Privacy-Preserving Mobile Crowdsensing for Location-Based Applications

Jianbing Ni†, Kuan Zhang‡, Xiaodong Lin‡, Qi Xia§ and Xuemin (Sherman) Shen†
†Department of Electrical & Computer Engineering, University of Waterloo, Canada
‡Faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada
§Center of Cyber Security, University of Electronic Science and Technology of China, China
Email: {j25ni,k52zhang,sshen}@uwaterloo.ca, xiaodong.lin@uoit.ca, xiaqi@uestc.edu.cn

Abstract—Mobile crowdsensing is a new paradigm which explores the mobility and intelligence of mobile users to collect high-quality data from social events and phenomena for conducting complex sensing tasks. Nevertheless, privacy preservation and task allocation become main obstacles that need additional attention. To achieve accurate task allocation, it is inevitable to share some sensitive information of mobile users and customers, such as identities, location and points of interest. In this paper, we propose a privacy-preserving mobile crowdsensing framework (PPMC) for location-based applications to balance the trade-off between privacy preservation and task allocation. In PPMC, we develop a matrix-based location matching mechanism for the service provider to achieve location-based task allocation without disclosing the location of mobile users and the sensing area of tasks. We also extend BBS+ signature and proxy re-encryption to preserve identity privacy and data privacy for both customers and mobile users under the condition that they are honest to release and perform tasks, respectively. Finally, we discuss security properties and demonstrate the efficiency of PPMC in terms of computational and communication overhead.

I. INTRODUCTION

With the support of user-centric mobile sensing and computing devices, e.g., smartphones, in-vehicle sensing devices and wearable devices, our knowledge of the physical world is elaborated by opening up a new door to sense, collect and process the information from social events and phenomena. This alternative has triggered the emergence of mobile crowdsensing services [1]. In mobile crowdsensing, individuals cooperatively collect data based on the tasks released by customers and extract information to measure and map phenomena of common interest using their mobile devices [2].

Although mobile crowdsensing is appealing, it also brings new challenges towards customers and mobile users, e.g., privacy leakage, indicating that mobile crowdsensing puts the privacy of both mobile users and customers at stake [3], [4]. On one hand, the sensing data collected by mobile devices may inadvertently reveal personal information about the mobile users, including identity, location, habit, health status, daily route, and political affiliation. Furthermore, the more tasks mobile users engaged in and the richer data they contribute to, the higher possibility that their sensitive information is disclosed. For example, when a mobile user reports a medical experience of a department in a hospital, the service provider is aware of that this mobile user may have certain disease related to this department. Therefore, ensuring the privacy of mobile users is the first-order security concern in mobile crowdsensing. Without sufficient privacy preservation to protect the sensitive information, mobile users may not be willing to participate in mobile crowdsensing activities [5]. On the other hand, a releasing task may pose some sensitive information about the customer, e.g., purchase intention and points of interest. For instance, a house agency may know Bob desires to buy a house in a particular area if he collects the information about traffic condition and noise level in the neighborhood. Several privacy-preserving mobile crowdsensing schemes [6], [7] have been proposed to prevent privacy invasion for either mobile users or customers by utilizing anonymity techniques [8]. However, anonymization is insufficient for privacy preservation, since sensitive information may be still exposed to the nearby eavesdroppers via physical observation. Furthermore, the de-anonymity technique becomes pretty mature nowadays [8]. For example, the sensing data is usually cross-referenced with other sources of data, e.g., social graph and mobility patterns, such that attackers may identify the data source even the identity is invisible. According to [9], human mobility patterns can be predicted and an anonymous user can be identified from four data points. An anonymous sensing report may be easily linked to a specific mobile user by analyzing some public or available data over the Internet, especially in big data era. Therefore, it is critical to tackle the privacy problems in mobile crowdsensing.

