# Privacy-Preserving Data Aggregation over Incomplete Data for Crowdsensing

Iman Vakilinia*, Jiajun Xin*, Ming Li*, Linke Guo†
* Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
* Department of Electrical and Computer Engineering, Binghamton University,
State University of New York, Binghamton, NY 13902, USA

***Abstract*—Crowdsensing recently attracts great attention from both industry and academia. By fusing and analyzing multi-dimensional sensing data collected from crowdsensing users, it is possible to support health caring, environment mentoring, traffic mentoring and social behavior mentoring. Nonetheless, how to preserve users' data privacy during data fusing, e.g., data aggregation, has been rarely discussed for crowdsensing before. Besides, due to the dynamics of sensing environments and available resources at users, there will be missing elements from users' sensing results. In this paper we aim to achieve privacy-preserving data aggregation over incomplete data for crowdsensing. A novel scheme is developed based on linear transformation and homomorphic encryption scheme. It enables the server to obtain aggregation results over recovered sensing results without learning their individual details. Security analysis and performance evaluation are conducted showing the effectiveness and efficiency of our scheme.**

***Index Terms*—Crowdsensing, privacy, data aggregation, incomplete data.**

## I. Introduction

Crowdsensing arises as a new sensing paradigm based on powerful capabilities of current mobile devices on sensing, computing, storage and communication. The increasing popularity of mobile devices, equipped with multiple sensors, such as GPS, microphone, camera, gyroscope, accelerometer etc., enables the ability to acquire local knowledge from individuals' surrounding environment. By fusing and analyzing the multi-dimensional sensing data, it is possible to facilitate the development of health caring, environment monitoring, traffic monitoring, and social behavior monitoring.

Recently, there has been a substantial growth of research interest in crowdsensing. Various crowdsensing systems have been developed, including (vehicle) traffic monitoring/prediction [1], localization [2], parking space allocation/searching [3], and ambient (e.g.,dust level) surveillance [4]. Meanwhile, in order to stimulate users to participate in crowdsensing, some works focus on designing effective incentive mechanisms by applying Stackelberg game, contract theory, and auction theory [5]–[8]. On the other hand, since a sensing result may contain its owner's critical information such as visiting history, commute routes, habits, and preferences, revealing users' sensing results to the server will violate their privacy. Crowdsensing will not be widely adopted unless its privacy issue is properly addressed. Recently, Kong *et al.* [9] discuss the problem of trajectory recovery in crowdsensing with users' location privacy protected. Considering that data aggregation is an important and fundamental operation for data fusing in crowdsensing, e.g., calculating the first- and second-order statistics over sensing data, in this work we study a more general problem, i.e., privacy-preserving data aggregation for crowdsensing compared with [9] .

Even though the general privacy-preserving data aggregation problem has been extensively studied, e.g., in smart grids [10]–[12], sensor networks [13]–[15], etc., it has different challenges in crowdsensing. Due to the dynamics of sensing environments and the available resources at users, there will be missing data from users' sensing results in general cases. For instance, users may experience energy outage occasionally, when sensing data are unavailable. If directly applying privacy-preserving data aggregation schemes over the incomplete data set, computation accuracy cannot be guaranteed. On the other hand, it is observed that sensing data collected from similar geographic area regarding the same sensing event are strongly correlated. For instance, the temperatures observed at multiple locations within one city are closely the same. Vehicles on the same highway section demonstrate similar velocities. Leveraging such correlations, we propose to first recover the missing data from sensing results by using matrix completion algorithms [16], [17], and then perform data aggregation. The challenge lies in the fact that matrix completion algorithms work by exploiting correlations between different matrix entries (i.e., sensing results from different users in this work). Also due to this reason, existing solutions for privacy-preserving data aggregation do not work here.

In this work we aim to realize privacy-preserving data aggregation over incomplete data for crowdsensing. The proposed scheme consists of four components, *system initialization*, *sensing data masking*, *matrix recovery*, and *data aggregation*. The basic idea is described as follows. In *sensing data masking*, each user masks his original sensing data by applying linear transformation. Masked data ensure that the server cannot learn their plaintext. Besides, as masked data share the same correlations as original sensing data, the server is able to apply existing matrix completion algorithms to recover missing elements during *matrix recovery*. Finally, in

*data aggregation* we develop a novel algorithm based on the homomorphic encryption scheme [18], allowing the server to obtain aggregated results of the full original sensing data set.

