Enabling Strong Privacy Preservation and Accurate Task Allocation for Mobile Crowdsensing

Jianbing Ni, *Member, IEEE,* Kuan Zhang, *Member, IEEE,* Qi Xia, Xiaodong Lin, *Fellow, IEEE,* Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Mobile crowdsensing engages a crowd of individuals to use their mobile devices to cooperatively collect data about social events and phenomena for customers with common interest. It can reduce the cost on sensor deployment and improve data quality with human intelligence. To enhance data trustworthiness, it is critical for service provider to recruit mobile users based on their personal features, e.g., mobility pattern and reputation, but it leads to the privacy leakage of mobile users. Therefore, how to resolve the contradiction between user privacy and task allocation is challenging in mobile crowdsensing. In this paper, we propose SPOON, a strong privacy-preserving mobile crowdsensing scheme supporting accurate task allocation based on geographic information and credit points of mobile users. In SPOON, the service provider enables to recruit mobile users based on their locations, and select proper sensing reports according to their trust levels without invading user privacy. By utilizing proxy re-encryption and BBS+ signature, sensing tasks are protected and reports are anonymized to prevent privacy leakage. In addition, a privacy-preserving credit management mechanism is introduced to achieve decentralized trust management and secure credit proof for mobile users. Finally, we show the security properties of SPOON and demonstrate its efficiency in terms of computation and communication.

Index Terms—Mobile crowdsensing, task allocation, trust management, privacy preservation.

1 Introduction

The integration of sensors and embedded computing devices triggers the emergence of mobile crowdsensing [2], in which user-centric mobile devices, e.g., smartphones, in-vehicle devices and wearable devices, are utilized to sense, collect and process data about social events and phenomena. This "sensing as a service" [3] elaborates our knowledge of the physical world by opening up a new door for data collection and sharing [4]. Due to the increasing popularity of mobile devices, mobile crowdsensing supports a broad range of sensing applications nowadays, ranging from social recommendation, such as restaurant recommendation, parking space discovery and indoor floor plan reconstrction [5], to environment monitoring, such as air quality measurement, noise level detection and dam water release warning. With human intelligence and user mobility, mobile crowdsensing can significantly improve the trustworthiness of sensing data, extend the scale of sensing applications and reduce the cost on high-quality data collection [6].

While mobile crowdsensing makes data sensing appealing than ever, it also brings new challenges towards mobile users, one of which is privacy leakage, indicating that mobile crowdsensing puts the privacy of mobile users at stake [7], [8], [9]. The sensing data collected from the surrounding areas are necessarily people-centric and related to some aspects of mobile users and their social settings: where they are and where they are going; what places they are frequently visited and what they are seeing; how their health status is and which activity they prefer to do. Social event photos may expose the social relations, locations or even political affiliations of mobile users. The spatial data collected by the carried devices might disclose mobile users' trajectories. For example, Google Maps collect the "anonymous" location information of drivers for real-time traffic map generation, but still expose the driving routes and trajectories of drivers. Further, the more sensing tasks mobile users engaged in and the richer data the users contribute to, the higher probability that their sensitive information may be exposed with. Therefore, preserving the privacy of mobile users is the first-order security concern in mobile crowdsensing. If no effective privacy-preserving mechanism is on-shelf, it is of difficulty to motivate mobile users to join in mobile crowdsensing services. In addition, the sensing tasks may contain sensitive information about the customers who issue them, such as identities, locations, references and purchase intentions [10]. For example, a house agency may know Bob desire to buy a house in a particular area if Bob releases tasks to collect traffic condition and noise level in the neighborhood. To preserve the privacy

- *Part of this research work was presented in IEEE International Conference on Communications (ICC 2017) [1].*

- *J. Ni and X. Shen are with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. Email: {j25ni, sshen}@uwaterloo.ca.*
- *K. Zhang is with Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE 68182 USA. Email: kuan.zhang@unl.edu.*
- *Q. Xia is with Center of Cyber Security, University of Electronic Science and Technology of China, Chengdu, 611731, China, Email: xiaqi@uestc.edu.cn.*
- *X. Lin is with School of Computer Science, University of Guelph, Guelph, Ontario, Canada N1G 2W1, email: xlin08@uoguelph.ca.*

of customers or mobile users, several privacy-preserving mobile crowdsensing schemes [11], [12], [13], [14] have been proposed by utilizing anonymity techniques. Nevertheless, anonymization is insufficient for privacy preservation, since the mobile users may be traced by travel routes and social relations. It is possible to uniquely identify 35% of mobile users using their top-two locations and 85% of them from their top-three locations based on a large set of call data records provided by a US nationwide cell operator [15]. Therefore, it is of importance to explore strong privacy-preserving mechanisms to prevent identity, location and data privacy leakage at the same time for both customers and mobile users in mobile crowdsensing.

Once all profiles of mobile users and customers are perfectly preserved, it is impossible for service providers to accurately recruit mobile users for task fulfillment, while task allocation is a critical component in mobile crowdsensing to ensure the quality of sensing results. Different from traditional sensing networks, the produced data cannot be predicted as a priori, and their trustworthiness totally depends on the intelligence and behaviors of mobile users. In general, the higher quality the sensing data have, the more efforts and costs the mobile users should pay. Therefore, the set of mobile users would directly affect the quality of sensing data. How to identify the right group of mobile users to produce the desired data according to the targets of sensing tasks is a complex problem from the service provider's perspective. Geography-based and reputation-based approaches are popular in mobile crowdsensing to allocate tasks to mobile users, but either has its inherent weaknesses. Firstly, reputation-based task allocation mechanisms [12], [16], [17], [18] need a trusted third party (TTP) to perform heavy reputation management task and they are vulnerable to reputation-linking attacks, in which the anonymous mobile users can be re-identified based on their reputations. Secondly, geography-based task allocation schemes can optimize user selection based on their spatial and temporal correlation [19], but they disclose the contents of sensing tasks and the locations of mobile users to the service provider, while location privacy is one of the primary concerns for mobile users in pervasive environments. In summary, privacy preservation and task allocation become a pair of contradictory objectives in mobile crowdsensing.

To resolve this issue, we propose a Strong Privacy-preserving mObile crOwdseNsing scheme (SPOON) supporting location-based task allocation, decentralized trust management and privacy preservation for both mobile users and customers simultaneously. By leveraging blind signatures and randomized matrix multiplication, we fully prevent the privacy leakage from the profiles of both mobile users and customers, including locations, identities and credit points, without scarifying the normal mobile crowdsensing functions of service providers, such as task allocation, data filtering and trust management. The main contributions of this paper are summarized as three folds:

- We design a privacy-preserving location matching mechanism based on matrix multiplication to allow service providers to allocate sensing tasks based on the sensing areas of tasks and the geographic locations of mobile users. Specifically, the service provider can determine whether a mobile user is in the sensing area of a task based on two randomized matrices generated from the sensing area and the user's location, respectively. Thus, the service provider can learn the result of location matching, but acquire nothing about the interested areas of customers and the locations of mobile users.
- By extending the proxy re-encryption and the BBS+ signature, we protect the sensitive information about mobile users and customers, including their identities, credit points, sensing tasks and sensing reports. Specifically, we allow the registered customers and mobile users to anonymously prove their capacities and trust levels to participate in the crowdsensing services and securely perform the sensing tasks without exposing contents of sensing tasks or sensing reports. Besides, to prevent the mobile users from misbehaving for unfair rewards, a trusted authority enables to detect the greedy mobile users and recover their identities.
- We introduce a privacy-preserving credit management mechanism, in which mobile users are able to prove their trustworthiness without the exposure of credit points and the management of centralized servers. In particular, it supports the positive and negative updates of credit points for mobile users based on the contributions on the tasks. In addition, multiple service providers can cooperatively maintain a unique trust evaluation system, in the way that mobile users are allowed to participate in the mobile crowdsensing services offered by different service providers using unique credit points.

The remainder of this paper is organized as follows. We review the related work in section 2, and formalize system model, threat model and identify security goals in section 3. In section 4, we propose our SPOON, followed by the security discussion in section 5. In section 6, we discuss some extensions on SPOON and evaluate performance in section 7. Finally, we draw the conclusion in section 8.

2 Related Work

To achieve privacy-preserving mobile crowdsensing, the state-of-the-art solutions aim to protect the locations [20], [21], [22], [23], identities [13], [14], [24], [25] or sensing data [26], [27], [28] of mobile users. Christin et al. [20] investigated the location privacy of mobile users and presented a decentralized and collaborative mechanism to allow mobile users to exchange the sensing data for the protection of their travel routes. Wang et al. [21] designed a location aggregation method to cluster users into groups for k-anonymity. To et al. [22] and Ma et al. [23] protected mobile users' locations by adding noise to break the correlation of real locations and obfuscated locations. To preserve identity privacy, AnonySense [11] was proposed to allow mobile devices to deliver sensing data through Mix networks. Dimitriou et al. [13] raised the problem of customer's privacy leakage and designed a privacy-preserving access control scheme for mobile sensing to preserve the privacy of customers. However, none of above schemes enables to

TABLE 1
Features Comparison on SPOON and other works

Features	Identity Privacy		Data Privacy		Location Privacy		Credit Management		
	Users	Customers	Users	Customers	Users	Customers	Credit Privacy	No TTP	Greedy User Tracing
SPOON	√	√	√	√	√	√	√	√	√
[11], [12]	√	X	X	X	X	X	X	X	X
[13]	X	√	X	√	X	√	X	X	X
[14], [18], [29]	√	X	√	X	√	X	X	X	X
[35]	√	X	√	X	√	X	X	√	√
[20], [30]	X	X	X	X	√	X	X	X	X
[24], [25]	√	√	√	X	X	X	X	X	X
[26], [27], [36], [38]	X	X	√	X	X	X	X	X	X

preserve the privacy for both mobile users and customers simultaneously. Therefore, Cristofaro and Soriente [24] proposed a privacy-enhanced participatory sensing infrastructure (PEPSI) based on the blind extraction technique and identity-based encryption. Unfortunately, Günther et al. [25] demonstrated PEPSI is vulnerable to collusion attacks across mobile users and customers. To fix this drawback, a new infrastructure is designed from anonymous identity-based encryption. Qiu et al. [14] presented SLICER, a k-anonymous privacy-preserving scheme for mobile sensing that achieves strong privacy preservation for mobile users and high data quality. However, these schemes may be insufficient to preserve the privacy nowadays, since it is possible to re-identify the mobile users or customers through the combination of information from different sources, such as travel routes, social relations or payment records. To protect the sensing data, Zhou et al. [26] extended a generalized efficient batch cryptosystem to support fine-grained multi-receiver multi-file sharing in cloud-assisted mobile crowdsensing. Chen et al. [27] introduced a group management protocol to guarantee differential privacy of personal data. Rahaman et al. [28] designed a bilinear-map based group signature scheme, while supporting sub-linear revocation check with backward unlinkability and exculpability, for anonymous-yet-accountable crowdsensing. Although the privacy-preserving schemes that can protect identity, location or data privacy are on-shelf, designing a scheme to achieve strong privacy preservation is non-trivial.

However, after the privacy of mobile users and customers is preserved, it is difficult for the service provider to find proper mobile users for task fulfillment. Many privacy-preserving task allocation schemes have been proposed in mobile crowdsensing, which can divided into three categories, namely, location-based task allocation [21], [22], [29], [30], auction-based incentive [31], [32], [33] and reputation-based task allocation [12], [34], [35], [36]. Kazemi and Shahabi [29] focused on spatial task assignment for spatial crowdsourcing, in which the service provider allocates tasks based on the locations of mobile users. To et al. [22] introduced a framework to determine effective geocast regions for reaching high task assignment ratio, while protecting the locations of mobile users. Wang et al. [30] presented a personalized privacy-preserving task allocation scheme for mobile crowdsensing that can effectively allocate tasks while providing personalized location privacy protection. Incentive mechanisms also have been proposed to encourage mobile users participating crowdsensing tasks, in which auction is one of the commonly adopted incentive mechanisms [31], [32], [33]. Gao et al. [32] introduced a reverse-auction-based incentive mechanism to select winning bids with a nearly minimum social cost, considering the quality of sensing data. Zhang et al. [33] designed an auction-based incentive mechanism to offer rewards to users for both participation and solicitation, and eliminate malicious price manipulations against dishonest mobile users. With the consideration of bid privacy, Lin et al. [37] designed two privacy-preserving auction-based incentive mechanisms based on differential privacy. Jin et al. [38] integrated user incentive, data aggregation and data perturbation mechanisms to design an incentivizing privacy-preserving data aggregation scheme to generate high-accurate aggregated results in mobile crowdsensing. In addition, it is common to realize task allocation based on the reputation of mobile users. Kazemi et al. [34] defined reputation scores to represent the probability that a mobile user can perform a task correctly, and a confidence level to state that a task is acceptable. Huang et al. [12] demonstrated that mobile users are vulnerable to linking attacks of reputations and presented an anonymization scheme and a reputation management mechanism to minimize the risk of such attack. Wang et al. [35] proposed ARTSense to achieve the trust management without identity exposure in mobile sensing. ARTSense achieves both positive and negative updates of reputations for mobile users with no TTP, but it still requires a reputation database for each service provider to support reputation management. Ma et al. [36] designed two privacy-preserving reputation management schemes for edge computing enhanced mobile crowdsensing. The reputation is used to identify the malicious mobile users who are willing to participate in the tasks, but the precise reputations are directly revealed to the service provider and other curious entities.