Meanwhile, once the sensitive information about customers and mobile users, e.g., locations, references and reputations, is perfectly protected, it brings a new obstacle on task allocation from the service provider’s perspective [10]. Specifically, the set of the mobile users usually affects the quality of sensing results. An improper task allocation policy may trigger plenty of troubles on the mobile users, especially in location-based applications, since the participating mobile users have to stay in the sensing points to collect data [11]. One solution to optimize task allocation for location-based applications is to leverage spatial and temporal correlation of mobile users [12]. However, it discloses the points of interest of customers and the location of mobile users to the service provider, by which the anonymity of both customers and mobile users is invaded. Therefore, it is still challenging to effectively allocate.
crowdsensing tasks without privacy disclosure.

To achieve accurate task allocation and prevent privacy disclosure of customers and mobile users, in this paper, we propose a Privacy-Preserving Mobile Crowdsensing framework (PPMC) for location-based applications. The main contributions of this paper are two-fold.

- We propose a location-based task allocation method without disclosing location information to the service provider, preserving conditional privacy for customers and mobile users. By utilizing extended proxy encryption [13] and BBS+ signature [14], we allow the registered customers and mobile users to anonymously prove their capacity to participate in the crowdsensing services and securely perform the crowdsensing tasks without exposing contents of tasks and sensing reports. In addition, to prevent the mobile users from misbehaving for unfair reward, a trusted authority enables to detect greedy mobile users and trace their identities.

- We propose a matrix-based location matching approach to allow the service provider to allocate the crowdsensing tasks based on the sensing areas of tasks and the locations of mobile users. Specifically, the customer and the mobile user utilize a self-defined random matrix to hide the sensing area and the location, respectively. Thus, the service provider can determine whether the mobile user is located in the sensing area of the task without learning the points of interest of customer and the location of the mobile user.

The remainder of this paper is organized as follows. In section II, we present system model and threat model, and identify the design goals. In section III, we propose the PPMC in detail. We discuss the security properties of PPMC in section IV and evaluate performance in section V. We review related work in section VI and conclude the paper in section VII.

![System Model of Mobile Crowdsensing](attachment:image.png)

II. PROBLEM STATEMENT

In this section, we present system model, threat model, and identify our design goals.

A. System model

We consider a typical location-based mobile crowdsensing application (e.g., Gigwalk, mCrowd). The system model consists of four entities: an authority, a service provider, customers and mobile users, as shown in Fig. 1. The location-based mobile crowdsensing is performed in the following steps.

Step 1: Each customer or mobile user is required to register at the trusted authority in the initialization phase.

Step 2: To collect data in a specific area \( L \), a customer creates a sensing task \( ST \) and sends it to the service provider, along with the authentication message.

Step 3: The service provider offers mobile crowdsensing services and releases the received task for the customer.

Step 4: The registered mobile users \( U_i \), who are willing to preform sensing job, obtain their current or future location information via mobile devices that are capable of localization (e.g., by wireless access points or GPS) and send their locations to the service provider, along with the authentication information.

Step 5: Upon receiving the messages, the service provider firstly checks whether the mobile users are in the area \( L \) or they will be in the area \( L \) in the near future.

Step 6: Suppose the locations of the mobile users \( U_i \) can match the sensing area \( L \). The service provider then distributes the task \( ST \) to the mobile users \( U_i \).

Step 7: The mobile users can accept or reject the task according to rewards and costs of performing the task. Then, the mobile users, who accept the task, sense the event or phenomena, collect data, and generate sensing reports.

Step 8: The mobile users return the sensing reports to the service provider.

Step 9: The service provider forwards the sensing reports to the customer.

Step 10: The customer reads the reports and distributes rewards to the mobile users according to their contributions on the task.

B. Threat Model

In threat model, the service provider can honestly perform the mobile crowdsensing services but is curious about the privacy of customers and mobile users. The information obtained by the service provider (including the tasks issued by customers and the sensing reports submitted by mobile users) may contain plenty of knowledge, such as the identities of customers and mobile users, the places where the mobile users visit, or the area or events the customers are interested in. The service provider may build a spatio-temporal probability distribution for a specific mobile user and learn certain sensitive information about customers and mobile users, such as preference, social relation, political affiliation, and purchase intention. Although outside malicious attackers may also capture and exploit the sensitive information, the service provider is assumed not compromised to absurdly perform the operations, since the intrusion detection mechanisms and firewall are installed on the servers. In addition, we assume that the service provider would not collude with mobile users to invade the privacy of customers or other mobile users.