This paper is organized as follows. Next section reviews major works in privacy-preserving data aggregation and crowdsensing. In Section III and IV, we introduce preliminaries and our system model, respectively. Details of our proposed scheme are described in Section V, followed by Section VI privacy analysis and Section VII performance evaluation. We conclude our paper in Section VIII.

## II. Related Work

### A. Crowdsensing

The current research on crowdsensing mainly falls in the area of system development and incentive mechanism design. For system development, Ganti *et al.* [1] design a navigation application by using participatory sensing data to map fuel consumption on city streets, allowing drivers to find the most fuel-efficient routes. Chon *et al.* [2] propose to exploit captured images and audio clips from smartphones to link place visits with place categories. A mobile system ParkNet [3] is developed comprising vehicles that collect parking space occupancy information while driving by. Lee *et. al* [4] develop an ambience monitoring platform, which addresses the energy problem through opportunistic cooperation among nearby mobile users. Meanwhile, in order to stimulate users to participate in crowdsensing, some works focus on designing effective incentive mechanisms by applying Stackelberg game, contract theory, and auction theory [5]–[8].

Unlike the above works, we deal with privacy issues in crowdsensing. Recently, Kong *et al.* [9] discuss the problem of trajectory recovery in crowdsensing with the users' location privacy protected. Differently, we target at a more general yet fundamental problem, i.e., to protect users' data privacy while still allowing the server to conduct data aggregation in crowdsensing.

### B. Privacy-Preserving Data Aggregation

The traditional privacy-preserving data aggregation has been extensively studied. Its general goal is to allow the aggregator to conduct an aggregation function over participants' data, yet the aggregator or other participants cannot learn any useful information regarding data details. Main techniques include: *secret sharing*, *data masking*, and *homomorphic encryption*. For *secret sharing* [10], each user splits his data into random shares: one share for each member in the aggregation group. The user then sends the aggregated shares to the aggregator, who obtains the final aggregation result. Since the aggregator only receives aggregated values over random shares, it does not know the original data. The main idea of *data masking* [11], [13], [14] is that each user masks his data before uploading them to the aggregator. The masks are designed in a way such that aggregated masks can be canceled with each other. The approaches [12], [15] applying *homomorphic encryption* technique explore the property that operations over individual ciphertext of homomorphic encryption can result in the multiplication or addition over the plaintext.

All the above works assume that aggregation is conducted over perfect data set, i.e., there is no missing element. However, in this paper we discuss privacy-preserving data aggregation over incomplete data. The design challenges will be different.

## III. Preliminaries

### A. Building Blocks

**Pairing.** Let $\mathbb{G}$ and $\mathbb{G}_1$ be cyclic multiplicative groups of prime order $q$, where each group has unique binary representation. A pairing function, or a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$, has the following properties

1) Bilinearity: $\forall g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, $e(g^a, g^b) = e(g, g)^{ab}$.
2) Non-degeneracy: $e(g, g) \neq 1$.
3) Computability: There exists an efficient algorithm to compute $e$.

### B. Cryptographic Assumptions

- *Discrete Logarithm Problem (DLP)* : Let $g$, $g_1$ be two elements in $\mathbb{G}$, It is computationally intractable to find an integer $a$, such that $g_1 = g^a$.
- *Decisional Diffie-Hellman Assumption (DDH)*: Given $(g, g^a, g^b, g^c)$ for $g \in \mathbb{G}$ and $a, b, c \in {\mathbb{Z}_q}^*$, the following two probability distributions are computationally indistinguishable: $(g^a, g^b, g^c)$ and $(g^a, g^b, g^{ab})$.

## IV. System Model

### A. System Architecture

There are two types of entities in our crowdsensing system, the server and the crowdsening users, as shown in Fig. 1. In what follows, we describe their functions and interactions.

- **Users**. Crowdsensing users are the smart devices which sense and harvest large quantities of data of their surrounding environments. They then upload sensing data to the server for data aggregation. Since these data may contain their owners' critical information such as visiting history, commute routes, habits, and preferences, users do not want to share the plaintext of their sensing data with others during crowdsensing.
- **Server**. The server collects sensing data from users and then performs data aggregation over them.