For the above reasons, in our preliminary work [1], we proposed a privacy-preserving mobile crowdsensing framework enabling trajectory-based task allocation and privacy preservation for both mobile users and customers. In this paper, we extend this work to support privacy-preserving reputation-based task allocation in mobile crowdsensing. In specific, the proposed SPOON (1) provides strong privacy preservation for mobile users; (2) protects the identities, sensing areas and tasks for customers; (3) allows the service provider to allocate sensing tasks based on the locations and credit points of mobile users; and (4) supports privacy-preserving credit management without the centralized server. We show the comparison on the desirable features between SPOON and the existing works in Table 1.

3 Problem Statement

In this section, we formally define the system model and threat model, and identify our design goals.

3.1 System model

The mobile crowdsensing service provides customers a people-centric way for data collection from surrounding environment. The architecture consists of three entities: service providers, customers and mobile users.

Service Providers: The service providers develop cloud services by themselves or rent the cloud resources offered by cloud vendors. They have sufficient storage and computing resources to provide mobile crowdsensing services. The service providers receive sensing tasks from customers and allocate them to mobile users based on their locations. They also collect sensing reports from mobile users, select sensing reports based on the credit points of mobile users and generate sensing results for customers. Finally, the service providers distribute credit points to mobile users for incentive.

Customers: The customers can be individuals, corporations or organizations. They need to accomplish data collection tasks, e.g., to study traffic congestion in a city, pollution level of a creek and satisfactory on public transportation, but they do not have sufficient capabilities to perform tasks by themselves. Thereby, they issue their sensing tasks to the service providers.

Mobile Users: Every mobile user has several mobile devices, e.g., mobile phones, tablets, vehicles and smart glasses. These mobile devices, deployed with rich computing, communication and storage resources, are carried by their owners wherever they go and whatever they do. The mobile users make sure their devices have sufficient power to support the normal functions. They participate in sensing tasks and utilize their portable devices to collect data from their surrounding areas, and report sensing data to the service providers for earning credit points.

As shown in Fig. 1, the system model of mobile crowdsensing has the following steps.

Step 1: Every mobile user or customer needs to register at the trusted authority (TA) to access mobile crowdsensing services.

Step 2: A customer creates a sensing task for data collection in a sensing area and sends it to the service provider, along with the authentication message and the reward policy.

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

Step 4: The registered mobile users, who are willing to conduct data collection, obtain their current or future locations using mobile devices capable of localization (e.g., by wireless access points or GPS), and send their locations to the service provider, along with the authentication messages.

Step 5: Upon receiving the messages from mobile users, the service provider checks whether the mobile users are in the sensing area or they will be in the sensing area in the near future.

Step 6: The service provider recruits the mobile users whose locations can match the sensing area, and sends the sensing task to the matched mobile users.

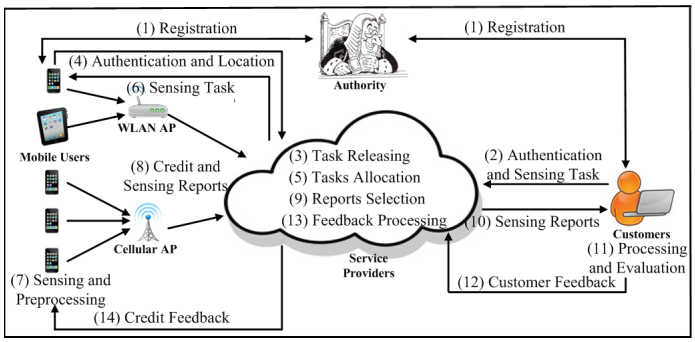


Fig. 1. System Model of Mobile Crowdsensing.

Step 7: The mobile users accept or reject the sensing task based on reward policy and costs of performing the task, and the mobile users, who accept the task, sense the event or phenomena, collect data, and generate sensing reports.

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

Step 9: The service provider selects the sensing reports generated by the mobile users with high trust levels.

Step 10: The service provider forwards the selected sensing reports to the customer.

Step 11: The customer reads the sensing reports and evaluates the quality and trust level of sensing reports to generate the feedback.

Step 12: The customer sends the feedback to the service provider.

Step 13: The service provider calculates the rewards (i.e., credit points) of the mobile users based on the feedback of the customer.

Step 14: The service provider distributes the credit points to the mobile users who make contributions on sensing area.

3.2 Threat Model

The service provider is responsible for offering the mobile crowdsensing service to customers, but it may strive to increase income and violate its privacy policy. For example, Uber, a crowdsourcing-based ride-sharing service provider, made ride-booking data publicly accessible without the permission of customers in January, 2017, for its own purpose. Therefore, the service provider is not fully trusted, but honest-but-curious. On one hand, the service provider would honestly perform the mobile crowdsensing service; on the other hand, it may learn a spatio-temporal probability distribution for a specific mobile user and other sensitive information about customers and mobile users, e.g., preference, social relation, political affiliation and purchase intention. The service provider would not collude with a mobile user to compromise the privacy of other mobile users, since the service provider and the mobile user have separate purposes in mobile crowdsensing, and the reputation of the service provider would be seriously damaged once the collusion is public.

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 participating in the

same tasks, and learn the background of customers they are working for to reach the expectations of customers. Also, mobile users may be greedy, such that they may anonymously submit more sensing reports than allowed to earn more credit points. In addition, the mobile users may maliciously forge, modify the sensing data or deliver ambiguous, biased sensing data to cheat customers. These forged or biased data can be discovered using redundancy or truth discovery approaches. The locations are extracted from GPS trusted chips in mobile devices or access points, we assume that mobile users cannot modify their location information.

The external attackers, such as eavesdroppers and hackers, also bring serious security threats towards mobile crowdsensing services. It is possible for an attacker to obtain the identities of the nearby mobile users or customers via physical observation, such that the anonymization may be insufficient to guarantee privacy preservation for customers and mobile users. The customers are fully trusted since they are the main beneficiaries of mobile crowdsensing services.

3.3 Design Goals

To enable strong privacy-preserving mobile crowdsensing under the aforementioned system model and against security threats, SPOON should achieve the following design goals:

- Strong Privacy Preservation: Strong privacy preservation refers that no sensitive information of mobile users would be exposed to the public, including location, identity, data or credit points. Compared with the traditional privacy preservation that focuses on obfuscating the location or the identity of mobile user, strong privacy preservation needs the protection of all the sensitive information, from the location and identity to the sensing data and credit points in mobile crowdsensing.
    1) Location Privacy Preservation: The locations of mobile users and the sensing areas of sensing tasks would not be exposed to others. The mobile users are only aware of whether they are in the sensing area or not.
    2) Data Confidentiality: No entity, except the delegated participants, can obtain the contents of releasing tasks or sensing reports, such that the privacy of customers and mobile users would not be disclosed to others.
    3) Anonymity of Mobile Users and Customers: The customer, mobile user, the service provider or their collusion is unable to link a sensing report to a mobile user or link a sensing task to a customer. It is even impossible for an attacker to identify whether two sensing reports are generated by the same mobile user, or two sensing tasks are issued by the same customer.
    4) Greedy User Tracing: The identity of the greedy mobile user, who submits more than one sensing reports for the same task in a reporting period, is identified to prevent the mobile user from being awarded unfair credit points.
    5) Privacy-preserving Credit Management: Credit points are used to record the trust of mobile users and encourage them to participate in the crowdsensing activities as rewards. The precise credit points of mobile users are hidden against the curious entities, including the service providers and the mobile users.
- Accurate Task Allocation: Accurate task allocation refers that the recruited mobile users are capable of fulfilling the sensing tasks. The capabilities include the fact that the mobile users are in sensing area or will visit the sensing area in the near future, and they are trusted to report available sensing data with high quality.
    1) Location-based Task Allocation: The sensing tasks are allocated to the mobile users in the sensing areas defined by the customers, and other mobile users out of the given areas cannot learn any information about the tasks.
    2) Trust-based Report Selection: The service provider selects the sensing reports based on the credit points of mobile users and awards credit points to mobile users if they report trustworthy sensing data. The balance of credit points is achieved, which means that it is impossible for the mobile users to forge credit points without being detected, such that the total credit points that a mobile user has should be equal to the awarded credit points plus the initial points.

4 SPOON

In this section, we review the preliminaries and propose our SPOON, which is composed of five phases, Service Setup, User Registration, Task Allocation, Data Reporting and Credit Assignment, based on the matrix multiplication, the BBS+ signature [41] and the proxy re-encryption [42].

4.1 Preliminaries

We review the preliminaries that are used to design our SPOON, including the bilinear map, the BBS+ signature and the proxy re-encryption.

Bilinear Map. Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be three cyclic groups with a prime order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is the bilinear pairing with the following properties:

- Bilinearity. For $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$, $a, b \in \mathbb{Z}_p$, $\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$.
- Non-degeneracy. For $g \neq 1_{\mathbb{G}_1}$, $h \neq 1_{\mathbb{G}_2}$ $\hat{e}(g, h) \neq 1_{\mathbb{G}_T}$.
- Computability. $\hat{e}$ is efficiently computable.
- (Unique Representation). The binary presentation for all elements in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ is unique.

If $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction, $\hat{e}$ is a type 3 bilinear pairing [43]. If $\mathbb{G}_1 = \mathbb{G}_2$, $\hat{e}$ is a type 1 bilinear pairing.

BBS+ Signature [41]. We briefly review the BBS+ signature with type 1 bilinear pairing due to [41], which can be utilized to sign ℓ-message vector $(m_1, \cdots, m_\ell)$.

Let $g, g_1, \cdots, g_{\ell+1}$ be generators of $\mathbb{G}_1$. Randomly choose x from $\mathbb{Z}_p$ as the secret key of the signature scheme, and compute the corresponding public key as $y = g^x$.

A signature on messages $(m_1, \cdots, m_\ell)$ is (A, e, s), where $A = (g g_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s})^{\frac{1}{x+e}}$ and (e, s) are random values chosen from $\mathbb{Z}_p$.

This signature can be checked as: $\hat{e}(g g_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s}, g) \stackrel{?}{=} \hat{e}(A, y g^e)$.

The security of BBS+ signature can be reduced to the q-SDH assumption and it can be utilized to construct a zero-knowledge proof-of-knowledge protocol that allows the signer to prove the possession of the message-signature pair.

Proxy Re-encryption [42]. Proxy re-encryption is a special public key encryption with a desirable property that a semi-trusted proxy enables to convert a ciphertext for Alice into a ciphertext for Bob without seeing the underlying plaintext, given a proxy re-encryption key. Thanks to this promising property, it has been widely employed in data sharing scenarios. The proxy re-encryption scheme was proposed by Ateniese et al. [42], the detailed scheme with the type 1 bilinear pairing is as follows:

- KeyGen$(\cdot)$. Alice picks a random value $a \in \mathbb{Z}_p$ as the secret key sk_a and compute the public key $pk_a = g^a$.
- RKeyGen(sk_a, pk_b). Alice delegates to Bob by sending the re-encryption key $rk_{A \to B} = g^{b/a}$ to a proxy by using Bob's public key.
- Encrypt(m, pk_a). To encrypt a message $m \in \mathbb{G}_T$ under pk_a, Alice chooses a random value $k \in \mathbb{Z}_p$ to compute $c_a = (g^{ak}, m\hat{e}(g, g)^k)$.
- Re-Enc$(c_a, rk_{A \to B})$. The proxy can change the ciphertext c_a into a ciphertext c_b for Bob with $rk_{A \to B}$. From c_a, the proxy calculates $\hat{e}(g^{ak}, g^{b/a}) = \hat{e}(g, g)^{bk}$ and releases $c_b = (\hat{e}(g, g)^{bk}, m\hat{e}(g, g)^k)$.
- Decrypt (c_b, sk_b). Bob enables to decrypt c_b to obtain m as $m = m\hat{e}(g, g)^k / (\hat{e}(g, g)^{bk})^{1/b}$.