Mobile users are interested in the privacy of the customers and the other mobile users. In particular, they are willing to know the other mobile users who are participating...
in the same tasks to get more rewards, and learn more information about customers they are working for to satisfy the requirements of customers. The location is obtained from GPS chip in the mobile device or the access points, we assume that mobile users cannot modify their location information. Greedy mobile users may gain unfair award by double reporting the data in a time period, or collude together to send false reports to deceive the customer. However, the majority of mobile users are assumed to be honest.

Some external threats, such as eavesdroppers on the communication channels and hackers, also trigger serious security problems towards mobile crowdsensing services. For example, an adversary captures transmitting messages by wiretapping on the channels. Furthermore, it is easy for an attacker to obtain the identities of the nearby mobile users or customers via physical observation, such that the anonymity of customers and mobile users may be violated.

C. Design Goals

To enable privacy-preserving mobile crowdsensing under the aforementioned system model and resist the security threats, our system should achieve following objectives.

- Location-based task allocation: The releasing tasks should be allocated to the mobile users in the area (defined by the customers). Other mobile users outside the sensing area cannot learn any information about the tasks.
- Location privacy preservation: The location of mobile users and the sensing area of releasing tasks should not be exposed to others. The mobile users are only aware of whether their geographical positions are in the sensing area or not. The sensing area of a task is invisible to the service provider and the mobile users who are not in the sensing area.
- Confidentiality of releasing tasks and sensing reports: No entity, except the delegated mobile users (customers), can obtain the content of releasing tasks (sensing reports), such that the privacy of mobile users (customers) should not be disclosed to others.
- Anonymity of mobile users and customers: No entity (including customers, mobile users, the service provider, or colluded entities) can link a sensing report to a specific mobile user or link a releasing task to a specific customer. It is even impossible for an attacker to identify whether two sensing reports are generated by the same mobile user or whether two tasks are issued by the same customer.
- Greedy user tracing: Mobile users should be prevented from submitting more than one sensing report for the same task in a reporting period to obtain unfair rewards. The authority should trace the identity of greedy mobile users after receiving the transcripts of two different sensing reports.

III. OUR CONSTRUCTION

We propose our PPMC, which consists of four phases, setup, registration, task allocation and data reporting, based on the BBS+ signature \[\text{[14]}\] and the AFGH proxy re-encryption \[\text{[13]}\].

A. Setup

This phase is run by the authority and the service provider to bootstrap the mobile crowdsensing service. Let \((\mathbb{G}, \mathbb{G}_T)\) be two cyclic groups with a prime order \(p\), where \(p\) is \(\lambda\) bits, and \(\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T\) be a bilinear map. The authority picks random generators \(g, g_0, g_1, g_2, g_3, h \in \mathbb{G}\), and computes \(G = \hat{e}(g, g)\) and \(H = \hat{e}(h, h)\), respectively. The authority also chooses a random \(\mathbb{G} \in \mathbb{G}_T\), and defines a cryptographic hash function \(\mathcal{H} : \{0, 1\}^\star \rightarrow \mathbb{Z}_p^\star\) and a pseudo-random function \(\mathcal{F} : \mathbb{Z}_p \times \{0, 1\}^\star \rightarrow \mathbb{Z}_p\). The public parameters are \((\mathbb{G}, \mathbb{G}_T, p, g, g_0, g_1, g_2, g_3, h, G, H, \mathcal{G}, \mathcal{H}, \mathcal{F})\). Finally, the authority randomly chooses \(\alpha \in \mathbb{Z}_p^\ast\) as its secret key and calculates the public key \(T = g^\alpha\).

