Hindawi
Security and Communication Networks
Volume 2021, Article ID 5518168, 20 pages
https://doi.org/10.1155/2021/5518168

Research Article

Anonymous Data Reporting Strategy with Dynamic Incentive Mechanism for Participatory Sensing

Yang Li[1], **Hongtao Song**[1], **Yunlong Zhao**[2], **Nianmin Yao**[3] **and Nianbin Wang**[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[2]*College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China*
[3]*School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Hongtao Song; songhongtao@hrbeu.edu.cn

Received 28 February 2021; Accepted 21 May 2021; Published 1 June 2021

Academic Editor: David Megías

Participatory sensing is often used in environmental or personal data monitoring, wherein a number of participants collect data using their mobile intelligent devices for earning the incentives. However, a lot of additional information is submitted along with the data, such as the participant's location, IP and incentives. This multimodal information implicitly links to the participant's identity and exposes the participant's privacy. In order to solve the issue of these multimodal information associating with participants' identities, this paper proposes a protocol to ensure anonymous data reporting while providing a dynamic incentive mechanism simultaneously. The proposed protocol first establishes a submission schedule by anonymously selecting a slot in a vector by each member where every member and system entities are oblivious of other members' slots and then uses this schedule to submit the all members' data in an encoded vector through bulk transfer and multiplayer dining cryptographers networks (DC-nets) . Hence, the link between the data and the member's identity is broken. The incentive mechanism uses blind signature to anonymously mark the price and complete the micropayments transfer. Finally, the theoretical analysis of the protocol proves the anonymity, integrity, and efficiency of this protocol. We implemented and tested the protocol on Android phones. The experiment results show that the protocol is efficient for low latency tolerable applications, which is the cases with most participatory sensing applications, and they also show the advantage of our optimization over similar anonymous data reporting protocols.

1. Introduction

Participatory sensing is a feasible paradigm in which individuals with sensing and computing devices collectively share information to measure and map phenomena of common interest. The development profits from most mobile terminal equipment (e.g., smart phones) have multiple embedded sensors such as GPS, accelerometer, and camera. The mobile participatory sensing completes the functionality of traditional Wireless Sensor Networks (WSN) with the help of mobile phones. Moreover, participatory sensing has its advantages: more available resources and better spatiotemporal coverage from ubiquitous mobile phones, more intelligent computing from people's participation, and high scalability without specialized infrastructure. Many participatory sensing applications have been widely used in daily life and researches. These applications are mainly used in two aspects [1]: (1) people-centric monitoring applications, which include personal health monitoring applications, sport and exercises monitoring applications, social media enhancing applications, and price auditing applications and (2) environment-centric sensing applications which are interested in the phenomena around people in the community scale, for example, air quality monitoring applications, thermal and noise monitoring applications, traffic jam alerts, wireless indoor localization, and small cell network monitoring applications.

Although participatory sensing is hugely convenient for data collection through people participation, it also raises some new research challenges. The primary concern is the privacy of participants who are contributing sensing data, since collected data contains or adds some multimodal

information to implicitly connect the identities of participants which maybe expose the participant's personal or private information. When a participant submits data, such as the participant's registration information, IP or even geographic location may be associated with the participant's identity. At the same time, additional information such as incentives and reputation accompanied by data quality may also be connected to participants. For the former, it is an inevitable problem brought about by the current network structure. For the latter, with incentives as an example, unlike other traditional applications, participatory sensing is not always based on gratuitous participation. Incentives are indispensable to attract users to participate and to encourage the participants to provide sufficient sensing data and improve the quality of the sensing service. However, it is more challenging to disassociate this multimodal information from participant identities to protect participant privacy and provide incentives, especially when anonymity is provided as the protection for privacy [2]. Furthermore, anonymity allows greedy users to submit multiple data reports of the same sensing task, copy or exchange identities for more incentives, and it is difficult to detect. This not only increases the cost of collecting data but also undermines the fairness of the system, which may make participants and task requesters no longer willing to join.