Assume there are a set of users $\mathcal{U} = \{u_1, \ldots, u_i, \ldots, u_N\}$ in the system, where $N = |\mathcal{N}|$. As smart devices are equipped with various sensors such as gyroscope, barometer, accelerometer and etc., the sensing result from a user contains multi-dimensional features/values, each of which corresponds to the data collected by a specific sensor. Denoting by $A_i$ $(1 \leq i \leq N)$ the sensing vector from $u_i$, then $A_{ij}$ stands for the $j$-th $(1 \leq j \leq T)$ feature of $A_i$, where $T$ is the maximal number of sensible features from all users. We assume that

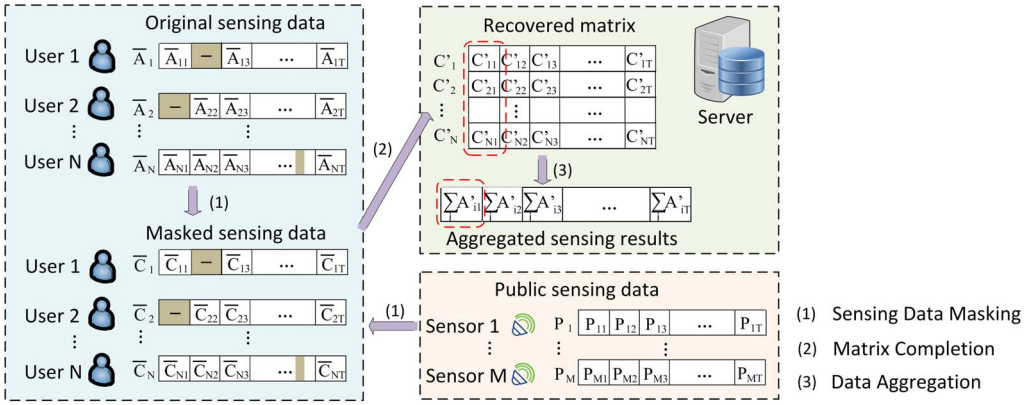



Fig. 1. The Overview of Our Proposed Scheme.

the sensing result takes value from a relative small integer set $\{0, 1, \dots V\}$[1], i.e., $0 \leq A_{ij} \leq V$. On the other hand, due to the dynamics of sensing environments and available resources at users, sensing results are incomplete in practical scenarios. We thus further denote by $\overline{A}_i$ the real sensing vector from $u_i$ with some entries empty.

### B. Design Objective

In this paper, we discuss how the server performs data aggregation over sensing results from all users for each feature. There are multiple kinds of aggregation functions, such as sum(), average(), maximum(), minimum(), etc. In this paper we focus on sum(), as it is one of the most commonly used function among all aggregation functions. Recall that $A_{ij}$ stands for the $j$-th ($1 \leq j \leq T$) feature of $A_i$. We aim to calculate $\sum_{1 \leq i \leq N} A_{ij}$ ($1 \leq j \leq T$), under the condition that 1) part of $A_{ij}$'s are missing; and 2) $A_{ij}$'s are hidden from the server.

The design objectives are summarized as follows.

- **Users Data Privacy**. The server should not be able to infer the individual sensing data from users, i.e., $A_{ij}$'s ($1 \leq i \leq N$, $1 \leq j \leq T$).
- **Aggregation Accuracy**. Despite the fact that some entries are missing in $\overline{A}_i$'s, the server can still calculate the approximate result for $\sum_{1 \leq i \leq N} A_{ij}$. Besides, the privacy-preserving property should not affect the aggregation accuracy.

In this work we assume that the server works under the semi-honest mode, i.e. it strictly follows the protocol but is curious about sensing data from users. Besides, as data privacy is one of the main concerns of users, they will not share their private data, including sensing results private keys, to the server for collusion purpose.

[1]Although our scheme operates over integers, the sensing data are not necessarily the case. To bridge this gap, we propose to have all users first transform their sensing data by multiplying $10^{\kappa}$, where $\kappa$ is the maximal number of digits in all sensing data. After the server calculates an aggregation result, it multiplies it with $10^{-\kappa}$ to obtain the original one.

## V. Our Proposed Scheme

In this section, we elaborate the construction of our proposed scheme, which consists of four components, *system initialization*, *sensing data masking*, *matrix recovery*, and *data aggregation*. Before describing each component, we first give a brief overview of the scheme.