To setup the crowdsensing service, the service provider randomly chooses its secret key \(\beta \in \mathbb{Z}_p^\ast\) and computes its public key \(S = h^\beta\). It also employs a matrix \(L_{m \times n}\) to denote the geographical region (i.e., longitude and latitude) that the crowdsensing service can cover. Each entry in the matrix denotes a small grid in the sensing region. Given the longitude of Ontario is from 74.40°W to 95.15°W and the latitude is from 41.66°N to 57.00°N, we have a 208 × 154 matrix or a 2075 × 1534 matrix more precisely to represent it.

B. Registration

A customer or mobile user is required to register at the authority to obtain an anonymous credential, which is used to participate in the crowdsensing service. Each registrant (i.e., customer or mobile user) is assigned a unique identity \(I\), which can be telephone number or mailing address in practise. The registrant randomly picks \(s', a \in \mathbb{Z}_p^\ast\) to compute \(C = g_1^s g_2^a\), \(\hat{A} = g^a\), and sends \((I, C, \hat{A})\) to the authority, along with the following zero-knowledge proof:

\[\mathcal{PK}_1\{ (s', a) : C = g_1^s g_2^a \land \hat{A} = g^a \}\].

The authority firstly checks the proof \(\mathcal{PK}_1\) to ensure \((C, \hat{A})\) are generated properly. Then, the authority randomly picks \(s'', e \in \mathbb{Z}_p^\ast\) to calculate \(A = (g_0CG_1^{s''}g_2^e)^{1/m}\), \(RK = \hat{A}^\frac{1}{m}\), and returns \((A, s'', e, RK)\) to the registrant. The tuple \((I, \hat{A})\) is stored by the authority in its database.

The registrant computes \(s = s' + s''\) and checks

\[\hat{e}(A, T g^e) = \hat{e}(g_0g_1^s g_2^a g_3^e, g)\].

The registrant stores \((A, e, s, a, I, \hat{A}, RK)\) on his mobile device.

C. Task Allocation

The time is divided into slots, each of which ranges from minutes to months depending on the specific requirements of the task. A customer with registered information \((A, e, s, a, I, \hat{A}, RK)\) has a crowdsensing task to allocate and requests the sensing data from mobile users slot by slot. The statement of the task is denoted as \(ST = (\text{task}, \text{expires}, \text{area})\), indicating the content (what to sense), the expiration time (when to sense) and the sensing area (where to sense), respectively. Other attributes (e.g., sensing
interval, acceptance conditions, rewards, and reporting periods) can be added as well. We consider that the content task contains all these attributes. To protect the privacy of the task, the customer picks four randoms $k, r_1, r_2, r_3 \in \mathbb{Z}_p^*$, and calculates $u = g^k, c_1 = S^{r_2}, c_2 = T^{r_1}$ and $c_3 = (\text{task}||u)G^{r_1}H^{r_2}$. Then, the customer generates a matrix $\hat{L}_{m \times n}$ to denote the target sensing region area. As depicted in Fig. 2, the entry in $\hat{L}_{m \times n}$ corresponding to each position in the sensing area is a random value chosen from $\mathbb{Z}_p^*$; otherwise, the value for a location outside is set to be zero. To mask the sensing area in $\hat{L}_{m \times n}$, the customer picks $m \times n$ random values from $\mathbb{Z}_p^*$ to generate an invertible matrix $\tilde{M}_{m \times n}$ and computes $\tilde{N}_{n \times n} = \tilde{M}_{m \times n} \cdot \hat{L}_{m \times n}$, where $\tilde{M}_{m \times n}$ is the transpose of the matrix $\hat{L}_{m \times n}$. Note that all non-zero entries in $\tilde{N}_{n \times n}$ should be distinct, unless an attacker still can learn the sensing region $\tilde{N}_{n \times n}$ contains all these attributes. To protect the content privacy $N_{n \times n}$, the service provider calculates the non-zero matrix, $\tilde{N}_{n \times n}$, and calculates $\tilde{N}_{n \times n} = \tilde{M}_{m \times n} \cdot \hat{L}_{m \times n}$. Finally, the customer secretly keeps $k$ and sends $(c_1, c_2, c_3, expires, \tilde{N}_{n \times n})$ to the service provider, along with the following zero-knowledge proof.