4.2 Main Ideas of SPOON

Intuitively, the service provider should have the detailed profiles of mobile users, such as location, identity and trust level, to provide accurate task allocation, which results in the privacy leakage of mobile users. In fact, the service provider only needs to have the knowledge whether a mobile user is capable of performing the targeted sensing task, instead of possessing all the profiles. According to this observation, the privacy-preserving location matching scheme is designed from randomized matrix multiplication. Specifically, the location of each mobile user is represented as a matrix $\widetilde{L}_{m \times n}$, which is randomized by a matrix $\widetilde{M}_{m \times n}$. The sensing area of the task is denoted as another matrix $\widehat{L}_{m \times n}$ and randomized by a matrix $\widehat{M}_{m \times n}$. Matrix randomization protects the location of the mobile user and the sensing area of the task, but it is difficult for the service provider to identify the mobile users whose locations can match the sensing area of the task, after the locations are randomized. Therefore, we construct two invertible random matrices $\widetilde{M}_{m \times n}$ and $\widehat{M}_{m \times n}$ to achieve privacy-preserving location matching based on the multiplication of two randomized matrices.

Furthermore, the customers encrypt the sensing tasks for preventing the curious service provider and the unmatched mobile users from accessing them. Nevertheless, when the customer encrypts the sensing task, she is not aware of the mobile users who can conduct the sensing work, such that she has no idea about which public key should be used. Although proxy re-encryption [42] can achieve the sharing of encrypted sensing task in the aid of a proxy, such as the service provider, it will expose the sensing task to the proxy. To protect the sensing task, we divide the proxy into two entities, i.e., TA and the service provider, and design a two-level proxy re-encryption scheme for the sensing task sharing among matched mobile users. Specifically, the sensing task is encrypted by the public keys of both TA and the service provider, and the mobile users can decrypt the sensing task on the condition of owning the proxy key of TA and the re-encrypted ciphertext of the service provider. In this way, the service provider cannot acquire the task due to the lack of the secret key of TA, and the unmatched mobile users learn nothing since they cannot decrypt any ciphertext. Moreover, to enable TA to be offline, the proxy key of TA can be delegated to each mobile user during the user registration.

Finally, the blind signature is one of the widely used approaches to protect the identities of the mobile users. By using the BBS+ blind signature [41], the mobile user generates the zero-knowledge proof of the BBS+ signature to convince the service provider that she is the registered user capable of joining in the crowdsensing activity without exposing her real identity, and the tracing tag is generated from the public key to identify the greedy mobile user who double-reports the sensing data. However, if the mobile users are anonymous, it is challenging to support trust management of mobile users. Therefore, we extend the BBS+ blind signature by integrating the credit point into the signature of the user's identity and enabling the addition operation of credit points. Thereby, the positive and negative update of credit points can be supported. Specifically, the credit assignment is enabled for the service provider to reward credit points to the mobile users if their submitted reports are trustworthy, or penalize credit points, otherwise. The service provider can generate the new BBS+ signature on the rewarded or punished credit point for credit finalization. The BBS+ signature can be proved to other service providers when the mobile user participates in new sensing tasks, along with the updated credit points. Different from the existing trust management mechanisms in which a centralized server is needed for credit point management, each mobile user can manage her own credit points and show her trust level to the service provider when necessary. Thus, unique trust management is achieved, and multiple service providers can share the unique trust management system, as long as they trust each other. In addition, the zero-knowledge range proof of the credit points is designed, along with the zero-knowledge proof of the BBS+ signature. Based on the range proof, the mobile user can convince the service provider that she has higher credit points than the chosen threshold. By doing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2908638, IEEE Transactions on Mobile Computing

so, the service provider can select the sensing reports from the mobile users who have high credit points based on the exposed thresholds, but it has no knowledge about the precise credit points of mobile users.

4.3 High-Level Description

We first provide a high-level description of SPOON. The notations frequently used in SPOON are listed in Table 2.

Service Setup: TA bootstraps the whole mobile crowdsensing service for the service provider by defining the public parameters $(\mathbb{G}, \mathbb{G}_T, p, g, g_0, g_1, g_2, g_3, h, h_0, h_1, h_2, h_3, h_4, G, H, \mathcal{G}, \mathcal{H}, \mathcal{F})$ and generates its secret-public key pair (α, T, T_0). The service provider also generates the secret-public key pair (β, S), and defines a matrix $L_{m\times n}$ to denote the geographic region of its crowdsensing service.

User Registration: TA registers the mobile users and customers, who are willing to participate in the mobile crowdsensing service. It evaluates the registrant to determine the initial credit point P_0 and interacts with the registrant to generate an anonymous credential (A, e, s, B, f, t). (A, e, s) is used to access the mobile crowdsensing service and (B, f, t) is used to credit management for the registrant. To achieve the anonymity, the ownership of (A, e, s) and (B, f, t) is proved by the registrant for identity authentication and credit evaluation using zero-knowledge proofs, respectively. Besides, RK is assigned to the registrant for the decryption of allocated sensing tasks.

Task Allocation: A customer generates a sensing task ST and sends the message $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w, \mathcal{PK}_2)$ to the service provider, which consists of the encrypted task (c_1, c_2, c_3), the expiration time $expires$, the randomized sensing area $\widehat{N}_{n\times n}$, the identity proof $\mathcal{PK}_2$ and other information. The latter releases $(num, expires, \gamma)$ to attract mobile users for participation, where num is the identifier of ST. A mobile user $U_{i\{i\in R\}}$ sends its location $\widetilde{N}_{n\times n}$ and identity proof $\mathcal{PK}_3$ to the service provider. Then, the service provider finds the set of mobile users $U_{i\{i\in\mathcal{L}\}}$ in the sensing area of ST based on two matrices $(\widehat{N}_{n\times n}, \widetilde{N}_{n\times n})$. Since $(\widehat{N}_{n\times n}, \widetilde{N}_{n\times n})$ are randomized matrices, the service provider can learn whether U_i is in the sensing area of ST based on matrix multiplication, but has no information about ST's sensing area and U_i's location. The service provider re-encrypts the ciphertext (c_1, c_2, c_3) to be decryptable for $U_{i\{i\in\mathcal{L}\}}$ using β. Finally, the service provider sends $(num, c_2, c_3, c_4, expires, \gamma, w)$ to $U_{i\{i\in\mathcal{L}\}}$.

Data Reporting: $U_{i\{i\in\mathcal{L}\}}$ encrypts the collected data m_i to generate (D_i, D_i'), and sends the sensing report $(num, D_i, D_i', C_i', X_i, Y_i, Z_i, Q_i, \tau_j, \mathcal{SPK})$ to the service provider, in which C_i' is the commitment on the identity I_i and credit point P_i, X_i is the identifier of this report, Y_i is the identifier of U_i, Z_i is a tag to identify the double-reporting user, Q_i is the claimed credit threshold to show that the number of credit points U_i has is larger than Q_i, τ_j is the current slot for reporting, and $\mathcal{SPK}$ is used to prove the ownership of its credit points P_i. The service provider selects $w-$sensing reports based on the claimed thresholds and forwards the selected reports to the customer. TA can recover the identity of the anonymous mobile user who double-reports sensing reports with the service provider using the double-reporting tag Z_i.

TABLE 2
Frequently Used Notations

$U_{i\{i\in R\}}$	Set of registered mobile users
$U_{i\{i\in\mathcal{L}\}}$	Set of mobile users in sensing area L
ST	A task issued by a customer
$task$	The detailed content of a task ST
$expires$	The expiration time of a task ST
$area$	The sensing region of a task ST
$L_{m\times n}$	A matrix to represent the service area of the service provider
$\widehat{L}_{m\times n}$	A matrix to represent the sensing area of a task ST
$\widetilde{L}_{m\times n}$	A matrix to represent the current and future locations of a user
$\widehat{M}_{m\times n}$	A random invertible matrix
$\widetilde{M}_{m\times n}$	A random invertible matrix
I	The unique identity of a registrant (mobile user or customer)
P_0	The initial credit point of a mobile user
ϵ	The trust level of a sensing report
γ	The maximum of trust level in a task ST
Q	The credit threshold chosen by a mobile user
A, e, s	The anonymous credential of a mobile user or customer
B, f, t	The anonymous credential of a mobile user with credit point P

Credit Assignment: The customer evaluates the trustworthiness of each report and returns the corresponding trust level $\epsilon_i \in [-\gamma, \gamma]$ to the service provider. The latter computes the number of credit points awarded to U_i, θ_i, and forwards $(B_i, t_i'', f_i, \theta_i, Y_i)$ to U_i, where (B_i, t_i'', f_i) is the ticket for awarded credit points θ_i, and Y_i is used to identify the mobile user U_i. Once receiving $(B_i, t_i'', f_i, \theta_i, Y_i)$, U_i updates its credit points $P_i' = P_i + \theta_i$ and generate the credit credential (B_i, f_i, t_i) of the new P_i'.

4.4 The Detailed SPOON

We then show the detailed SPOON as follows.

4.4.1 Service Setup

Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ be three cyclic groups with a prime order p, where p is λ bits, and $\hat{e}$ be the type 3 bilinear map. The authority picks random generators $g, g_0, g_1, g_2, g_3 \in \mathbb{G}_1$, $h, h_0, h_1, h_2, h_3, h_4 \in \mathbb{G}_2$ and computes $G = \hat{e}(g, h)$ and $H = \hat{e}(g, h_0)$ respectively. TA also chooses a random value $\mathcal{G} \in \mathbb{G}_T$ and defines a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$ and a pseudo-random function $\mathcal{F} : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_p$. The public parameters $param$ are $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g, g_0, g_1, g_2, g_3, h, h_0, h_1, h_2, h_3, h_4, G, H, \mathcal{G}, \mathcal{H}, \mathcal{F})$. TA randomly chooses $\alpha \in \mathbb{Z}_p$ as its secret key and calculates the public keys $T = g^\alpha$ and $T_0 = h^\alpha$.

To setup the mobile crowdsensing service, the service provider randomly chooses its secret key $\beta \in \mathbb{Z}_p$ and computes $S = g^\beta$ as its public key. It also employs a matrix $L_{m\times n}$ to denote the geographical region that the crowdsensing service can cover according to the longitude and latitude. Each entry in the matrix denotes a small grid in the sensing region, as shown in Fig. 3. Assume the longitude of Ontario is from $74.40°$W to $95.15°$W, the latitude is from $41.66°$N to $57.00°$N, we can use a 208×154 matrix or 2075×1534 matrix more precisely to represent the Ontario region.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2908638, IEEE Transactions on Mobile Computing

4.4.2 User Registration

Either customer or mobile user is required to register at the TA to obtain an anonymous credential, which is used to participate in the crowdsensing service. Each registrant is assigned a unique identity I in the system, which can be the telephone number or mailing address in practise. The registrant picks three random values $s', a, t' \in \mathbb{Z}_p$ to compute $C = g_1^{s'} g_2^a$, $C' = h_1^{t'} h_2^a$, $\widehat{A} = h_0^a$, and sends $(I, C, C', \widehat{A})$ to TA, along with the following zero-knowledge proof:

$$\mathcal{PK}_1\{(s', t', a) : C = g_1^{s'} g_2^a \wedge C' = h_1^{t'} h_2^a \wedge \widehat{A} = h_0^a\}.$$

TA firstly checks the proof $\mathcal{PK}_1$ for ensuring that $(C, C', \widehat{A})$ are generated correctly. Then, it evaluates the registrant's initial credit point according to its credit record, which is assumed to be P_0. After that, TA randomly picks $s'', e, t'', f \in \mathbb{Z}_p$ to calculate $A = (g_0 C g_1^{s''} g_3^I)^{\frac{1}{\alpha+e}}$, $B = (h_0 C' h_1^{t''} h_3^I h_4^{P_0})^{\frac{1}{\alpha+f}}$, $RK = \widehat{A}^{\frac{1}{\alpha}}$, and returns $(A, B, s'', t'', e, f, P_0, RK)$ to the registrant through secure channel. Finally, TA stores the tuple $(I, P_0, \widehat{A})$ in its database.

The registrant computes $s = s' + s'', t = t' + t''$ and checks

$$\hat{e}(A, T_0 h^e) \stackrel{?}{=} \hat{e}(g_0 g_1^s g_2^a g_3^I, h),$$
$$\hat{e}(T g^f, B) \stackrel{?}{=} \hat{e}(g, h_0 h_1^t h_2^a h_3^I h_4^{P_0}).$$

The registrant stores $(A, e, s, B, f, t, a, I, P_0, \widehat{A}, RK)$ secretly on the read-only memory of mobile device.