Most of the researches have been proposed to address them separately, only a few researches address all of them simultaneously. In works [2–5], authors assume that the communications between users and the task requesters are anonymized (e.g., Mix networks) with the help of a trusted third party (TTP). The trusted third party is often used for providing privacy protection and incentive mechanisms and can help users hide their real identities (e.g., using pseudonyms) and transfer micropayments. However, relying on a trusted third party increases the complexity of the system and may not be able to convince participants of its credibility. For TTP-free scheme, works [6, 7] use the blind signature mechanism to sign a token and a pseudonym, and verify them by comparing for determining whether the participant is payable. However, the pseudonym and the signed token are independent. If the participants collude, they maybe exchange or duplicate others' pseudonyms for earning more payments. Our previous work [8] proposed a privacy protection scheme with incentive mechanism for mobile crowd-sensing. It uses slot reservation based on shuffle to establish a submission schedule by anonymously assigning a slot in a vector to each member in a group of participants. The shuffle leverages the participants to disrupt the order of the multiple encrypted reservation message. Data submission based on bulk transfer and multiplayer DC-nets uses members' reserved slots to transmit all thedata in an encoded vector to service provider. Finally, Blind signature ensures the incentive mechanism. The consumption of multiple encrypted reservation messages is good enough for usual MCS application, but the encryption is still a very complicated operation. The incentive mechanism ensures that participants receive payment while protecting their privacy from exposure. It will be better if it can enable each participant to get a dynamic incentive to encourage the participant to provide higher data quality.

In order to address the above issues, this paper proposes a strategy to protect participant's privacy through data reporting anonymity while providing incentives in the form of micropayments. Our contributions are as follows: (1) we present a data reporting protocol for guaranteeing anonymity based on multiplayer DC-nets and bulk transfer with an incentive strategy followed by works [9–13], and adapted to participatory sensing in Yao et al. [14] and our previous work [8]. (2) We optimize slot reservation and data submission stage for efficiency, which uses random slot selection instead of previous multiple encryption shuffles. And, the bulk transfer is simplified for this protocol maintaining the same security as our previous work. (3) And, we improve the incentive mechanism to support anonymously dynamic micropayments for different participants. It reversely uses the blind signature mechanism to carry dynamic incentives, and uses the similar mechanism as data submission to confuse the participant's identity. The theoretical analysis in this paper proves the integrity, anonymity, and efficiency of the protocol. Finally, we prove the practicality of this protocol for participatory sensing applications by implementation and performance evaluation.

The rest of this paper is organized as follows. Section 2 reviews some related work and preliminaries. Section 3 presents the system model and the adversary model. Section 3 proposes our protocol. And the protocol analysis and performance evaluation is shown in Sections 5 and 6, respectively. At last, Section 7 concludes this paper.

2. Related Works

Participatory sensing is an active research area full of open questions and challenges. The main issue is to protect the privacy of participants who contribute sensing data, because the private information of data contributors may be exposed during data submission. Some studies have been proposed to solve the privacy problem in the data publishing [15]. The first solution is to restrict the data publishing, by setting some pre-configurations to decide whether or not to publish the given data to protect the privacy of users. This requires prior interaction with the user to determine these configurations [16]. Another solution is to control the granularity of the data, that is, to reduce the accuracy of the data to a certain extent, and then sending modified data instead of accurate data to protect privacy [1, 17]. Furthermore, data perturbation [18] is also proposed to perturb individual data, while allowing the reconstruction of the original data statistics in the crowd. Although all these solutions are based on actual applications, they are a compromise between privacy protection and data quality.