$$\mathcal{PK}_2( (A, e, s, a, I) : \hat{e}(A, T^g) = \hat{e}(g_0 g_a^s g_2 g_{3,i}^l, g))$$

Then, the service provider checks the validity of the proof $\mathcal{PK}_2$. If yes, it assigns a task number $num$, releases $(num, expires)$ and stores $(num, c_1, c_2, c_3, expires, \tilde{N}_{n \times n})$ in its database.

When a mobile user $U_i \in R$ with $(A_i, e_i, s_i, a_i, I_i, \tilde{A}_i, RK_i)$ wants to participate in a crowdsensing task, $U_i$ first picks a random $\nu \in \mathbb{Z}_p^*$ to calculate $\mu = h^\nu$. Then, $U_i$ generates a matrix $\tilde{L}_{m \times n}$ according to its current location and the places it will visit. For each location where $U_i$ will reach, the corresponding entry in $\tilde{L}_{m \times n}$ is set as a random value chosen from $\mathbb{Z}_p^*$ and all the other entries are zero. The non-zero entries in $\tilde{L}_{m \times n}$ should be different. To protect this location information, it also generates a random invertible matrix $\tilde{M}_{m \times n}$ by picking $m \times n$ random values from $\mathbb{Z}_p^*$, and calculates $\tilde{N}_{n \times n} = \tilde{M}_{m \times n} \cdot \tilde{L}_{m \times n}$. Finally, $U_i$ keeps $\nu$ secretly and sends $(\mu, \tilde{N}_{n \times n})$ to the service provider, along with the following zero-knowledge proof:

$$\mathcal{PK}_3( (A_i, e_i, s_i, a_i, I_i) : \hat{e}(A_i, T^g) = \hat{e}(g_0 g_a^s g_2 g_{3,i}^l, g))$$

The service provider returns failure if the verification of $\mathcal{PK}_3$ outputs invalid; otherwise, for each unexpired task, it uses $\tilde{N}_{n \times n}$ to calculate $N_{n \times n} = \tilde{N}_{n \times n} \cdot \tilde{N}_{n \times n}$ and checks whether $N_{n \times n}$ is zero matrix or not. If $N_{n \times n}$ is non-zero matrix, $U_i$ successfully performs $ST$. Then, the service provider calculates $c_4 = \hat{e}(\mu, c_1)\hat{e}$ and releases $(num, c_2, c_3, c_4, expires)$ for $U_i$. If there is no task matching $U_i$, the service provider responds failure.

When obtaining $(num, c_2, c_3, c_4, expires)$, $U_i$ decrypts $(c_2, c_3, c_4)$ by using $(\nu, a_i)$, i.e., $\text{task}||u = c_3 c_1 \hat{e}(c_2, RK_i)\hat{e}$. Then, $U_i$ evaluates the task and determines to participate in or abandon this task according to the reward and cost. If $U_i$ accepts the task $ST$, it starts to perform the sensing work according to the details in task.

D. Data Reporting

$U_i$ collects the data $m_i$ and periodically submits a sensing report (including the collection time, the sensing location and the detailed content) to the customer. The reporting periods are defined by the customer. Denote $\tau_j$ as the current slot. To prevent attackers from learning $m_i$, $U_i$ uses $u$ to encrypt $m_i$ as $D_i = u^{\tau_i}, D'_i = m_i G^{\tau_i}$, where $\tau_i$ is randomly chosen from $\mathbb{Z}_p^*$. Then, $U_i$ computes $X_i = H(num||m_i||\tau_j), v_i = F_{a_i}(num||I||\tau_j), Y_i = H^{\tau_i}$ and $Z_i = \hat{e}(g, \tilde{A}_i)G^{\tau_i}v_i$. Finally, $U_i$ sends the report $(num, D_i, D'_i, X_i, Y_i, Z_i, \tau_j)$ to the service provider.