4.4.3 Task Allocation

A customer with registered information $(A, e, s, B, f, t, a, I, P_0, \widehat{A}, RK)$ has a sensing task to be allocated to mobile users and requests the sensing data slot by slot, where each slot ranges from minutes to days depending on the specific requirements of the sensing task. The statement of the task is defined as $ST = (task, expires, area, \gamma, w)$, which indicate the content (what to sense), the expiration time (when to sense), the sensing area (where to sense), the maximum trust level and the number of required reports, respectively. Other attributes (e.g., sensing intervals, acceptance conditions, benefits, reporting periods) can be illustrated in $task$. To protect the content of the task, the customer randomly picks $k, r_1, r_2, r_3 \in \mathbb{Z}_p$ to calculate $u = g^k, c_1 = S^{r_1}$, $c_2 = T^{r_2}$ and $c_3 = (task||u)G^{r_1}H^{r_2}$. Then, the customer generates a matrix $\widehat{L}_{m\times n}$ to represent the target sensing region $area$. As depicted in Fig. 2, for each position in the sensing area, the corresponding entry in $\widehat{L}_{m\times n}$ is set to be a random value chosen from $\mathbb{Z}_p^*$, and the value for a location outside is set to be zero. To mask the sensing area in $\widehat{L}_{m\times n}$, the customer picks $m \times n$ random numbers from $\mathbb{Z}_p^*$ to generate an invertible matrix $\widehat{M}_{m\times n}$ and computes $\widehat{N}_{n\times n} = \widehat{L}_{m\times n}^T \cdot \widehat{M}_{m\times n}$, where $\widehat{L}_{m\times n}^T$ is the transpose of the matrix $\widehat{L}_{m\times n}$. Note that all non-zero entries in $\widehat{L}_{m\times n}$ should be distinct, unless an attacker still can learn the sensing region from $\widehat{N}_{n\times n}$. Finally, the customer keeps k in private and sends $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w)$ to the service provider, along with the following zero-knowledge proof:

$$\mathcal{PK}_2\{(A, e, s, a, I) : \hat{e}(A, T_0 h^e) \stackrel{?}{=} \hat{e}(g_0 g_1^s g_2^a g_3^I, h)\}.$$

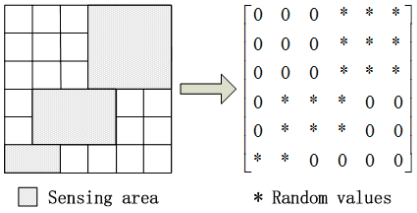


Fig. 2. Sensing Area and the Matrix $\widehat{L}_{6\times 6}$.

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

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

$$\mathcal{PK}_3\{(A_i, e_i, s_i, a_i, I_i) : \hat{e}(A_i, T_0 h^{e_i}) \stackrel{?}{=} \hat{e}(g_0 g_1^{s_i} g_2^{a_i} g_3^{I_i}, h)\}.$$

The service provider returns failure if $\mathcal{PK}_3$ is invalid. Otherwise, for each unexpired task, it uses $\widehat{N}_{n\times n}$ to calculate $N_{n\times n} = \widetilde{N}_{n\times n} \cdot \widehat{N}_{n\times n}$ and checks whether $N_{n\times n}$ is zero matrix or not. If $N_{n\times n}$ is non-zero matrix, which means that U_i can match ST, the service provider calculates $c_4 = \hat{e}(c_1, \mu)^{\frac{1}{\beta}}$ and releases $(num, c_2, c_3, c_4, expires, \gamma, w)$ for U_i. If there is no task to match U_i, the service provider responds failure.

When U_i obtains $(num, c_2, c_3, c_4, expires, \gamma, w)$, it decrypts (c_2, c_3, c_4) by using (ν, a_i) as $task||u = c_3 c_4^{-\frac{1}{\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}}$. Then, U_i evaluates the task and determines to participate in or abandon this task according to benefit and cost. If the task ST is accepted, U_i starts to perform the sensing work according to the details in $task$. The correctness of $task||u$ is elaborated as follows:

$$\begin{aligned} & c_3 c_4^{-\frac{1}{\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}} \\ = \; & c_3 \hat{e}(c_1, \mu)^{-\frac{1}{\beta\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}} \\ = \; & (task||u) G^{r_1} H^{r_2} \hat{e}(S^{r_1}, h^\nu)^{-\frac{1}{\beta\nu}} \hat{e}(T^{r_2}, h_0^{\frac{a_i}{\alpha}})^{-\frac{1}{a_i}} \\ = \; & (task||u) G^{r_1} H^{r_2} G^{-r_1} H^{-r_2} \\ = \; & task||u. \end{aligned} \tag{1}$$

4.4.4 Data Reporting

U_i collects and pre-processes the data $m_i \in \mathbb{G}_T$ and submits a sensing report to the customer periodically, which includes the collection time, the sensing location and the detailed content. The reporting periods are defined by the customer, and we assume the current slot is τ_j. To prevent attackers

from learning m_i, U_i uses u to encrypt m_i as $D_i = u^{\hat{r}_i}$, $D'_i = m_i G^{\hat{r}_i}$, where $\hat{r}_i$ is a value randomly chosen from $\mathbb{Z}_p$. Then, U_i randomly picks $t'_i \in \mathbb{Z}_p$ to compute $C'_i = h_1^{t'_i} h_2^{a_i} h_3^{I_i} h_4^{P_i}$. Next, U_i computes $X_i = \mathcal{H}(num||m_i||\tau_j)$, $v_i = \mathcal{F}_{a_i}(num||I||\tau_j)$, $Y_i = H^{v_i}$ and $Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}$. Finally, U_i chooses a credit threshold Q_i and sends the report $(num, D_i, D'_i, C'_i, X_i, Y_i, Z_i, Q_i, \tau_j)$ to the service provider, along with the following zero-knowledge proof:

$$\mathcal{SPK}\left\{\begin{array}{c}(B_i, f_i, t_i, t'_i, a_i, I_i, P_i, v_i): \\ \hat{e}(Tg^{f_i}, B_i) \stackrel{?}{=} \hat{e}(g, h_0 h_1^{t_i} h_2^{a_i} h_3^{I_i} h_4^{P_i}) \wedge \\ C'_i = h_1^{t'_i} h_2^{a_i} h_3^{I_i} h_4^{P_i} \wedge \\ P_i > Q_i \wedge \\ Y_i = H^{v_i} \wedge \\ Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}\end{array}\right\}(num).$$

The service provider returns failure if $\mathcal{SPK}$ is invalid; otherwise, the service provider checks whether there is another report $(num, \widetilde{D}_i, \widetilde{D}'_i, \widetilde{C}'_i, \widetilde{X}_i, Y_i, \widetilde{Z}_i, \widetilde{Q}_i)$ that has the same Y_i and different $\widetilde{X}_i$ with the new received report $(num, D_i, D'_i, C'_i, X_i, Y_i, Z_i, Q_i)$. If yes, the service provider computes and sends $W = (\frac{\widetilde{Z}_i^{X_i}}{Z_i^{\widetilde{X}_i}})^{\frac{1}{X_i - \widetilde{X}_i}}$ to TA, and TA can find the mobile user's identity I_i by utilizing $\widehat{A}_i$ in the database to check $W = \hat{e}(g, \widehat{A}_i)$. Such that, the identity of the greedy mobile user is recovered by TA, if it submits two different sensing reports in a reporting slot. Then, according to the claimed thresholds of mobile users, the service provider chooses w reports that have top-w thresholds, and releases them to the customer. Note that the mobile users, whose reports are not selected, can increase their thresholds in the next reporting slot $\tau_j + 1$.

When the customer retrieves the reports, it can decrypt them using the stored k as $m_i = D'_i \hat{e}(D_i, h)^{\frac{1}{k}}$ one by one.

4.4.5 Credit Assignment

After the customer obtains the sensing result, it evaluates the trustworthiness of each report and responds the corresponding trust level to the service provider. The trust level of m_i is defined as $\epsilon_i \in [-\gamma, \gamma]$. If ϵ_i is positive, m_i is trustworthy, otherwise, m_i is incredible.

Upon receiving trust levels, the service provider randomly picks $t''_i, f_i \in \mathbb{Z}_p$ to compute $\theta_i = INT(\epsilon_i Q_i)$, $B_i = (h_0 h_1^{t''_i} C'_i h_4^{\theta_i})^{\frac{1}{\beta + f_i}}$, and releases $(B_i, t''_i, f_i, \theta_i, Y_i)$ for U_i, where $INT(x)$ is the nearest integer function.

U_i retrieves $(B_i, t''_i, f_i, \theta_i, Y_i)$ from the service provider, computes $t_i = t'_i + t''_i$, $P'_i = P_i + \theta_i$ and checks whether $\hat{e}(Sg^{f_i}, B_i) = \hat{e}(g, h_0 h_1^{t_i} h_2^{a_i} h_3^{I_i} h_4^{P'_i})$ or not. If yes, U_i uses the new tuple (B_i, f_i, t_i, P'_i) to replace the previous one and stores them together with $(A_i, e_i, s_i, a_i, T_i, \widehat{A}_i, RK_i)$. Meanwhile, U_i stores P'_i in the read-only memory, which can be used to show the credit points in the future crowdsensing activities. Further, since (B_i, f_i, t_i, P'_i) are managed by U_i, U_i enables to prove the ownership of (B_i, f_i, t_i) cross service providers. The credit points awarded by different service providers can be accumulated and U_i can prove the credit points she has to multiple service providers during the participation of mobile crowdsensing services offered by different service providers.

5 Security Analysis

In this section, we show that SPOON satisfies five security goals defined in 3.3: location privacy, anonymity, data confidentiality, credit balance and greedy user tracing.

5.1 Location Privacy

The sensing region of a task is represented as a matrix $\widehat{L}_{m\times n}$, which is randomized by a random matrix $\widehat{M}_{m\times n}$ to generate $\widehat{N}_{n\times n}$. The location of the mobile user is also transformed to be $\widetilde{N}_{n\times n}$. From two matrices $\widehat{N}_{n\times n}$ and $\widetilde{N}_{n\times n}$, the service provider cannot learn any information about the location of the mobile user or the sensing area of the task. With the multiplication of the invertible matrices $\widetilde{M}_{m\times n}$ and $\widehat{M}_{m\times n}$, respectively, $\widetilde{L}_{m\times n}$'s rank is equal to $\widetilde{N}_{n\times n}$, and $\widehat{L}_{m\times n}$'s rank is equal to $\widehat{N}_{n\times n}$. The service provider computes $N_{n\times n} = \widetilde{N}_{n\times n} \cdot \widehat{N}_{n\times n} = \widetilde{M}^T_{m\times n} \cdot \widetilde{L}_{m\times n} \cdot \widehat{L}^T_{m\times n} \cdot \widehat{M}_{m\times n}$. Thus, the rank of $N_{n\times n}$ is equal to the rank of $\widetilde{L}^T_{m\times n} \cdot \widehat{L}_{m\times n}$. If $N_{n\times n}$'s rank is 0, i.e., $N_{n\times n}$ is zero matrix, one of both entries with the same index in $\widetilde{L}_{m\times n}$ and $\widehat{L}_{m\times n}$ must be 0, indicating that there is no overlapping between the sensing area of the task and the location of the mobile user. Otherwise, there is some overlapping between the sensing area of task and the location the mobile user. If one overlapping grid exists, whose corresponding entry is $\widehat{L}_{ij}$ in $\widehat{L}_{m\times n}$ and is $\widetilde{L}_{ij}$ in $\widetilde{L}_{m\times n}$, respectively, the entries in j-row of $\widehat{N}_{n\times n}$ are nonzero, as well as the entries in j-column of $\widetilde{N}_{n\times n}$. Thus, the service provider enables to know that there are some overlapping locations on the j-column of the sensing area, but it cannot distinguish which location is overlapped from m locations. Further, $\widehat{N}_{n\times n} \cdot \widetilde{N}_{n\times n}$ or $\widetilde{N}_{n\times n} \cdot \widehat{N}_{n\times n}$ cannot give more information to the service provider. The results are the same if the overlapping grids are more than one. Therefore, the sensing area or the location of mobile user would not be exposed to the service provider or other entities.

5.2 Data Confidentiality

We aim to ensure that only the mobile users whose locations can match the sensing area have the capacity to decrypt the corresponding sensing task. In SPOON, the adversaries may be the service provider, unmatched mobile users and external attackers. To resist these adversaries, the task protection consists of two stages. In the first stage, the sensing task is encrypted by the customer under the public keys of the TA and the service provider; in the second one, the service provider partially decrypts the ciphertext using its secret key and then re-encrypts the result for the matched mobile users. Therefore, we demonstrate the task confidentiality in the following two procedures:

- Firstly, the first-stage ciphertext should not be entirely decryptable for the service provider or the mobile users. To be specific, given the first-stage ciphertext (c_1^*, c_2^*, c_3^*) and two plaintexts $(task_1||u_1, task_2||u_2)$, if an adversary can distinguish which one out of $(task_1||u_1, task_2||u_2)$ is the plaintext of (c_1^*, c_2^*, c_3^*), we show how to construct a simulator $\mathcal{S}$ to solve the $q-$DBDHI problem [42].