Pseudonyms have been widely used to hide the true identities of participants to ensure the anonymity of data reports [19–21]. It is not enough to protect privacy by using only the pseudonyms since malicious attackers can infer the true identity of the participant from the network context and location information [22]. Due to pseudonymity limitation, k-anonymity model [23] was proposed, which aims to

reduce the data granularity until making a given report maps onto at least kk other reports. It releases data with lower precision through generalization and concealment technology, so that each record has at least the same quasi-identifier attribute value as other $k-1$ records in the data table, thereby reducing privacy leakage caused by link attacks [24, 25]. However, this scheme is usually only effective for protecting sensitive data; protecting participant identity to the level of k individuals is not the same as protecting the sensitive data, as network context and location information can still expose the user's identity.

Secure Multiparty Computation (SMC) [26] is often used to protect user privacy in distributed systems. It protects the input data privacy of mutually distrustful participants in a group without the help of the trusted third party, jointly complete certain multi-input calculations, and ensure the correctness of the output results. Existing solutions can be basically divided into two categories: oblivious transfer scheme [27] and encoding and homomorphic encryption scheme [28]. Among them, encoding and homomorphic encryption scheme is often used to solve the problem of multiparticipant collaborative transmission of data [29, 30], and there are many studies to improve the encryption or encoding [31, 32]. It can allow participants to use encrypted or encoded data to calculate, and finally obtain the calculation results, but cannot know the input data of each participant. Because using SMC cannot obtain individual data provided by participants, it is usually suitable for systems that only need to obtain statistical results. Furthermore, if malicious entities collude with stronger eavesdroppers, the privacy of participants may still be exposed.

Differential privacy protection is widely used for privacy protection. It hides the impact of a single datum by adding random disturbances to the published data, so that the calculation result does not reveal too much information of a single record in the data set [33]. At present, there are many researches focusing on the security of balancing random disturbances and privacy protection [34–36]. In order to ensure the preserving of privacy in the data collection, works [37–39] proposed several local differential privacy schemes, which allow users to perturb the data locally, and then send the results with the perturbation to the task requester. Sei and Ohsuga provide an estimation technique, which estimates the true data distribution based on the reported data on the data collector's side [40]. Ni et al. [41] propose a differentially private double auction mechanism to select participants, and the use of utility function for pricing. However, these solutions are still difficult to prevent malicious service providers from colluding with global eavesdroppers to link each datum with its contributors. Moreover, it usually disturbs all the original data excessively to ensure the anonymity of the data.

Some studies have improved the system architecture. For privacy and report integrity, Cornelius et al. [42] propose to use trusted entities to reduce the sensitivity of the data, while using a mix network to avoid network tracking. Work [43] mainly leverages an idea of blind matching of data reports and sensing tasks with the help of a third party. Although both of these solutions are good at protecting the privacy of participants, they still rely on a trusted third party and unrealistic assumptions. On the other hand, in [20, 44, 45], decentralized approaches to protect privacy are proposed. But, their technologies are still focused on specific scenarios, and it is difficult to apply to all applications.

Unlike previous works, communication anonymity techniques aim to hide the identity of the sender and receiver (in network) by removing identifying patterns from the network flow. Therefore, it preserves privacy protection during the data collection, and can submit the original individual data without sensitive information in general. Most of these techniques are based on laundering traffic through intermediates, nodes, and encryption. Dissent proposed in Corrigangibbs and Ford [10] leverage the verifiable shuffles to establish a transmission plan and then uses multiplayer DC-nets and bulk transfer for anonymous data transmission. Yao et al. [14] improve the slot reservation and data submission stage, so that participants can perform data transmission without negotiation and also preserve anonymity. The modified approach is applicable to Mobile crowdsensing (MCS), but the calculation of the slot reservation stage suffers from relatively high latency. Work [8] improved Yao's scheme, reducing the number of encryptions in the slot reservation stage while maintaining the same anonymity, but the multi-encrypted shuffle mechanisms still is a complicated operation.