The service provider checks whether there is another report $(num, D_i, D'_i, X_i, Y_i, Z_i, \tau_j)$ that has the same $X_i$ but different $\tau_j$ with the new received report $(num, D_i, D'_i, X_i, Y_i, Z_i, \tau_j)$. If yes, the service provider computes $W = (Z_i^{X_i^{\tau_i}})^\frac{1}{\tau_i} Z_i^{X_i^{\tau_i}}$, and sends it to the authority. Then, the authority finds the mobile user’s identity $I_i$ by utilizing $A_i$ in the database to check whether $W = \hat{e}(g, A_i)$ or not, until it finds a match. Thus, the identity of the greedy mobile user $U_i^*$ is recovered by the authority if $U_i^*$ submits two different sensing reports in a single reporting slot.

When retrieving the reports, the customer decrypts them using the stored $k$ as $m_i = D'_i \hat{e}(g, D_i)\hat{e}$, and distributes the rewards according to the contributions of mobile users.

IV. ACHIEVED SECURITY GOALS

In this section, we explain the achieved security goals described in [II-C], including location privacy preservation, confidentiality of tasks and reports, anonymity of both mobile users and customers, and greedy user tracing.

A. Location Privacy Preservation

The location of the sensing region is represented as a matrix $\hat{L}_{m \times n}$, which is randomized by a random matrix $\tilde{M}_{m \times n}$ to generate $\tilde{N}_{n \times n}$. The location of the mobile user is transformed to be $\tilde{N}_{n \times n}$. Receiving these two matrices, the service provider cannot learn any information about the location of the mobile user and the sensing area of the customer from $\tilde{N}_{n \times n}$ and $\tilde{N}_{n \times n}$, respectively. The service provider computes $\tilde{N}_{n \times n} = \tilde{N}_{n \times n} \cdot \tilde{N}_{n \times n}$. If there is no overlapping in the crowdsensing area and location of users, $\tilde{N}_{n \times n}$ must be a zero matrix. If just one overlapping grid exists (where the corresponding entry is $\tilde{L}_{ij}$ in $\tilde{L}_{m \times n}$, and is $L_{ij}$ in $\hat{L}_{m \times n}$, respectively), the entries in $j$-row of $\tilde{N}_{n \times n}$ are nonzero, as well as the entries in $j$-column of $\tilde{N}_{n \times n}$. Then, the service provider knows that there are some overlapping locations in the $j$-column of the sensing area. But,
it is unable to distinguish which location is overlapped from \( m \) locations. Furthermore, \( \tilde{N}_{n,n} \cdot \tilde{N}_{n,n} \) and \( \tilde{N}_{n,n} \cdot \tilde{N}_{n,n} \) cannot provide more information to the service provider. The results are the same if the overlapping grids are more than one. Therefore, the sensing area and the location of mobile user are not exposed to the service provider and other entities.

B. Tasks Confidentiality

Having the location information matching the sensing area, the mobile users are able to decrypt the ciphertext of the task. In PPMC, the adversaries may be the service provider, unmatched mobile users and external attackers. To resist these adversaries, the protection of the task consists of two stages. In the first stage, the task is encrypted by the customer under the public keys of the authority and the service provider; in the second stage, the service provider decrypts the ciphertext using its secret key and then re-encrypts the result for the matched mobile users. Specifically, the security of the first-stage ciphertext can be reduced to the \( q \)-DBDH assumption \([13]\) and the security of the second-stage ciphertext depends on the AFGH proxy re-encryption scheme \([13]\).