Given the simplified $q-$DBDHI tuple $g, T_1 = g^{z_1}, T_2 = g^{z_2} \in \mathbb{G}_1, h \in \mathbb{G}_2, Q \in \mathbb{G}_T$, the simulator $\mathcal{S}$'s goal is to determine whether $Q = \hat{e}(g,h)^{\frac{z_1}{z_2}}$ via interactions with the adversary. $\mathcal{S}$ sets $T = T_1$. The adversary possessing the secret key of the service provider, β, can query any chosen message $task||u$ to the simulator $\mathcal{S}$ to obtain the corresponding the ciphertext. Then, $\mathcal{S}$ picks two messages $(task_1||u_1, task_2||u_2)$ and a random bit $b \in \{0,1\}$ to compute the challenge $(c_1^*, c_2^*, c_3^*) = (S^{r_2}, T_2, (task_b||u_b)QH^{r_2})$, where r_2 is a random value chosen from $\mathbb{Z}_p$, and returns $(task_1||u_1, task_2||u_2)$ to the adversary, along with (c_1^*, c_2^*, c_3^*). Finally, the adversary returns $\hat{b} \in \{0,1\}$ to $\mathcal{S}$. If $\hat{b} = b$, $\mathcal{S}$ can address the simplified $q-$DBDHI problem as $Q \stackrel{?}{=} \frac{c_3^*}{(task_b||u_b)\hat{e}(c_1^*,h_0)^{-\frac{1}{\beta}}}$.

The task confidentiality against the adversary, who possesses α, also relies on the simplified $q-$DBDHI problem, given $h, T_1 = g^{z_1}, T_2 = g^{z_2} \in \mathbb{G}_1, h_0 \in \mathbb{G}_2, Q \in \mathbb{G}_T$, the simulator $\mathcal{S}$'s goal is to determine whether $Q = \hat{e}(g,h_0)^{\frac{z_1}{z_2}}$ via interactions with the adversary. The proof is the same as that above with one difference that the challenge is $(c_1^* = T_2, c_2^* = T_1^{r_1}, c_3^* = (task_b||u_b)QG^{r_1})$, where r_1 is a random value chosen from $\mathbb{Z}_p$. Finally, $\mathcal{S}$ can address the simplified $q-$DBDHI problem as $Q \stackrel{?}{=} \frac{c_3^*}{(task_b||u_b)\hat{e}(c_2^*,h)^{-\frac{1}{\alpha}}}$.

- Secondly, the sensing task should only be recovered by the matched mobile users from the second-stage ciphertext. To prevent unmatched mobile users from learning the content of sensing task, the service provider encrypts the sensing task with the temporary public key μ using the proxy re-encryption scheme [42]. Therefore, the security of the second-stage ciphertext can be reduced to the $q-$DBDHI assumption as well.

To guarantee the confidentiality of sensing reports, each mobile user employs the proxy re-encryption scheme [42] to encrypt m_i under the temporary public key $u = g^k$, which is distributed to the mobile users along with the sensing task. The decryption key k is kept by the customer secretly. Therefore, the confidentiality of m_i directly depends on the sematic security of the proxy re-encryption scheme, which can be reduced to the simplified $q-$DBDHI assumption [42].

5.3 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 prove that the mobile user's identity is preserved properly, unless the DDH assumption [44] does not hold. Specifically, if there exists an adversary $\mathcal{A}$ that can identify an honest mobile user out of two challenging identities, we show how to construct a simulator $\mathcal{S}$ to solve an instance of the DDH problem. That is, given a tuple $T_1, T_2, T_3, T_4 \in \mathbb{G}_T$, $\mathcal{S}$ can tell whether exists (z_1, z_2), such that $T_2 = T_1^{z_1}$, $T_3 = T_1^{z_2}$, $T_4 = T_1^{z_1 z_2}$. $\mathcal{S}$ generates $(param, S, T)$, picks two identities (I_0, g^{a_0}), (I_1, g^{a_1}), where $a_0, a_1 \in \mathbb{Z}_p$, and sends them to $\mathcal{A}$. $\mathcal{S}$ acts on behalf of the users I_0 and I_1 to register at TA. $\mathcal{S}$ then interacts with $\mathcal{A}$ in the following interactions:

- $\mathcal{S}$ acts as I_0 honestly to submit the location information. For I_1, in the j-th query, $\mathcal{S}$ randomly chooses $\mu_j \in \mathbb{G}_2$ and simulates the zero-knowledge proof $\mathcal{PK}_3$ to prove its identity interacting with $\mathcal{A}$.
- $\mathcal{S}$ honestly acts on behalf of I_0 to report the data. For I_1, $\mathcal{S}$ sets $H = T_1$, $\mathcal{G} = T_2$. For the j-th query, $\mathcal{S}$ randomly chooses $X_j, v_j \in \mathbb{Z}_p$ and computes $Y_j = T_1^{v_j}, Z_j = \hat{e}(g, h_0^{a_1}) T_2^{X_j v_j}$. $\mathcal{S}$ simulates the zero-knowledge proof $\mathcal{SPK}$ and sends $(X_j, Y_j, Z_j, \mathcal{SPK})$ to $\mathcal{A}$, along with a random sensing report.

$\mathcal{S}$ picks a random bit $b \in \{0,1\}$. If $b = 0$, $\mathcal{S}$ honestly reports the data acting as I_0. If $b = 1$ and $\mathcal{S}$ randomly chooses $X_1 \in \mathbb{Z}_p$ and calculates $\mathcal{G} = T_2$, $Y_1 = T_3$, $Z_1 = \hat{e}(g, h_0^{a_1}) T_4^{X_1}$. Then, $\mathcal{S}$ simulates $\mathcal{SPK}$ and a sensing report, and sends them to $\mathcal{A}$. It is easy to see that the simulation is perfect if $log_{T_1} T_4 = log_{T_1} T_2 log_{T_1} T_3$; otherwise, it contains no information about I_0 and I_1.

Finally, $\mathcal{A}$ returns $\hat{b}$. If $\hat{b} = b$, $\mathcal{S}$ can confirm that there exists (z_1, z_2), such that $T_2 = T_1^{z_1}$, $T_3 = T_1^{z_2}$, $T_4 = T_1^{z_1 z_2}$. Thus, $\mathcal{S}$ resolves the DDH problem.

In the proof of customer's anonymity, a simulator $\mathcal{S}$ simulates the transcript of the zero-knowledge proof of the signature (A, e, s), $\mathcal{PK}_2$, to interact with the adversary $\mathcal{A}$. Since $\mathcal{S}$ can perfectly simulates $\mathcal{PK}_2$, the adversary cannot obtain any identity information about the customer, such that it is impossible to distinguish an honest customer from two for $\mathcal{A}$. Therefore, the customer's anonymity can be fully guaranteed.

5.4 Credit Balance

Credit balance means that no one can own more credit points than the initial credit points plus the credit points awarded by service providers. This is the most significant requirement for credit management from the respective of security. Assume P_0 be the initial credit points and θ_j be the earned points from the service provider in the j-th query. If the adversary $\mathcal{A}$ at most makes $\hat{R}$ reporting queries, and owns final credit points P_f, where $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$, while service providers do not identify the double-reporting, there must exist a simulator $\mathcal{S}$ to conduct a forgery attack on the underlying BBS+ signature [41].

Firstly, we assume that the zero-knowledge proofs $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$ and $\mathcal{SPK}$ are sound. That is, there exist extract algorithms $\mathcal{EX}_1, \mathcal{EX}_2, \mathcal{EX}_3$ and $\mathcal{EX}_S$ to obtain the witnesses of the zero-knowledge proofs, respectively.

Then, we show the simulator $\mathcal{S}$ that interacts with $\mathcal{A}$. $\mathcal{S}$ generates the public parameters $param$, the public keys (T, S) and the secret keys (α, β), and is allowed to access the signature oracle $\mathcal{SO}$ to get the BBS+ signature of an input. $\mathcal{S}$ sends $(param, S, T)$ to $\mathcal{A}$ and interacts with $\mathcal{A}$ as follows:

- $\mathcal{A}$ randomly chooses $C \in \mathbb{G}_1$, $C', \widehat{A} \in \mathbb{G}_2$, generates the proof $\mathcal{PK}_1$ and sends them to $\mathcal{S}$. $\mathcal{S}$ extracts the witness (s', t', a) from $\mathcal{PK}_1$ using $\mathcal{EX}_1$, and then picks a random credit point P_0 and queries the signature oracle $\mathcal{SO}$ to obtain (A, e, s) and (B, f, t).

Finally, $\mathcal{S}$ calculates $s'' = s - s'$, $t'' = t - t'$ and $RK = \widehat{A}^{\frac{1}{\alpha}}$, and returns $(A, e, s'', B, f, t'', P_0, RK)$ to $\mathcal{A}$.

- For the j-th query, $\mathcal{A}$ picks a random $C'_j \in \mathbb{G}_2$ and executes $\mathcal{SPK}$ with $\mathcal{S}$. $\mathcal{S}$ utilizes $\mathcal{EX}_S$ to extract the witness $(B_j, f_j, t_j, t'_j, a_j, I_j, P_j, v_j)$. If (B_j, f_j, t_j) is not an output of $\mathcal{SO}$, it is a forgery of the BBS+ signature. Otherwise, $\mathcal{S}$ queries $\mathcal{SO}$ to obtain a signature (B_j, f_j, t_j) on input $(a_j, P_j + \theta_j, I_j)$. $\mathcal{S}$ receives (B_j, f_j, t_j), computes $t''_j = t_j - t'_j$, and returns (B_j, f_j, t''_j) to $\mathcal{A}$.

Finally, assume $\mathcal{A}$ executes $\hat{R}$ queries. $\mathcal{A}$ wins the game if it can prove $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$. However, if $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$, $\mathcal{A}$ must have conducted a forged BBS+ signautre or double-reported the data. While the BBS+ signature is secure under the $q-$SDH assumption [41], $\mathcal{A}$ cannot forge a BBS+ signature, unless the $q-$SDH assumption [41] does not hold. If $\mathcal{A}$ double-reports the sensing data, it must generate another $\widetilde{Z}_i$, which is unequal to the previous Z_i, in the same time slot. Due to the soundness of zero-knowledge proof protocol, $Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}$ is the only valid Z_i to accompany the specific report identified by X_i and Y_i. Since X_i should be different in two reports, $\hat{e}(g, \widehat{A}_i)$ would be obtained as long as the proof is valid. We assume the proof $\mathcal{SPK}$ is sound. Thus, the success probability of double-reporting for $\mathcal{A}$ is negligible. Therefore, the probability to obtain $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$ is negligible if the $q-$SDH assumption holds.

5.5 Greedy User Tracing

Greedy user tracing consists of two objectives, namely, slandering prevention and hiding prevention. Slandering prevention means that an attacker cannot slander an honest mobile user, and hiding prevention means that a greedy user must be identified by the TA. For the slandering, the attacker releases pieces of reporting transcripts that can link to other reports submitted by an honest mobile user. It is infeasible for the attacker to compute the tracing information about an honest mobile user since the proof $\mathcal{SPK}$ is sound. Therefore, no attacker enables to slander an honest mobile user. In terms of the hiding, the attacker is required to generate different pieces of tracing information without being traced. However, it is impossible for a greedy mobile user to compute Z_i if the pseudo-random function $\mathcal{F}$ is correct.

In summary, SPOON achieves location privacy preservation, sensing data confidentiality, anonymity of both mobile users and customers, credit balance, and greedy user tracing.

6 Extension

In this section, we propose an approach to evaluate the trust levels of sensing reports and a new privacy-preserving location matching mechanism to achieve communication-efficient task allocation for mobile crowdsensing.

6.1 Evaluation on Trust Level

The service provider uploads w sensing reports to the customer, and the customer evaluates the trust level of each report, and distributes credit points to mobile users. Sensing reports have distinct trustworthiness due to the various intelligence of data sources. Furthermore, some mobile users may forge the sensing data or deliver ambiguous, biased data to gain credit points by cheating. Therefore, we propose a fair trust evaluation mechanism as follows:

- The customer generates the weights of sensing data associated with grids in the sensing region $\omega_{z\{z\in\mathcal{L}\}} \in (0, 1]$, such that $\sum_{\{z\in\mathcal{L}\}} \omega_z = 1$, and divides w sensing reports into $|\mathcal{L}|$ groups, where $|\mathcal{L}|$ means the number of the grids in the sensing area. If the data in a sensing report are collected from several grids, this report is in the groups associated with these grids meanwhile.
- For each sensing report in a group $z \in \mathcal{L}$, the customer computes the similarity $V_{i,z}$. If a sensing report is significantly different from the others in the same group (e.g., an opposite result), its similarity is set to be a negative value $V_{i,z} \in [-\gamma, 0]$. Otherwise, the customer sets a positive similarity $V_{i,z} \in (0, \gamma]$ for the report.
- For each sensing report in a group $z \in \mathcal{L}$, the customer computes $\rho_{i,z} = V_{i,z}Q_i$ and $Exp_z = \sum_{i\in\mathcal{Z}} \rho_{i,z}$, where $\mathcal{Z}$ denotes the set of the sensing reports in the group z. Then, the customer sets the trust level of the sensing report to be $\epsilon_{i,z} = (\frac{\rho_{i,z}}{Exp_z})\omega_z\gamma$.
- If the report only contains the data collected from one grid, its trust level is $\epsilon_i = \epsilon_{i,z}$; otherwise, the trust level is set to be the average of the trust levels for all the grids where m_i is collected as $\epsilon_i = AVE_z(\epsilon_{i,z})$.