In addition, most of the previous works consider privacy issues without considering incentive mechanisms. Few researches address both issues at the same time, since protecting the privacy and providing incentives simultaneously is more challenging [2], especially when anonymity is provided as the privacy-preserving mechanism. Blind signatures [46, 47] have been widely used in anonymous electronic payment systems and digital currencies. It is natural to use blind signatures to achieve privacy protection and credit. However, blind signature technology prevents the signer from linking to any specific user. Using blind signature technology alone may cause malicious users to harm other users, steal their credit lines, and spend these credit lines without being discovered. Works [3, 4] are the works that address both issues simultaneously. However, those solutions pay more attention to data content protection and anonymous incentives over communication anonymity. The authors assume that the communications between users and the SP are anonymized (e.g., Mix networks) which rely on the anonymization of the third party. Similarly, work by [48] relies on TOR anonymization network for communication anonymity which is the third party solution by just switching the problem from the task request server to the third party server. In work by [49], a multiple encryption strategy is used to keep the untraceability of pseudonymity and the network service provider is assumed to be credible and the only entity who could know the identity. Work in [43] mainly leverages an idea of blind matching of data reports and sensing tasks with the help of the third party. Gao et al. proposed an anonymous approach based on proxy ring signature to hide the identities through groups[50]. A similar scheme is used in the smart grid which has a creditable entity in the system naturally [51].

For TTP-free scheme, Li and Cao [4] present another mechanism for TTP-free scheme to use blind signature, partially blind signature, and an extended Merkle tree technique to protect user privacy and prevent abuse attacks. However, in the case of collusion between other participants and system entities, the identities of honest participants can still be exposed. Wang et al. [6] used blind signatures and differential data to hide the real identity of user and provide a reputation mechanism . The server needs to sign two sets of signatures to verify the authenticity of the information while remaining anonymous. And, Wu et al. [7] leverage blind signatures and Hash to provide a similar method. Both of them use blind signature to sign a token and a pseudonym, and verify them by comparing for determining whether the participant is payable. However, the pseudonym and the signed token are independent. If the participants collude with each other, they may exchange or duplicate others' pseudonyms for earning more payments.

3. Problem Statement

For the usual participatory sensing application, the system includes a task requester and a number of participants with their own mobile sensing device. The participants voluntarily use their devices to collect sensor data based on the task requester's request and report the data to the task requester periodically. Then, the task requester processes the reported data to extract useful statistics in varied forms according to the realistic requirements. However, the participants' sensitive information may be revealed during sensor data collection, which puts their privacy at risk. Privacy-preserving is an indispensable mechanism for the system and also obligatory to encourage volunteers. Obtaining incentives is the purpose of the participants to perform tasks, and the incentive mechanism is indispensable, which ensures the participation of users and revokes malicious participants.

The content of the reported data and the process of reporting the data to the server threaten the participants' privacy. Considering current Internet architecture, network-based traceback techniques can trace the origin of most Internet data packets [52]. Furthermore, the participants' IP addresses may expose their location, even help in inferring their real identity through analyzing of the reporting pattern. Therefore, anonymous communication is indispensable for privacy protection. This paper presents an approach to break the link between the reported data that do not contain sensitive information and the identity of the data reporter to ensure the anonymity of the communicator, so that malicious attackers cannot determine the participant submitting the data, thereby protecting privacy.

3.1. System Architecture. The goal of this paper was to hide the data reporter's identity in network and provide a way to ensure dynamic incentive mechanisms. The conception objective is to protect participants' privacy by breaking the link between data reporter identity and the reported sensing data. Most of the existing solutions could protect the privacy with incentives mechanism relying on a trusted third party. Unlike those solutions, our approach particularly does not rely on any trusted third party. In the system, participants collaborate to hide reports' sources in a decentralized model. We still envision the participatory sensing system model composed of the following entities as shown in Figure 1:

1) Task Requester (TR): TR is the participatory sensing task sponsor. It collects the reported data from a given participatory sensing application and will provide incentives for participants.
(2) Service Provider (SP): SP is the intermediary between TR and participants. It allocates the sensing task from TRs to elected participants and transfers the sensing data reports reversely.
(3) Credit Authority (CA): CA is a neutral service. It manages the real identities of other entities. Moreover, the *CA* manages also participants' and TRs' accounts, and performs credit transactions from TRs' account to the participant account who has the micropayments token.
(4) Participant: they are mobile sensing devices (e.g., smart phones) equipped with sensing, computation, and communication abilities controlled by people. The owners are volunteers who participate in the sensing tasks and report sensing data to SP before earning the incentives. We use "participant," "user," or "group member" interchangeably in the remainder of this paper.