### A. Overview

In order to achieve computation accuracy, rather than directly performing data aggregation over $\overline{A}_i$'s, we propose to recover their missing data first. Since $\overline{A}_i$'s are correlated, i.e., $A$ is a low-rank matrix, as discussed before, we are able to utilize matrix completion algorithms [16], [17] to recover it and obtain $A'_i$'s, with $A'_i \approx A_i$. Thereafter, data aggregation is performed over $A'_i$'s. On the other hand, as we also aim to protect $A_i$'s ($A'_i$'s) from the server, each user $u_i$ is proposed to first mask its original sensing vector $\overline{A}_i$ into $\overline{C}_i$ in *sensing data masking*. Then the server obtains $C'_i$, which is the masked form of $A'_i$ via *matrix recovery*. Finally, in *data aggregation* we develop an algorithm based on the homomorphic encryption scheme [18] allowing the server to obtain $\sum_{1 \leq i \leq N} A'_{ij}$ ($1 \leq j \leq T$) from $C'_i$'s. Fig. 1 illustrates the overview of our scheme. Some important notations are summarized in Table I.

### B. System Initialization

At the beginning, all users agree on a set of public bilinear parameters $(q, \mathbb{G}, \mathbb{G}_1, e, g)$. Each user $u_i$ chooses one of his secret keys $k_i \in \mathbb{Z}_q^*$ and a random number $r_i \in \mathbb{Z}_q^*$. Users collaboratively compute $K = \prod_{1 \leq i \leq N} k_i$ without revealing their secret keys to others by adopting the method introduced in [19]. $u_i$ then computes his public key as $pk_i = g^{K/k_i} \in \mathbb{G}$ and a public value $e(g, g)^{r_i} \in \mathbb{G}_1$.

### C. Sensing Data Masking

In our crowdsensing system, there also exist a set of public sensing data $\{P_1, \dots, P_m, \dots, P_M\}$, where a sensing vector $P_m$ consists of $T$ elements $P_{mj}$'s ($1 \leq j \leq T$) and $M << N$.

TABLE I
IMPORTANT NOTATIONS

| Symbol | Definition |
|---|---|
| $N$ | Number of users in the system |
| $T$ | Number of features of the sensing result |
| $M$ | Number of public sensing vectors |
| $V$ | The maximal value of sensing data |
| $A_i$/$A_{ij}$ | Perfect sensing vector of $u_i$/The $j$-th element of $A_i$ |
| $\overline{A}_i$/$\overline{A}_{ij}$ | Incomplete sensing vector of $u_i$/The $j$-th element of $\overline{A}_i$ |
| $A'_i$/$A'_{ij}$ | Recovered sensing vector of $u_i$/The $j$-th element of $A'_i$ |
| $\overline{C}_i$/$\overline{C}_{ij}$ | Masked value of $\overline{A}_i$/The $j$-th element of $\overline{C}_i$ |
| $C'_i$/$C'_{ij}$ | Recovered value of $\overline{C}_i$/The $j$-th element of $C'_i$ |
| $P_m$ | Public sensing vector from the $m$-th sensor |
| $P_{mj}$ | The $j$-th element of $P_m$ |
| $\overline{P}^i_m$ | $P_m$ with null entries whose positions are the ones where data are missing in $\overline{A}_i$ |
| $\overline{P}^i_{mj}$ | The $j$-th element of $\overline{P}^i_m$ |
| $\Delta_i$ | Set of missing data indices of $\overline{A}_i$ |

They are collected by $M$ sensors that are deployed by the server and shared with users. Since these sensors have stronger sensing capabilities than smart devices, we consider that their sensing results are complete.

In this component, $u_i$ first picks a set of secret keys $s_{im}$'s ($1 \leq m \leq M$) in $\mathbb{Z}_q$ following any arbitrary probability distribution, keeping the corresponding probability distribution function (PDF) private. In addition, $s_{im}$'s should be chosen in such a way that $\overline{C}_{ij}$ (as calculated following (1)) is larger than $q \cdot V$, where $\overline{C}_{ij}$ stands for the $j$-th element of $\overline{C}_i$. We discuss the reason for these requirements in Section VI. $u_i$ further constructs vectors $\overline{P}^i_m$'s ($1 \leq m \leq M$) based on $P_m$'s with some null entries, whose positions are the ones where data are missing in $\overline{A}_i$. Following the similar approach in [9], $u_i$ calculates his masked data as