6.2 Efficiency-enhanced Task Allocation

In SPOON, we use the matrices $\widehat{L}_{m\times n}$ and $\widetilde{L}_{m\times n}$ to represent the sensing area of the task and the location of the mobile user, respectively. To prevent attackers from acquiring the location information, $\widehat{M}_{m\times n}$, $\widetilde{M}_{m\times n}$ are exploited to randomize $\widehat{L}_{m\times n}$, $\widetilde{L}_{m\times n}$. Although this approach is computationally efficient and achieves location protection, the service provider has to define its service region in service setup phase and the communication overhead in task allocation phase is a little heavy, since both the customer and the mobile user are required to transmit the matrices $\widehat{N}_{n\times n}$, $\widetilde{N}_{n\times n}$ to the service provider. To reduce the communication cost, we propose a communication-efficient location matching mechanism by employing the BGN encryption [45].

In the service setup phase, apart from generating $param$ and the secret-public key pair (α, T, T_0), TA setups the BGN encryption. It chooses two random λ-bit primes q_1, q_2, sets $n = q_1 q_2$, and generates two bilinear groups $\mathbb{G}_1, \mathbb{G}_2$ of order n that satisfies the bilinear map $\tilde{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. It also picks a random generator $l \in \mathbb{G}_1$ to compute $l_1 = l^{q_2}$. Thus, the public key of the TA is $(n, \mathbb{G}_1, \mathbb{G}_2, l, l_1, T)$ and the secret key is (α, q_1).

When a customer with $(A, e, s, B, f, t, a, I, P, \widehat{A}, RK)$ has a sensing task $ST = (task, expires, area, \gamma, w)$ to be allocated to the mobile users, it generates $(c_1, c_2, c_3, u, \mathcal{PK}_2)$,

TABLE 3
Computational Overhead of SPOON

Phase	User Registration		Task Allocation			Data Reporting			Credit Assignment	
	Authority	User	Customer	Provider	User	Customer	Provider	User	Provider	User
Point Multiplication	16	19	11	12	9	0	19	25	3	5
Point Addition	12	13	5	8	5	0	14	16	3	5
Bilinear Map	0	4	1	1	2	1	5	2	0	2
Exponentiation in $\mathbb{G}_T$	0	0	6	15	8	1	19	15	0	0
Running Time (ms)	48.361	289.246	97.943	51.925	152.547	55.144	132.398	201.034	9.755	132.288

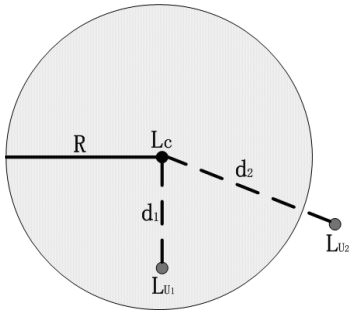


Fig. 3. Sensing Circle and Location.

following the steps given in 4.4.3. The sensing area $area$ is defined as a circle, which is uniquely identified by a center $L_c = (L_{cx}, L_{cy})$ and a radius R, where L_{cx} is the longitude and L_{cy} is the latitude. As shown in Fig. 3, assume the geographical location of a mobile user U_1 is L_{u_1}. If the distance between L_{u_1} and L_c is shorter than R, U_1 is located in the sensing area; otherwise, it is out of the crowdsensing region. To protect the sensing region, the customer picks three random values $r'_x, r'_y, r'_r \in \mathbb{Z}_n$ and computes $C_x = l^{L_{cx}} l_1^{r'_x}$, $C_y = l^{L_{cy}} l_1^{r'_y}$, and $C_R = l^R l_1^{r'_r}$. The customer sends $(c_1, c_2, c_3, expires, \mathcal{PK}_2, C_x, C_y, C_R, \gamma, w)$ to the service provider and the latter releases the task ST as described in 4.4.3.

If a mobile user U_i wants to perform sensing tasks, it chooses a random $\nu \in \mathbb{Z}_p$ to compute $\mu = h^\nu$. Then, U_i retrieves the location information $L_{u_i} = (L_{ix}, L_{iy})$ from the GPS device or the access point, and encrypts it using the public key of the TA as follows: pick random values $r''_x, r''_y \in \mathbb{Z}_n$ to calculate $U_x = l^{L_{ix}} l_1^{r''_x}$, $U_y = l^{L_{iy}} l_1^{r''_y}$. Finally, U_i generates $\mathcal{PK}_3$ and sends $(\mu, U_x, U_y, \mathcal{PK}_3)$ to the service provider.

Upon receiving $(\mu, U_x, U_y, \mathcal{PK}_3)$, the service provider firstly determines whether a sensing task's sensing region covers the location of U_i. For each unexpired task, the service provider computes $X' = \tilde{e}(\frac{C_x}{U_x}, \frac{C_x}{U_x})$, $Y' = \tilde{e}(\frac{C_y}{U_y}, \frac{C_y}{U_y})$ and $Z' = X'Y'$, and sends Z' to the TA, along with (C_R, num) for every task. The TA decrypts Z' and C_R to recover d_i and R, respectively, and checks whether $d_i < R$ to find the set of matching sensing tasks, and returns the task numbers num to the service provider. Then, the service provider generates c_4 and releases $(num, c_2, c_3, c_4, expires, \gamma)$ for U_i. Finally, U_i obtains the sensing task and collects data.

The location matching mechanism is designed from the BGN encryption and the homomorphic property is utilized to compute the distance from the circle center to the locations of mobile users. The security of this mechanism can be reduced to the sematic security of the BGN encryption scheme. Moreover, the customer is required to send (C_x, C_y, C_R) and the mobile user is needed to deliver (U_x, U_y) to the operation center, which are shorter than $\widehat{N}_{n\times n}$ and $\widetilde{N}_{n\times n}$.

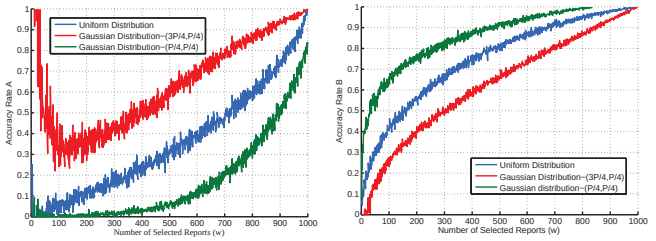
7 Performance Evaluation

In this section, we evaluate the performance of our SPOON in terms of computational and communication overheads, and analyze privacy rate and accuracy rate for credit management.

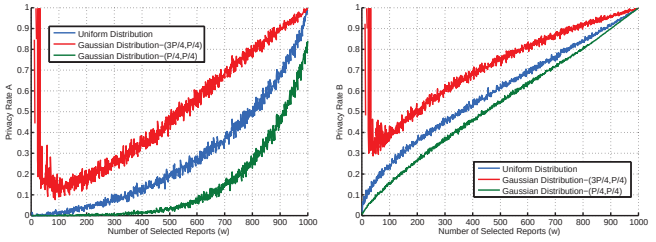
7.1 Computational Overhead

The computational overhead refers to the executing time of our proposed SPOON for each entity in the system. The computational overhead can be represented by the number of the time-consuming cryptographic operations, including point multiplication, point addition, bilinear map and exponentiation in $\mathbb{G}_T$, because the running time of other operations, e.g., multiplication in $\mathbb{G}_T$, addition, multiplication and inverse operations in $\mathbb{Z}_p$, is negligible compared with these four operations. Besides, since the bilinear map is the most time-consuming operation in cryptographic calculations, we utilize the pre-processing technique to reduce the computational burden for each entity. Specifically, TA precomputes the bilinear maps $\{E_i\}_{i=0}^3, F, \{F_i\}_{i=0}^4$ in service setup phase as shown in Appendix A, and the bilinear maps $\{\hat{e}(g, \widehat{A}_i)\}_{i=0}^N$ in user registration phase, where N is the number of registrants. The mobile user U_i also can precompute $\hat{e}(g, \widehat{A}_i)$ in user registration phase. Table III shows the number of the operations executed by each entity in each phase of SPOON, respectively.

We also conduct an experiment to show the time cost of each entity to perform the proposed SPOON. The operations of TA and service provider are performed on a notebook with Intel Core i5-4200U CPU, the clock rate is 2.29GHz and the memory is 4.00 GB. The operation system is 64-bit Windows 10 and the C++ compiler is Visual Studio 2008. 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 R-ATE pairing [48] is utilized to realize the type-3 bilinear pairing. The Barreto–Naehrig curve [48], i.e., $\mathbb{F}_p$-256BN, $E : y^2 = x^3 + 3$ defined over $\mathbb{F}_p$. z is an integer so that $n = 36z^4 + 36z^3 + 18z^2 + 6z + 1$ and

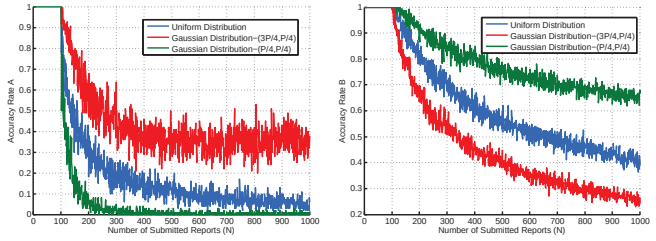


(a) Accuracy Rate A (b) Accuracy Rate B

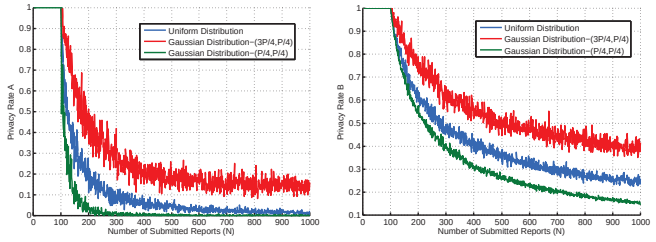


(c) Privacy Rate A (d) Privacy Rate B

Fig. 4. Accuracy and Privacy Rates with N=1000



(a) Accuracy Rate A (b) Accuracy Rate B



(c) Privacy Rate A (d) Privacy Rate B

Fig. 5. Accuracy and Privacy Rates with w=100

$p = 36z^4 + 36z^3 + 24z^2 + 6z + 1$ are prime and $\#E(\mathbb{F}_p) = n$. The embedding degree $k = 12$ is the smallest positive integer with $n|p^4 - 1$. $E[n] \subsetneq E(\mathbb{F}_{p^{12}})$, where $E(n)$ denotes the set of all $n-$torsion points on E. Let $\mathbb{G}_1 = E(\mathbb{F}_p)$, $\mathbb{G}_2$ be the trace–0 order–n subgroup of $E(n)$, and $\mathbb{G}_T$ be the order–n subgroup of $\mathbb{F}^*_{p^{12}}$. p is a large prime with approximately 256 bits. (g, g_0, g_1, g_2, g_3), $(h, h_0, h_1, h_2, h_3, h_4)$ and $\mathcal{G}$ are randomly chosen from $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$, respectively. $\mathcal{H}$ is SHA-256, $\mathcal{F}$ is simulated by AES-256. Table 3 shows the time cost for each entity in every phase of SPOON if it is implemented on the corresponding devices. The running time is less than 300ms for each entity. Therefore, our SPOON is quite efficient to be deployed on mobile devices.