3.2. Adversary Model. For security and privacy, this protocol requires that each group of participants include at least two honest participants to make it impossible to distinguish between two data reporters, and these two participants could not collude with others or global eavesdroppers. Besides, all entities could be honest-but-curious in this system, in which each entity may try to learn about other participants passively, but faithfully follows the protocol. In other words, all entities follow protocol specifications but try to get some private information about other participants. This protocol allows global and passive eavesdroppers to exist, who can monitor all the communications in the network. This consideration is realistic considering network providers and government agencies. Furthermore, system entities can collude with each other and even with powerful eavesdroppers trying to get the link between participants to reported data.

For obtaining incentives, participants can try to earn more rewards than that of the sensing report's worth. For example, the participant uses data duplication attack, or uses different identities to report the same sensing data content, even report faked data content. We also assume that participants may try to steal others' certificates, use an invalid certificate, or use a valid certificate to get rewards repeatedly.

3.3. System Assumptions. In this system, all entities have their own private/public key pair, including each participant P_i having (sk_i, pk_i), TR (sk_{TR}, pk_{TR}), SP (sk_{SP}, pk_{SP}), and CA

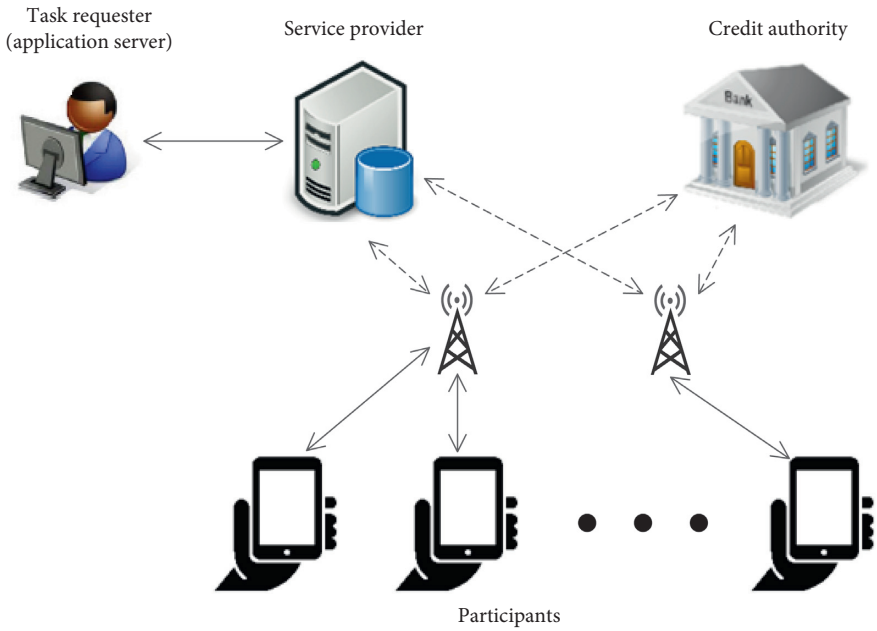

Figure 1: The system model with different entities.

(sk_{CA}, pk_{CA}). And the CA has another private/public key pair (sk^R_{CA}, pk^R_{CA}) for encrypting the tag of incentives. Each entity can generate the key pair itself or apply for a key pair from a certification authority. The members in the same group can communicate with each other in a peer to peer mode, and all communications between different entities are encrypted. Finally, our discussion will be in one group causing the functions of every group to be in the same manner.