$$\overline{C}_{ij} = k_i \cdot \overline{A}_{ij} + \sum_{m=1}^{M} s_{im} \cdot \overline{P}^i_{mj}, \ (1 \leq j \leq T) \tag{1}$$

i.e., a linear combination of $\overline{A}_{ij}$ and $\overline{P}^i_{mj}$'s. $u_i$ also prepares a set of parameters $\mu_{ij}$'s ($1 \leq j \leq T$) that will assist the server to compute the aggregation later. Specifically, $u_i$ first computes

$$\phi_{ij} = (e(g,g)^{\sum_{m=1}^{M} s_{im} \cdot P_{mj}})^{K/k_i}.$$

Then it computes the parameter $\mu_{ij}$ as

$$\mu_{ij} = \phi_{ij} \cdot (e(g,g)^{r_{i+1}} / e(g,g)^{r_{i-1}})^{r_i},$$

where $e(g,g)^{r_{i+1}}$ and $e(g,g)^{r_{i-1}}$ are received from user $u_{i+1}$ and $u_{i-1}$ respectively. In our scheme, all users are arranged in a ring. Thus $u_{i+1}$ and $u_{i-1}$ stand for the next and the previous user of $u_i$ in the ring, respectively. Besides, the next user of $u_N$ is $u_1$. The ring can be formed according to the lexicographical order of users' MAC addresses. Due to the construction of $\mu_{ij}$, we have

$$\begin{aligned}\prod_{i=1}^{N} \mu_{ij} &= \prod_{i=1}^{N} (\phi_{ij} \cdot (e(g,g)^{r_{i+1}} / e(g,g)^{r_{i-1}})^{r_i}) \\ &= \prod_{i=1}^{N} \phi_{ij} \cdot \prod_{i=1}^{N} (e(g,g)^{r_{i+1}} / e(g,g)^{r_{i-1}})^{r_i} = \prod_{i=1}^{N} \phi_{ij}.\end{aligned}$$

Finally, $u_i$ sends $\{\{\overline{C}_{ij}\}, \{\mu_{ij}\}, pk_i\}$ to the server. We will discuss the reason $u_i$ sending $\mu_{ij}$ instead of $\phi_{ij}$ later.

### D. Matrix Completion

Upon receiving masked sensing vectors from users, the server constructs $[\overline{C}_1; \ldots; \overline{C}_N; P_1; \ldots; P_M]$, which is also a low-rank matrix. Besides, for a particular row $\overline{C}_i$ ($1 \leq i \leq N$), it is a linear combination of $\overline{A}_i$ and $\overline{P}^i_m$'s according to (1). The server runs the matrix completion algorithm [16] to recover the missing data and obtains a matrix $[C'_1; \ldots; C'_N; P_1; \ldots; P_M]$, where $C'_i = k_i \cdot A'_i + \sum_{m=1}^{M} s_{im} \cdot P_m$. Here $A'_i$ is the recovered $\overline{A}_i$. Denote by $\Delta_i$ the set of missing data indices of $\overline{A}_i$. It is proved in [9] that $A'_{ij} = A_{ij}$ ($\forall j \notin \Delta_i$) and $A'_{ij} \approx A_{ij}$ ($\forall j \in \Delta_i$), where $A'_{ij}$ is the $j$-th element of $A'_i$. Notice that the difference between $A'_{ij}$ and $A_{ij}$ is introduced by the matrix completion algorithm, rather than the masking operation.

### E. Data Aggregation

Finally, the problem becomes how the server calculates $\sum_{1 \leq i \leq N} {A'_{ij}}^2$ with the knowledge of $C'_{ij}$'s. For this purpose, the server first computes $E_{ij} = e(pk_i, g^{C'_{ij}})$ ($1 \leq i \leq N$). Then it computes