7.2 Communication Overhead

The communication overhead refers to the number of data exchanged among TA, the service provider, the mobile users and the customers. Note that only the data (i.e. payload) of IP packets are taken into account, the packet headers and tailers are not counted in communication overhead, since they are fixed if the number of packets is determined. The binary length of data exchanged between any two entities in SPOON is recorded. The public parameters are set the same as those in the experiment, that is, $|p|$=256 bits and $|q|$=1024 bits. In user registration phase, a registrant, either customer or mobile user, sends a registering request $(I, C, C', \widehat{A}, \mathcal{PK}_1)$ to TA, which is $|I| + 2176$ bits, where $|I|$ is the binary length of the identity, and TA returns $(A, B, s'', t'', e, f, P_0, RK)$ to the registrant, whose binary length is $|P_0| + 2176$ bits, where $|P_0|$ is the binary length of credit point. In task allocation, the customer uploads $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w, \mathcal{PK}_2)$ and the mobile user sends $(\mu, \widehat{N}_{n\times n}, \mathcal{PK}_3)$ to the service provider, which are $4512 + 160n^2 + |expires| + |\gamma| + |w|$ bits and $2976 + 160n^2$ bits, respectively. The service provider responds $(num, c_2, c_3, c_4, expires, \gamma)$, which is $2560 + |num| + |expires| + |\gamma|$ bits, to a matched mobile user or false, 1 bit, to an unmatched one. After the mobile user obtains the sensing data, it generates the sensing report $(num, D_i, D'_i, C'_i, X_i, Y_i, Z_i, Q_i, \tau_j, \mathcal{SPK})$ to the service provider, which is $8864 + |num| + |P_0| + |\tau_j|$ bits. The service provider needs to send 1024-bit W to TA if a mobile user double-submits data, and then sends w sensing reports $(num, D_i, D'_i, Y_i, Q_i, \tau_j)$, which is of binary length $w * (2560 + |num| + |P_0| + |\tau_j|)$, to the customer. Finally, the customer returns $(1024 + |\gamma|)$-bit (ϵ_i, Y_i) to the service provider for each report and the service provider sends $(B_i, t''_i, f_i, \theta_i, Y_i)$ to every mobile user, which is $1856 + |P_0|$ binary bits.

7.3 Credit Analysis

To prevent credit points from disclosing to other entities, each mobile user claims a threshold Q_i, which is proved to be less than its exact credit point P_i, such that the cloud provider can select the sensing reports based on the claimed thresholds. In this way, neither the cloud provider nor the customer enables to learn the precise credit points of mobile users. Unfortunately, this method reduces the accuracy of report selection as the cloud provider may select a sensing report of the mobile user, whose threshold is larger than others', while the credit point has the opposite trend. On the other hand, customers may prefer mobile users to choose the thresholds that are approximate to their credit points, while the privacy of mobile users are sacrificed. Therefore, it seems to be impossible to reconcile the contradiction between privacy and accuracy, because they have an opposite of trends.

To balance this trade-off, it is critical to find a reasonable strategy for mobile users to determine the thresholds. We define four parameters to evaluate privacy and accuracy in credit claiming. Specifically, accuracy rate A denotes the maximum probability of a given threshold in the selected reports can possess top-w credit points in sensing reports. Accuracy rate B denotes the maximum probability that a given credit point in the sensing reports is larger than the minimum threshold in the selected reports. Privacy rate A

means the probability that a given sensing report, whose credit point is larger than the minimum of thresholds in selected reports, has top-w credit point in all sensing reports. Privacy rate B means the probability that a given sensing report would be selected by the service provider, whose credit point is larger than the minimum of thresholds in selected reports. To determine how the threshold choosing strategy affects the defined privacy and accuracy rates, we simulate the credit points of mobile users on Matlab and use different threshold choosing strategies to compute the accuracy rates and the privacy rates. The simulation results are illustrated in Fig. 4 and Fig. 5. We set the number of the mobile users to be 1000 in Fig. 4 and the number of the selected reports to be 100 in Fig. 5. We compare three threshold choosing strategies, the first one is basing on uniform distribution; the second one is basing on Gaussian distribution, in which the mean is three quarters of the number of credit points a mobile user has and the standard deviation is one quarter; the last one is basing on Gaussian distribution, where the mean and the standard deviation are one quarter of the number of credit points. The second strategy achieves the highest accuracy and the third strategy achieves the best privacy preservation on credit points in three strategies.

8 Conclusion

In this paper, we have proposed a strong privacy-preserving mobile crowdsensing scheme with credit management to support the privacy preservation of mobile users and task allocation for customers. The service provider is allowed to select mobile users for sensing task fulfillment according to the sensing areas of tasks and the geographic locations of mobile users, and select the sensing reports based on the credit points of mobile users. The sensitive information, including identities, locations, credit points, sensing tasks and sensing reports are preserved for mobile users and customers during task allocation and report selection. Furthermore, no trusted third party is required to achieve the credit management for mobile users. Finally, we have demonstrated the advantages of the proposed scheme on security and efficiency. It enables service providers to build secure and efficient mobile crowdsensing services that support accurate task allocation and trust management for customers. In the future work, we will design a privacy-preserving context-aware task allocation framework in mobile crowdsensing.

Appendix A
Details of $\mathcal{PK}_1 \sim \mathcal{PK}_3$ and $\mathcal{SPK}$

$\mathcal{SPK}$ demonstrates that the number of credit points P_i a mobile user has is larger than the claimed credit threshold Q_i. Thanks to the zero-knowledge range proof due to Camenisch et al. [46], the mobile user can prove that the value $P_i - Q_i$ is non-negative. We fix the internal of $[0, V]$, where V is the unbound of the credit points of all mobile users, chosen by the service provider. Utilizing the efficient interval proofs in [47], the mobile user demonstrates that $P_i - Q_i$ is one element in the interval $[0, V]$. To instantiate the zero-knowledge proofs $\mathcal{PK}_1 \sim \mathcal{PK}_3$ and $\mathcal{SPK}$, in the setup phase, the service provider generates $\eta = g^{\varphi}$ for a randomly chosen value $\varphi \in \mathbb{Z}_p$, and computes $\phi_\iota = g^{\frac{1}{\iota+\varphi}}$, for each $\iota = 0$ to V. To improve the efficiency, TA can pre-compute $E_0 = \hat{e}(g_0, h)$, $E_1 = \hat{e}(g_1, h)$, $E_2 = \hat{e}(g_2, h)$, $E_3 = \hat{e}(g_3, h)$, and $F = \hat{e}(g, h)$, $F_0 = \hat{e}(g, h_0)$, $F_1 = \hat{e}(g, h_1)$, $F_2 = \hat{e}(g, h_2)$, $F_3 = \hat{e}(g, h_3)$, and $F_4 = \hat{e}(g, h_4)$. These parameters are included in the public parameters as $param \cap \{E_0, E_1, E_2, E_3, F_0, F_1, F_2, F_3, F_4\}$. The service provider also releases the parameters $\{S, L_{m\times n}, \eta, \{\phi_\iota\}_{\iota\in[0,V]}\}$. To deduce the number of interactions in zero-knowledge proofs, we utilize the Fiat-Shamir transformation, where the hash function $\mathcal{H}$ can be viewed as a random oracle. The details of $\mathcal{PK}_1 \sim \mathcal{PK}_3$ and $\mathcal{SPK}$ are shown below.

$\mathcal{PK}_1\{(s', t', a) : C = g_1^{s'} g_2^{a} \wedge C' = h_1^{t'} h_2^{a} \wedge \widehat{A} = h_0^{a}\}$.

1) The TA sends a random challenge $R \in \mathbb{Z}_p$.
2) The registrant randomly chooses $\rho_{s'}, \rho_{t'}, \rho_a \in \mathbb{Z}_p$ and computes $T_1 = g_1^{\rho_{s'}} g_2^{\rho_a}$, $T_2 = h_1^{\rho_{t'}} h_2^{\rho_a}$ and $T_3 = h_0^{\rho_a}$.
3) The registrant computes $c = \mathcal{H}(T_1, T_2, T_3, R)$.
4) The registrant computes $z_{s'} = \rho_{s'} - cs'$, $z_{t'} = \rho_{t'} - ct'$, $z_a = \rho_a - ca$, and sends $c, z_{s'}, z_{t'}, z_a$ to the TA.
5) The TA computes $T_1' = C^c g_1^{z_{s'}} g_2^{z_a}$, $T_2' = C'^c h_1^{z_{t'}} h_2^{z_a}$, $T_3' = \widehat{A}^c h_0^{z_a}$ and accepts the proof if $c = \mathcal{H}(T_1', T_2', T_3', R)$; otherwise, rejects.

$\mathcal{PK}_2\{(A, e, s, a, I) : \hat{e}(A, T_0 h^e) \stackrel{?}{=} \hat{e}(g_0 g_1^s g_2^a g_3^I, h)\}$.

1) The service provider sends a random challenge $R \in \mathbb{Z}_p$.
2) The customer randomly chooses $r_1, r_2 \in \mathbb{Z}_p$ to compute $B_1 = g_2^{r_1} g_3^{r_2}$, $B_2 = A g_3^{r_1}$. The customer picks random $\rho_{r_1}, \rho_{r_2}, \rho_{\delta_1}, \rho_{\delta_2}, \rho_e, \rho_s, \rho_a, \rho_I \in \mathbb{Z}_p$ and computes $T_1 = g_2^{\rho_{r_1}} g_3^{\rho_{r_2}}$, $T_2 = B_1^{-\rho_e} g_2^{\rho_{\delta_1}} g_3^{\rho_{\delta_2}}$, $T_3 = \hat{e}(B_2, h)^{-\rho_e} \hat{e}(g_3, T_0)^{\rho_{r_1}} E_3^{\rho_{\delta_1}} E_1^{\rho_s} E_2^{\rho_a} E_3^{\rho_I}$.
3) The customer computes $c = \mathcal{H}(T_1, T_2, T_3, R)$.
4) The customer computes $z_{r_1} = \rho_{r_1} - cr_1$, $z_{r_2} = \rho_{r_2} - cr_2$, $z_{\delta_1} = \rho_{\delta_1} - c\delta_1$, $z_{\delta_2} = \rho_{\delta_2} - c\delta_2$, $z_e = \rho_e - ce$, $z_s = \rho_s - cs$, $z_a = \rho_a - ca$, $z_I = \rho_I - cI$, and sends $c, B_1, B_2, z_{r_1}, z_{r_2}, z_{\delta_1}, z_{\delta_2}, z_e, z_s, z_a, z_I$ to the service provider.
5) The service provider computes $T_1' = B_1^c g_2^{\rho_{r_1}} g_3^{\rho_{r_2}}$, $T_2' = B_1^{-z_e} g_2^{z_{\delta_1}} g_3^{z_{\delta_2}}$, $T_3' = (\frac{\hat{e}(B_2, T_0)}{E_0})^c \hat{e}(B_2, h)^{-z_e} \hat{e}(g_3, T_0)^{z_{r_1}} E_3^{z_{\delta_1}} E_1^{z_s} E_2^{z_h} E_3^{z_I}$ and accepts if $c = \mathcal{H}(T_1, T_2, T_3, R)$; otherwise, rejects.

$\mathcal{PK}_3\{(A_i, e_i, s_i, a_i, I_i) : \hat{e}(A_i, T g^{e_i}) \stackrel{?}{=} \hat{e}(g_0 g_1^{s_i} g_2^{a_i} g_3^{I_i}, g)\}$ is the same as $\mathcal{PK}_2$

$$\mathcal{SPK}\left\{\begin{array}{c}(B, f, t, t', a, I, P, v) : \\ \hat{e}(T g^f, B_i) \stackrel{?}{=} \hat{e}(g, h_0 h_1^t h_2^a h_3^I h_4^P) \wedge \\ C' = h_1^{t'} h_2^a h_3^I h_4^P \wedge \\ P > Q \wedge \\ Y = H^v \wedge \\ Z = \hat{e}(g, \widehat{A}) \mathcal{G}^{Xv}\end{array}\right\}(num).$$

1) The mobile user randomly chooses $r_1, r_2, r_3, r_4 \in \mathbb{Z}_p$ to compute $B_1 = h_2^{r_1} h_3^{r_2}$, $B_2 = T h_3^{r_1}$, $S_1 = g_2^{r_3} g_3^{r_4}$, $S_2 = \phi_{P-Q} g_3^{r_4}$. The mobile user picks random $\rho_{r_1}, \rho_{r_2}, \rho_{r_3}, \rho_{r_4}, \rho_f, \rho_t, \rho_{t'}, \rho_a, \rho_I, \rho_P, \rho_v, \rho_{\omega_1}, \rho_{\omega_2}, \rho_{\omega_3}, \rho_{\omega_4} \in \mathbb{Z}_p$ and computes $T_1 = h_1^{\rho_{t'}} h_2^{\rho_a} h_3^{\rho_I} h_4^{\rho_P}$,

$T_2 = h_2^{\rho_{r_1}} h_3^{\rho_{r_2}}$, $T_3 = B_1^{-\rho_f} h_2^{\rho_{\omega_1}} h_3^{\rho_{\omega_2}}$, $T_4 = \hat{e}(g, B_2)^{-\rho_f} \hat{e}(T, h_3)^{\rho_{r_1}} F_3^{\rho_{\omega_1}} F_1^{\rho_t} F_2^{\rho_a} F_3^{\rho_I} F_4^{\rho_P}$, $T_5 = g_2^{\rho_{r_3}} g_3^{\rho_{r_4}}$, $T_6 = S_1^{-\rho_P} g_2^{\rho_{\omega_3}} g_3^{\rho_{\omega_4}}$, $T_7 = \hat{e}(g_3, \eta)^{\rho_{r_4}} E_3^{\rho_{\omega_4}} \hat{e}(S_2, h)^{-\rho_P}$, $T_8 = H^{\rho_v}$, $T_9 = \mathcal{G}^{X\rho_v}$.