C. Reports Confidentiality

To guarantee the confidentiality of reports, each mobile user employs the AFGH proxy re-encryption scheme \([13]\) to encrypt \( m_i \) under the temporary public key \( u = g^k \), which is distributed to the mobile users along with the task. The decryption key \( k \) is secretly kept by the customer. Therefore, the confidentiality of \( m_i \) directly depends on the semantic security of the AFGH proxy re-encryption scheme, which can be reduced to the simplified \( q \)-DBDH assumption \([13]\).

D. Anonymity

The anonymity of the mobile user is defined via the game in which the adversary cannot distinguish an honest mobile user out of two under the extreme condition that all other interactions are specified by the adversary. We demonstrate that the mobile user’s identity is preserved properly, unless the DDH assumption \([15]\) does not hold.

E. Greedy User Tracing

If there exist two sensing reports uploaded by the same mobile user in a time slot, it is easy to see that the service provider can compute a correct \( W \) and the authority is able to trace the identity of the greedy mobile user by checking the equation \( W = \hat{e}(g, A_i) \).

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our PPMC in terms of computational and communication overhead.

A. Computational Overhead

We demonstrate the computational overhead of the PPMC by counting the number of the basic cryptographic operations, such as point multiplication, point addition, bilinear map and exponentiation in \( \mathbb{G}_T \). Other operations, e.g., multiplication in \( \mathbb{G}_T \), addition, multiplication and inverse operations in \( \mathbb{Z}_p^* \), can be negligible in computational time. We utilize the pre-processing mechanism to reduce the computational burden for each entity. Specifically, the authority pre-computes the bilinear maps \( E_0 = \hat{e}(g_0, g) \), \( E_1 = \hat{e}(g_1, g) \), \( E_2 = \hat{e}(g_2, g) \), \( E_3 = \hat{e}(g_3, g) \), \( E_4 = \hat{e}(g, S) \) in the setup phase and \( \{\hat{e}(g, A_i)\}_{i=0}^{N} \) in registration phase, where \( N \) is the number of registrants in the system. The mobile user \( U_i \) also can pre-compute \( \hat{e}(g, A_i) \) in the registration phase. Table I shows the number of the operations required in each phase of PPMC, respectively.

We also conduct an experiment to show the efficiency of PPMC. The operations of authority and service provider are performed on a notebook with Intel Core i5-4200U CPU (the clock rate is 2.29GHz) and 4.00GB memory. The operations of customers and mobile users are run on HUAWEI MT2-L01 smartphone with Kirin 910 CPU and 1250M memory. The operation system is Android 4.2.2 and the toolset is Android NDK r8d. We use MIRACL library 5.6.1 to implement number-theoretic based methods of cryptography. The Weil pairing is utilized to realize the bilinear pairing operation. The parameter \( p \) is approximately 160 bits and the elliptic curve is defined as \( y = x^3 + 1 \) over \( \mathbb{F}_q \), where \( q \) is 512 bits. The rough operation time of each entity in every phase of PPMC is shown in Table I.

B. Communication Overhead

We demonstrate the communication burden of PPMC. The public parameters are the same as those in the experiment, that is, \( |p|=160 \) bits and \( |q|=512 \) bits. In the registration phase, a registrant (i.e., either customer or mobile user) sends a registering request \((I, C, A, PK_a)\) to the authority. This registering request is of \(|I| + 1344\) bits, where \(|I|\) denotes the binary length of the identity. The authority returns \((A, s'', e, RK)\) to the registrant, whose binary length is 1344 bits. In task allocation, the customer uploads \((c_1, c_2, c_3, expires, \tilde{N}_{n,n}, PK_a)\) and the mobile user sends \((\mu, \tilde{N}_{n,n}, PK_a)\) to the service provider, which are of \(4512 + 160n^2 + |\text{expires}| \) bits and

<table>
<thead>
<tr>
<th>Phase</th>
<th>Authority</th>
<th>User</th>
<th>Customer</th>
<th>Provider</th>
<th>User</th>
<th>Customer</th>
<th>Provider</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Multiplication</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Point Addition</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Exponentiation ( e )</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Exponentiation ( n )</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>14</td>
<td>8</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Running Time (ms)</td>
<td>45.524</td>
<td>151.304</td>
<td>98.701</td>
<td>138.473</td>
<td>147.685</td>
<td>56.405</td>
<td>1.283</td>
<td>3.661</td>
</tr>
</tbody>
</table>