$$\begin{aligned}&\prod_{i=1}^{N} E_{ij} / \prod_{i=1}^{N} \mu_i = \prod_{i=1}^{N} e(pk_i, g^{C'_{ij}}) / \prod_{i=1}^{N} \mu_{ij} \\ =&\prod_{i=1}^{N} e(g^{K/k_i}, g^{k_i \cdot A'_{ij} + \sum_{m=1}^{M} s_{im} \cdot P_{mj}}) / \prod_{i=1}^{N} \mu_{ij} \\ =&(\prod_{i=1}^{N} e(g,g)^{K \cdot A'_{ij} + K/k_i \cdot \sum_{m=1}^{M} s_{im} \cdot P_{mj}}) / \prod_{i=1}^{N} \mu_{ij} \\ =&(\prod_{i=1}^{N} e(g,g)^{K \cdot A'_{ij}} \cdot \prod_{i=1}^{N} e(g,g)^{K/k_i \cdot \sum_{m=1}^{M} s_{im} \cdot P_{mj}}) / \prod_{i=1}^{N} \mu_{ij} \\ =&(\prod_{i=1}^{N} e(g^K, g^{A'_{ij}}) \cdot \prod_{i=1}^{N} \phi_{ij}) / \prod_{i=1}^{N} \mu_{ij} \\ =&\prod_{i=1}^{N} e(g^K, g^{A'_{ij}}) = e(g^K, g^{\sum_{i=1}^{N} A'_{ij}}).\end{aligned}$$

Since values of $A'_{ij}$'s are from a small set, the server performs the exhaustive search to find out which value of $\sum_{i=1}^{N} A'_{ij}$

[2]For analysis simplicity, in this work we only discuss how to obtain $\sum_{1 \leq i \leq N} A'_{ij}$. In fact, the server can further calculate $\sum_{1 \leq i \leq N} A'_{ij} + \sum_{1 \leq m \leq M} P_{mj}$, i.e., the data aggregation over both users' and sensors' sensing data without difficulty.

having the above equation hold. As $A'_{ij} = A_{ij}$ ($\forall j \notin \Delta_i$) and $A'_{ij} \approx A_{ij}$ ($\forall j \in \Delta_i$), we have $\sum_{i=1}^{N} A'_{ij} \approx \sum_{i=1}^{N} A_{ij}$.

**Remark** Now we show that if $u_i$ sends $\mu_{ij}$ instead of $\phi_{ij}$ to the server in *sensing data masking*, the server can derive $A'_{ij}$. Since $E_{ij}$ can be expressed as

$$\begin{aligned} E_{ij} &= e(pk_i, g^{C'_{ij}}) \\ &= e(g^{K/k_i}, g^{k_i \cdot A'_{ij} + \sum_{m=1}^{M} s_{im} \cdot P_{mj}}) \\ &= e(g,g)^{K \cdot A'_{ij} + K/k_i \cdot \sum_{m=1}^{M} s_{im} \cdot P_{mj}} \\ &= e(g,g)^{K \cdot A'_{ij}} \cdot e(g,g)^{K/k_i \cdot \sum_{m=1}^{M} s_{im} \cdot P_{mj}} \\ &= e(g^K, g^{A'_{ij}}) \cdot \phi_{ij}, \end{aligned}$$

the server then has $E_{ij}/\phi_{ij} = e(g^K, g^{A'_{ij}})$. Because $A'_{ij}$ is from a small size set, $K$ is a public value and $E_{ij}$ is available at the server, it can derive $A'_{ij}$ by exhaustive search with the knowledge of $\phi_{ij}$.

## VI. Privacy Analysis

In this section, we analyze the privacy protection property of our proposed scheme.

**Theorem 1.** *There is no efficient way for the server to infer the plaintext of users' sensing results but random guessing.*

*Proof.* We sketch the proof as follows. Without loss of generality, we show that there is no efficient way for the server to learn $\overline{A}_{ij}$ from $\overline{C}_{ij}$. Regarding the sensing data $\overline{A}_{ij}$ from $u_i$, the only value that contains $\overline{A}_{ij}$ and is available at the server is $\overline{C}_{ij}$, which is calculated by $\overline{C}_{ij} = k_i \cdot \overline{A}_{ij} + \sum_{m=1}^{M} s_{im} \cdot \overline{P}^{i}_{mj}$. First of all, recall that $k_i$ and $s_{im}$ ($1 \le m \le M$) are private keys picked by $u_i$ within $\mathbb{Z}_q^*$ and $\mathbb{Z}_q$, respectively. Thus, the server cannot determine the exact value of $\overline{A}_{ij}$ from $\overline{C}_{ij}$, even though the values $\overline{P}^{i}_{mj}$'s are available. Observing that the only part contains $\overline{A}_{ij}$ in $\overline{C}_{ij}$ is $k_i \cdot \overline{A}_{ij}$, the server's viable strategy for guessing $\overline{A}_{ij}$ is to 1) analyze the distribution of $k_i \cdot \overline{A}_{ij}$; 2) then factorize each possible value of $k_i \cdot \overline{A}_{ij}$ and derive the distribution of $\overline{A}_{ij}$; 3) finally, identify the value with the highest probability as the best guess of $\overline{A}_{ij}$. On the other hand, in order to analyze the distribution of $k_i \cdot \overline{A}_{ij}$ in the first step, the server should know the distribution of either $k_i$ or $s_{im}$'s, both of which nonetheless are kept private at $u_i$. Thus the above-mentioned cryptanalysis cannot be successfully conducted. Together with the fact that $\overline{C}_{ij} > q \cdot V$ according to our scheme, the server can only know that $k_i \cdot \overline{A}_{ij}$ belongs to $[1, q \cdot V]$ without knowing its PDF. Therefore, there is no efficient way for the server to learn $\overline{A}_{ij}$ but random guessing. Similarly, we can prove that there is no efficient way for the server to learn $A'_{ij}$ from $C'_{ij}$. □

