatial distribution crowdsensing: A game theoretic approach," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 804–818, 2022.

[14] H. Jin, L. Su, and K. Nahrstedt, "CENTURION: Incentivizing multi-requester mobile crowd sensing," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[15] 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.

[16] Y. Qu et al., "Posted pricing for chance constrained robust crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 1, pp. 188–199, Jan. 2020.

[17] B. Zhao, S. Tang, X. Liu, and X. Zhang, "PACE: Privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1924–1939, May 2021.

[18] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2016, pp. 344–353.

[19] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 2053–2061.

[20] J. Sun and H. Ma, "Privacy-preserving verifiable incentive mechanism for online crowdsourcing markets," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–8.

[21] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Comput. Netw.*, vol. 102, pp. 157–171, Jun. 2016.

[22] I. Vakilinia, J. Xin, M. Li, and L. Guo, "Privacy-preserving data aggregation over incomplete data for crowdsensing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[23] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, "Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[24] Q. Li, G. Cao, and T. F. L. Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 2, pp. 115–129, Mar. 2014.

[25] X. Yan, B. Zeng, and X. Zhang, "Privacy-preserving and customization-supported data aggregation in mobile crowdsensing," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19868–19880, Oct. 2022.

[26] L. Yang, M. Zhang, S. He, M. Li, and J. Zhang, "Crowd-empowered privacy-preserving data aggregation for mobile crowdsensing," in *Proc. 18th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2018, pp. 151–160.

[27] L. Zhang, T. Zhu, P. Xiong, W. Zhou, and P. S. Yu, "More than privacy: Adopting differential privacy in game-theoretic mechanism design," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–37, Sep. 2022.

[28] Y. Liu, T. Feng, M. Peng, J. Guan, and Y. Wang, "DREAM: Online control mechanisms for data aggregation error minimization in privacy-preserving crowdsensing," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1266–1279, Mar. 2022.

[29] D. Wang, S. Shi, Y. Zhu, and Z. Han, "Federated analytics: Opportunities and challenges," *IEEE Netw.*, vol. 36, no. 1, pp. 151–158, Jan. 2022.

[30] M. Zhang, L. Yang, S. He, M. Li, and J. Zhang, "Privacy-preserving data aggregation for mobile crowdsensing with externality: An auction approach," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1046–1059, Jun. 2021.

[31] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable privacy-preserving aggregation in people-centric urban sensing systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 268–278, Sep. 2013.

[32] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 589–602, Jan. 2020.

[33] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 2045–2053.

[34] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by-hop data aggregation protocol for sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 4, pp. 1–43, 2008.

[35] M. Kim, A. Mohaisen, J. H. Cheon, and Y. Kim, "Private over-threshold aggregation protocols over distributed datasets," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2467–2479, Sep. 2016.

[36] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador, "Privacy-preserving mechanisms for crowdsensing: Survey and research challenges," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 855–869, Aug. 2017.

[37] J. Hu, K. Yang, K. Wang, and K. Zhang, "A blockchain-based reward mechanism for mobile crowdsensing," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 1, pp. 178–191, Feb. 2020.

[38] Q. Hu, Z. Wang, M. Xu, and X. Cheng, "Blockchain and federated edge learning for privacy-preserving mobile crowdsensing," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12000–12011, Jul. 2021.

[39] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, and N. Kourtellis, "PPFL: Privacy-preserving federated learning with trusted execution environments," in *Proc. 19th Annu. Int. Conf. Mobile Syst., Appl., Services*, Jun. 2021, pp. 94–108.

[40] D. Demmler, T. Schneider, and M. Zohner, "ABY—A framework for efficient mixed-protocol secure two-party computation," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–12.

[41] B. Zhao, X. Liu, W.-N. Chen, and R. Deng, "CrowdFL: Privacy-preserving mobile crowdsensing system via federated learning," *IEEE Trans. Mobile Comput.*, vol. 22, no. 8, pp. 4607–4619, Aug. 2022.

[42] G. Cardone et al., "Fostering participaction in smart cities: A geo-social crowdsensing platform," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.
[43] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao, and Y. Wang, "Scalable privacy-preserving participant selection for mobile crowdsensing systems: Participant grouping and secure group bidding," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 855–868, Apr. 2020.
[44] T. Wen, Y. Zhu, and T. Liu, "P2: A location privacy-preserving auction mechanism for mobile crowd sensing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[45] 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.
[46] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. Annu. Int. Cryptol. Conf.* Cham, Switzerland: Springer, 1991, pp. 420–432.

Bowen Zhao (Member, IEEE) received the Ph.D. degree in cyberspace security from the South China University of Technology, China, in 2020. He was a Research Scientist with the School of Computing and Information Systems, Singapore Management University, from 2020 to 2021. He is currently an Associate Professor with the Guangzhou Institute of Technology, Xidian University, Guangzhou, China. His current research interests include privacy-preserving computation and learning and privacy-preserving crowdsensing.

Xiaoguo Li received the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2019. He was a Post-Doctoral Research Fellow with Hong Kong Baptist University, Hong Kong, China, from 2019 to 2021. He is currently a Research Fellow with Singapore Management University, Singapore. His current research interests include privacy-preserving database outsourcing, public-key cryptography, and secure signal processing.

Ximeng Liu (Senior Member, IEEE) received the B.Sc. degree in electronic engineering and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2010 and 2015, respectively. He is currently a Full Professor with the College of Computer Science and Data Science, Fuzhou University. Also, he was a Research Fellow with the Peng Cheng Laboratory, Shenzhen, China. He has published more than 200 papers on the topics of cloud security and big data security, including papers of IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, and IEEE INTERNET OF THINGS JOURNAL. His research interests include cloud security, applied cryptography, and big data security. He received the "Minjiang Scholars" Distinguished Professor, "Qishan Scholars" in Fuzhou University, and ACM SIGSAC China Rising Star Award in 2018.

Qingqi Pei (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science and cryptography from Xidian University, in 1998, 2005, and 2008, respectively. He is currently a Professor and a member of the State Key Laboratory of Integrated Services Networks, also a Professional Member of ACM, and a Senior Member of the Chinese Institute of Electronics and China Computer Federation. His research interests focus on digital contents protection and wireless networks and security.

Yingjiu Li (Member, IEEE) is currently a Ripple Professor with the Computer and Information Science Department, University of Oregon. He has published over 140 technical papers in international conferences and journals and served in the program committees for over 80 international conferences and workshops, including top-tier cybersecurity conferences and journals. His research interests include the IoT security and privacy, mobile and system security, applied cryptography and cloud security, and data application security and privacy.

Robert H. Deng (Fellow, IEEE) is currently an AXA Chair Professor in cybersecurity and the Director of the Secure Mobile Centre, School of Computing and Information Systems, Singapore Management University. His research interests include applied cryptography, data security and privacy, and network security.