2) The user computes $c = \mathcal{H}(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, num)$.

3) The user computes $z_{r_1} = \rho_{r_1} - cr_1$, $z_{r_2} = \rho_{r_2} - cr_2$, $z_{r_3} = \rho_{r_3} - cr_3$, $z_{r_4} = \rho_{r_4} - cr_4$, $z_f = \rho_f - cf$, $z_t = \rho_t - ct$, $z_{t'} = \rho_{t'} - ct'$, $z_a = \rho_a - ca$, $z_I = \rho_I - cI$, $z_P = \rho_P - cP$, $z_v = \rho_v - cv$, $z_{\omega_1} = \rho_{\omega_1} - cr_1\omega_1$, $z_{\omega_2} = \rho_{\omega_2} - cr_2\omega_2$, $z_{\omega_3} = \rho_{\omega_3} - c(P-Q)r_3$, $z_{\omega_4} = \rho_{\omega_4} - c(P-Q)r_4$, and sends $c, B_1, B_2, S_1, S_2, z_{r_1}, z_{r_2}, z_{r_3}, z_{r_4}, z_f, z_t, z_{t'}, z_a, z_I, z_P, z_v, z_{\omega_1}, z_{\omega_2}, z_{\omega_3}, z_{\omega_4}$ to the service provider.

4) The service provider computes $T_1' = C'^c h_1^{z_{t'}} h_2^{z_a} h_3^{z_I} h_4^{z_P}$, $T_2' = B_1^c h_2^{z_{r_1}} h_3^{z_{r_2}}$, $T_3' = B_1^{-z_f} h_2^{z_{\omega_1}} h_3^{z_{\omega_2}}$, $T_4' = (\frac{\hat{e}(T,B_2)}{F_0})^c \hat{e}(g, B_2)^{-z_f} \hat{e}(T, h_3)^{z_{r_1}} F_3^{z_{\omega_1}} F_1^{z_t} F_2^{z_a} F_3^{z_I} F_4^{z_P}$, $T_5' = S_1^c g_2^{z_{r_3}} g_3^{z_{r_4}}$, $T_6' = S_1^{-Q_c} S_1^{-z_P} g_2^{z_{\omega_3}} g_3^{z_{\omega_4}}$, $T_7' = (\hat{e}(S_2, \eta h^{-Q}) F^{-1})^c \hat{e}(g_3, \eta)^{z_{r_3}} E_3^{z_{\omega_3}} \hat{e}(S_2, h)^{-z_P}$, $T_8' = Y^c H^{z_v}$, $T_9' = (\frac{Z}{\hat{e}(g,\hat{A})})^c \mathcal{G}^{Xz_v}$, and accepts the proof if $c = \mathcal{H}(T_1', T_2', T_3', T_4', T_5', T_6', T_7', T_8', T_9', num)$; otherwise, rejects.

REFERENCES

[1] J. Ni, K. Zhang, X. Lin, Q. Xia, and X. Shen, "Privacy-preserving mobile crowdsensing for location-based applications," in *Proc. ICC*, 2017, pp. 1–6.

[2] R.K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, 2011.

[3] Y. Hui, Z. Su, and S. Guo, "Utility based data computing scheme to provide sensing service in Internet of Things," *IEEE Trans. Emerging Tpoics in Computing*, 2017, DOI: 10.1109/TETC.2017.2674023.

[4] A. Doan, R. Ramakrishnan, and A.Y. Halevy, "Crowdsourcing systems on the world-wide web," *Commun. ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[5] X. Zhang, Z. Yang, Y. Liu, J. Li, and Z. Ming, "Toward efficient mechanisms for mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1760–1771, 2017.

[6] L. Xiao, T. Chen, C. Xie, H. Dai, and H.V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1535–1545, 2018.

[7] J. Ni, A. Zhang, X. Lin, and X. Shen, "Security, privacy and fairness in fog-based vehicular crowdsensing," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 146–152, 2017.

[8] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 53, no. 8, pp. 75–81, 2015.

[9] I. Krontiris, M. Langheinrich, and K. Shilton, "Trust and privacy in mobile experience sharing: Further challenges, and avenues for research," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 50–55, 2014.

[10] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Trans. Dependable Secur. Comput.*, 2018, DOI: 10.1109/TDSC.2018.2791432.

[11] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "AnonySense: Privacy-aware people-centric sensing," in *Proc. Mobisys*, 2008, pp. 211–224.

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

[13] T. Dimitriou, I. Krontiris, and A. Sabouri, "PEPPeR: A querier privacy enhancing protocol for participatory sensing," in *Proc. MobiSec*, 2012, pp. 93–106.

[14] F. Qiu, F. Wu, and G. Chen, "Privacy and quality preserving multimedia data aggregation for participatory sensing system," *IEEE Trans. Mob. Comput.*, vol. 14, no. 6, pp. 1287–1299, 2015.

[15] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proc. MobiCom*, 2011, pp. 145–156.

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

[17] H. Mousa, S.B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Comput. Netw.*, vol. 90, no. 29, pp. 49–73, 2015.

[18] D. Christin, C. Rokopf, M. Hollick, L.A. Martucci, and S.S. Kanhere, "IncogniSense: An anonymity-preserving reputation framework for participatory sensing applications," *Pervasive Mob. Comput.*, vol. 9, no. 3, pp. 353–371, 2013.

[19] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. MDM*, 2014, pp. 73–82.

[20] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S.S. Kanhere, "Privacy-preserving collaborative path hiding for participatory sensing applications," in *Proc. MASS*, 2011, pp. 341–350.

[21] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang, "Incentivizing crowdsensing with location-privacy preserving," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 10, pp. 6940–6952, 2017.

[22] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," in *Proc. VLDB*, vol. 7, no. 10, pp. 919–930, 2014.

[23] Q. Ma, S. Zhang, T. Zhu, K. Liu, L. Zhang, W. He, and Y. Liu, "PLP: Protecting location privacy against correlation analyze attack in crowdsensing," *IEEE Trans. Mob. Comput.*, vol. 16, no. 9, pp. 2588–2598, 2017.

[24] E.D. Cristofaro and C. Soriente, "Extended capabilities for a privacy-enhanced participatory sensing infrastructure (PEPSI)," *IEEE Trans. Inf. Forensic Secur.*, vol. 8, no. 12, pp. 2021–2033, 2013.

[25] F. Günther, M. Manulis, and A. Peter, "Privacy-enhanced participatory sensing with collision resistance and data aggregation," in *Proc. CANS*, 2014, pp. 321–336.

[26] J. Zhou, Z. Cao, and X. Dong, "Secure and efficient fine-grained multiple file sharing in cloud-assisted crowd sensing networks," *Peer Peer Netw. Appl.*, vol. 9, no. 4, pp. 1–21, 2016.

[27] X. Chen, X. Wu, X.-Y. Li, Y. He, and Y. Liu, "Privacy-preserving high-quality map generation with participatory sensing," in *Proc. IEEE INFOCOM*, 2014, pp. 2310–2318.

[28] S. Rahaman, L. Cheng, D. D. Yao, H. Li, and J.-M. J. Park, "Provably secure anonymous-yet-accountable crowdsensing with scalable sublinear revocation," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 384–403, 2017.

[29] L. Kazemi and C. Shahabi, "TAPAS: Trustworthy privacy-aware participatory sensing," *Knowl. Inf. Syst.*, vol. 7, no. 1, pp. 105–128, 2013

[30] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mob. Comput.*, 2018, DOI: 10.1109/TMC.2018.2861393.

[31] H. Jin, L. Su, D. Chen, H. Guo, K. Nahrstedt, and J. Xu, "Thanos: Incentive mechanism with quality awareness for mobile crowd sensing," *IEEE Trans. Mob. Comput.*, 2018, DOI: 10.1109/TMC.2018.2868106.

[32] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mob. Comput.*, vol. 17, no. 12, pp. 2982–2997, 2018.

[33] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Robust incentive tree design for mobile crowdsensing," in *Proc. ICDCS*, pp. 458–468, 2017.

[34] L. Kazemi, C. Shahabi, and L. Chen,"Geotrucrowd: trustworthy query answering with spatial crowdsourcing," in *Proc. ACM SIGSPATIAL GIS*, 2013, pp. 314-323.

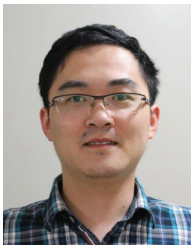
[35] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Trans. Mob. Comput.*, vol. 13, no. 12, pp. 2777–2789, 2014.

[36] L. Ma, X. Liu, Q. Pei, and Y. Xiang, "Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing," *IEEE Trans. Serv. Comput.*, 2018, DOI: 10.1109/TSC.2018.2825986.

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

[38] 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, 2018.

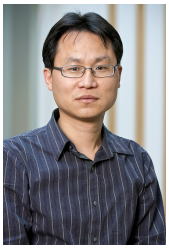
[39] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdSensing," *IEEE Trans. Mob. Comput.*, 2018, DOI: 10.1109/TMC.2018.2879098.
[40] H. Li, T. Li, W. Wang, and Y. Wang, "Dynamic participant selection for large-scale mobile crowd sensing," *IEEE Trans. Mob. Comput.*, 2018, DOI: 10.1109/TMC.2018.2884945.
[41] M.H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-TAA," in *Proc. SCN*, 2006, pp. 111–125.
[42] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in *Proc. NDSS*, 2005, pp. 29–43.
[43] D. Pointcheval and O. Sanders, "Short Randomizable Signatures," in *Proc. CT-RSA*, 2016, pp. 111–126, 2016.
[44] D. Cash, E. Kiltz, and V. Shoup, "The twin Diffie-Hellman problem and applications," in *Proc. Eurocrypt*, 2008, pp. 127–145.
[45] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. TCC*, 2005, pp. 325–341.
[46] J. Camenisch, R. Chaabouni, and A. Shela, "Efficient protocols for set membership and range proofs," in *Proc. Asiacrypt*, 2008, pp. 234–252.
[47] M.H. Au, J.K. Liu, J. Fang, Z.L. Jiang, W. Susilo, and J. Zhou, "A new payment system for enhancing location privacy of electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 63, no, 1, pp. 3–18, 2014.
[48] J. Fan, F. Vercauteren, and I. Verbauwhede, "Faster $\mathbb{F}_p$–Arithmetic for Cryptographic Pairings on Barreto-Naehrig Curves", in *Proc. CHES*, 2009, pp. 240–253.



Jianbing Ni (M'18) received the Ph.D. degree in Electrical and Computer Engineering from University of Waterloo, Waterloo, Canada, in 2018, and received the B.E. degree and the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He is currently a postdoctoral fellow at the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. His research interests are applied cryptography and network security, with current focus on cloud computing, smart grid, mobile crowdsensing and Internet of Things.

Kuan Zhang (S'13-M'17) received the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Canada, in 2016. He was also a postdoctoral fellow with Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since 2017, he has been an assistant professor at the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, USA. His research interests include security and privacy for mobile social networks, e-healthcare systems, cloud computing and cyber physical systems.

Qi Xia received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China (UESTC) in 2002, 2006, and 2010, respectively. She was a Visiting Scholar with the University of Pennsylvania, Philadelphia, PA, USA, from 2013 to 2014. She is currently the PI in cyber security with the National Key Research and Development Program of China. She is also the Vice Dean with the Center for Cyber Security and currently an Associate Professor with UESTC. She was a recipient of the National Scientific and Technological Progress Second Prize in 2012.



Xiaodong Lin (M'09-SM'12-F'17) received the PhD degree in Information Engineering from Beijing University of Posts and Telecommunications, China, and the PhD degree (with Outstanding Achievement in Graduate Studies Award) in Electrical and Computer Engineering from the University of Waterloo, Canada. He is currently an associate professor in the School of Computer Science at the University of Guelph, Canada. His research interests include computer and network security, privacy protection, applied cryptography, computer forensics, and software security. He is a Fellow of the IEEE.

Xuemin (Sherman) Shen (M'97-SM'02-F'09) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.

Dr. Shen received the R.A. Fessenden Award in 2019 from IEEE, Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society. He has also received the Excellent Graduate Supervision Award in 2006 and the Outstanding Performance Award 5 times from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, the Chair for the IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL and the Vice President on Publications of the IEEE Communications Society.