## VII. Performance Evaluation

We evaluate the performance of our proposed model in terms of computation and communication cost. The testbed is built on Linux Ubuntu 15.04 with 2.5GHz CPU and 4GB RAM. We use the Pairing Based Cryptography Library[3] for the implementation of pairing function in our scheme. The parameters of 160-bit long group order and 512-bit long base field is applied in "Type A" elliptic curve generator. Since the computation complexity mainly comes from pairing (paring), operations of exponentiation (exp), multiplication (mul), and finding inverse element (inv), the computation performance is evaluated based on them. All the experimental results represent the average of 10 trials. The default setting in our experiments are $N = 100$, $T = 10$, $V = 100$, and missing data ratio of users' sensing results is $10\%$.

TABLE II
Computation Time

| | System Initialization | Sensing Data Masking | Matrix Completion | Data Aggregation |
|---|---|---|---|---|
| user | $1.1ms$ | $23.3ms$ | – | – |
| server | – | – | $85.4ms$ | $217.3s$ |

### A. Computation Complexity

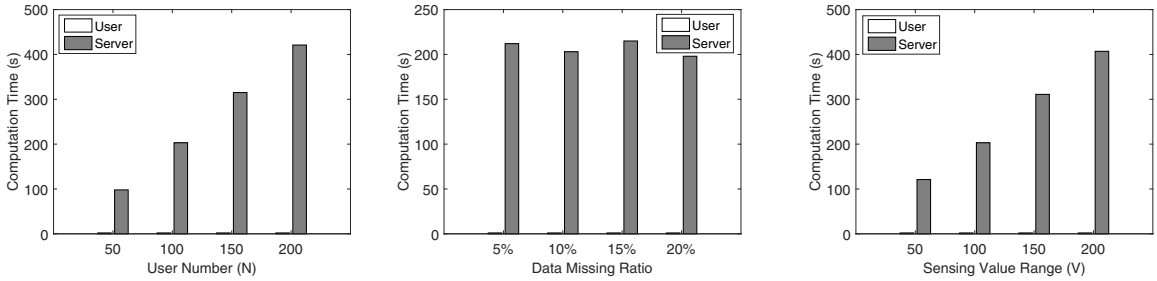
In the system initialization, each user's computation complexity contains: $1 \times$ paring, $(N+1) \times$ mul, $3 \times$ exp, and $1 \times$ inv. In the sensing data masking component, each user's computation complexity contains: $2T \times$ mul, $2T \times$ exp, and $T \times$ inv. In the matrix completion component, the server applies the matrix completion algorithm [16] to recover the masked matrix. In the data aggregation component, the server's computation complexity contains: $NT(V+1) \times$ pairing[4], $2T(N-1) \times$ mul, and $NT \times$ exp. Table II shows the computation time of a user and the server in each of four components. We find that the time used in data aggregation component is the highest, i.e., 217.3s. This is because the server has to conduct exhaustive search to find out $\sum_{i=1}^{N} A'_{ij}$. We further show the total computation time at both the user and server under different user number (Fig. 2(a)), data missing ratio (Fig. 2(b)), and sensing value range (Fig. 2(c)). We find that the computation load is light at SUs, while most of the computation is done at the server. This is a promising result as the server is more powerful in computing capability compared with mobile devices. Besides, as the number of user grows, the computation time at the server increases linearly. We also notice that missing data ratio does not influence the computation time for matrix recovery.