Table I: COMPUTATIONAL OVERHEAD OF PPMC
2976 + 160n^2$ bits, respectively. Here, $\lceil\text{expires}\rceil$ denotes the binary length of $\text{expires}$. The service provider responds $(\text{num}, c_{2}, c_{3}, c_{4}, \text{expires})$, which is $2560 + |\text{num}| + |\text{expires}|$ bits, to a matched mobile user or false, 1 bit, to an unmatched one, where $|\text{num}|$ denotes the binary length of $\text{num}$. After the mobile user obtains the sensing data, it sends the sensing report $(\text{num}, D_{i}, D_{j}', X_{i}, Y_{i}, Z_{i}, \tau_{j})$ to the service provider. This sensing report is of $3768 + |\text{num}| + |\tau_{j}|$ bits, where $|\tau_{j}|$ denotes the binary length of $\tau_{j}$. The service provider sends a $1024$-bit $W$ to the authority in case that a mobile user double-submits data, and forwards all the sensing reports to the customer.

VI. RELATED WORK

Mobile crowdsensing has attracted great interest from the research community in recent years, especially from the security and privacy perspectives. Cornelius et al. [5] proposed Anon-Sense, a basic architecture for privacy-aware task releasing and data sensing. The participating mobile devices collaboratively sense the data and submit them through Mix networks. Gibert et al. [7] presented a trustworthy mobile sensing platform to preserve data integrity and mobile users’ privacy. Dimitriou et al. [16] raised the problem of customer’s privacy leakage and proposed a privacy-preserving access control mechanism named PEPPeR in sensing applications, which preserves the privacy of customers. Kazemi and Shahabi [17] proposed a trustworthy and privacy-preserving framework to preserve mobile user’s privacy. However, none of above frameworks enables to preserve the privacy for both mobile users and customers. To this end, Cristofaro and Soriente [18] proposed a privacy-enhanced participatory sensing infrastructure (PEPSI), which utilizes a blind extraction technique in identity-based encryption to realize the anonymity for both mobile users and customers. The PEPSI also utilizes a blind matching method to find the sensing reports for a specific task. Qiu et al. [8] developed a $k$-anonymous privacy-preserving scheme for mobile sensing to preserve the privacy of mobile users and improve the data quality by integrating the data code technique and message transfer strategies. Jin et al. [19] proposed a new mobile crowdsensing framework by combining incentive, data aggregation and data perturbation mechanisms. By selecting mobile users with reliable data and compensating their costs on sensing data, this framework can guarantee the privacy of mobile users and the accuracy of crowdsensing results. Gong et al. [20] introduced a flexible optimization framework to balance the trade-offs among utility, privacy and efficiency, and proposed a task recommendation framework, which recommends mobile crowdsourcing tasks without violating the privacy of mobile users. Different from the aforementioned works, in this paper, we aim to improve the privacy preservation for customers and mobile users, and propose a strong privacy-preserving mobile crowdsensing framework that can achieve identity privacy, data privacy and location privacy for both customer and mobile users. Meanwhile, a matrix-based location matching mechanism is developed to support accurate task allocation without violating points of interest of customers and location of mobile users.

VII. CONCLUSIONS

In this paper, we have proposed a mobile crowdsensing framework supporting location-based task allocation and preserving the identities of both customer and mobile users, simultaneously. Specifically, we not only achieve identity privacy, location privacy and data privacy for both customers and mobile users, but also allow the service provider to select mobile users for performing mobile crowdsensing based on the points of interest of customers and the location of mobile users. In addition, we have discussed the security feature of conditional privacy preservation and analyzed the computational and communication overhead to show the efficiency of PPCM. In our future work, we will extend our work to design a secure context-aware task allocation framework for mobile crowdsensing.

REFERENCES