### B. Communication Cost

For this part, we analyze and show the communication cost in terms of the size of data payload transmitted. Since the server does not need to send out any data, we focus on the communication cost at users. In the system initialization, in order to obtain $K$, each user sends out $e(g,g)^{r_i}$ to two

[3] http://crypto.stanford.edu/pbc

[4] It contains $NT \times$ paring for calculating $NT$ $E_{ij}$'s, and $NV \times$ pairing (the worst case) in performing exhaustive search to find out $\sum_{i=1}^{N} A'_{ij}$ (thus $(N \cdot V \cdot T) \times$ pairing of all $\sum_{i=1}^{N} A'_{ij}$'s for $T$ features).




(a) (b) (c)

Fig. 2. Total computation time at the user and server.

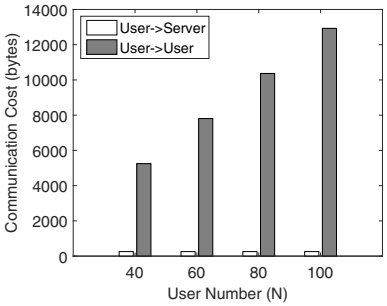



Fig. 3. Communication cost.

neighboring users and $k_i(e(g,g)^{r_{i+1}}/e(g,g)^{r_{i-1}})^{r_i}$ to the rest $N-1$ users, resulting in the communication cost of $128 \times (N-1) + 256$ bytes. In the masking sensing result component, note that users do not need to send $e(g,g)^{r_i}$ to two neighboring users, as this is done when obtaining $K$ during the system initialization. In this component, each user sends $\{\{\overline{C}_{ij}\}, \{\mu_{ij}\}, pk_i\}$ to the server, with the communication cost as $128 \times T + 20$ bytes. The communication cost is shown in the Fig. 3.

## VIII. Conclusion

In this paper we study the problem of privacy-preserving data aggregation over incomplete data for crowdsensing. A novel scheme is developed based on linear transformation and homomorphic encryption scheme. It enables the server to obtain aggregation results over recovered sensing results without learning their individual details. Security analysis and detailed performance evaluation are conducted showing the effectiveness and efficiency of our scheme.

## References


[1] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *MobiSys*. ACM, 2010, pp. 151–164.

[2] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 481–490.

[3] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *MobiSys*. ACM, 2010, pp. 123–136.

[4] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "Comon: cooperative ambience monitoring platform with continuity and benefit awareness," in *MobiSys*. ACM, 2012, pp. 43–56.

[5] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 173–184.

[6] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *INFOCOM*. IEEE, 2012, pp. 1701–1709.

[7] D. Zhao, X.-Y. Li, and H. Ma, "Omg: How much should i pay bob in truthful online mobile crowdsourced sensing?" *arXiv preprint arXiv:1306.5677*, 2013.

[8] P. Minder, S. Seuken, A. Bernstein, and M. Zollinger, "Crowdmanager-combinatorial allocation and pricing of crowdsourcing tasks with time constraints," in *Workshop on Social Computing and User Generated Content in conjunction with ACM Conference on Electronic Commerce (ACM-EC 2012)*, 2012.

[9] L. Kong, L. He, X.-Y. Liu, Y. Gu, M.-Y. Wu, and X. Liu, "Privacy-preserving compressive sensing for crowdsensing based trajectory recovery," in *ICDCS*. IEEE, 2015, pp. 31–40.

[10] F. D. Garcia and B. Jacobs, "Privacy-Friendly Energy-Metering via Homomorphic Encryption," in *Security and Trust Management (STM)*, 2010, pp. 226–238.

[11] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-Friendly Aggregation for the Smart-Grid," in *Privacy Enhancing Technologies Symposium (PETS)*, 2011, pp. 175–191.

[12] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications," *IEEE TPDS*, vol. 23, no. 9, pp. 1621–1631, 2012.

[13] E. Shi, T. H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. NDSS*, vol. 2, 2011, pp. 1–17.

[14] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "Pda: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM*. IEEE, 2007, pp. 2045–2053.

[15] Q. Li and G. Cao, "Efficient and privacy-preserving data aggregation in mobile sensing," in *ICNP*. IEEE, 2012, pp. 1–10.

[16] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[17] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[18] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of cryptography*. Springer, 2005, pp. 325–341.

[19] T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation," in *INFOCOM*. IEEE, 2013, pp. 2634–2642.