4. Protocol Design

The goal of communication anonymity techniques is to confuse the identities and data through multiple participants submitting data together. Most of them use the idea of Chaum's Mix network and DC-nets. Mix networks are mainly designed to work in a centralized fashion with the help of fixed servers; DC-net hides the real data covered by negotiated masks. Nevertheless, this idea has been extended in bulk transfer in Dissent [10], which makes it feasible in a distributed way. The anonymous data reporting protocol in this paper (called OADR scheme for short in the remainder of the article) also leverages the DC-net and bulk transfer followed by works [8–14]. In this paper, we firstly modify bulk transfer protocol. Then, all the processes of OADR scheme shows in Section Protocol Description. The related notations used in the remainder of the article are summarized in Table 1.

4.1. Modified Bulk Transfer. The bulk transfer protocol aims to submit a random arrangement of N participants' data in a vector of N slots. We assume that participants $P_1, P_2, \ldots, P_N$ initially hold an amount of messages $\text{msg}_1, \text{msg}_2, \ldots, \text{msg}_N$ willing to report. Each participant P_i knows its own slot position in the vector but does not know the other participants' slots. The lengths of the whole vector and each slot are published. Each pair of participants (P_i and P_j) shares a common secret seed S_{ij}, and note that $S_{ij} = S_{ji}$.

First of all, P_i generates a bitstream C_i^{msg} with the length of the whole vector (let beL_v). C_i^{msg} could be split into N slots which are filled in P_i's slot with msg_i, and the bits in the other slot are all zero (the placeholder #), i.e.,

$$C_i^{\text{msg}} = \overbrace{00\cdots0}^{L_b} \,|\text{msg}_i|\underbrace{00\cdots0}_{L_v}, \tag{1}$$

where | denotes concatenation and L_b is the demanded bits of the other slots ahead of P_i's. Then, P_i continues generating $N-1$ pseudorandom bitstreams C_{ij}, all with the length L_v and based on the seed shared with other participants as

$$C_{ij} = \text{PRF}\{L_v, S_{ij}\}, \quad j \neq i, \tag{2}$$

, where the j is in [1, N] corresponding to all participants beside itself. The PRF is the signification bits of length L_v generated by a pseudorandom function taking S_{ij} as seed.

After that, P_i generates vector C_i using C_i^{msg} XOR all C_{ij}, i.e.,

$$C_i = C_i^{msg} \oplus C_{i1} \oplus C_{i2} \oplus \cdots \oplus C_{i(i-1)} \oplus C_{i(i+1)} \oplus \cdots \oplus C_{iN}. \tag{3}$$

The bitstreamsC_i for $i \in [1, N]$ are sent to the service provider, where all of them are XORed.

$$C = C_1 \oplus C_2 \oplus \cdots \oplus C_N = \text{msg}_{\pi N(1)}|\text{msg}_{\pi N(2)}|\cdots|\text{msg}_{\pi N(N)}. \tag{4}$$

Finally, the service provider obtains a concatenation of all data in the vector C (see Figure 2), where $\text{msg}_{\pi N(i)}$ is the random permutation in $\text{msg}_1, \text{msg}_2, \ldots, \text{msg}_N$. The service provider has been sent all the messages from participants. The

Table 1: Summary of notations.

Notation	Description
P_i	The i^{th} participant in order (member, user)
S_{ij}	The secret seed between the i^{th} participant and the j^{th} participant
SRM_i	The slot reservation message for the i^{th} participant
$\vert$	Concatenation operator
#	The placeholder
PN_i	Random secret pseudonym of the i^{th} participant
$\pi_{i-1}(j)$	The j^{th} element in order after the random permutation of $i-1$ elements
n_{R_j}	The nonce for the j^{th} slot, earlier agreed on its creation rule
$\mathrm{PRF}\{L, x\}$	Pseudorandom generation function based on seed x, the length of the generated pseudorandom is L

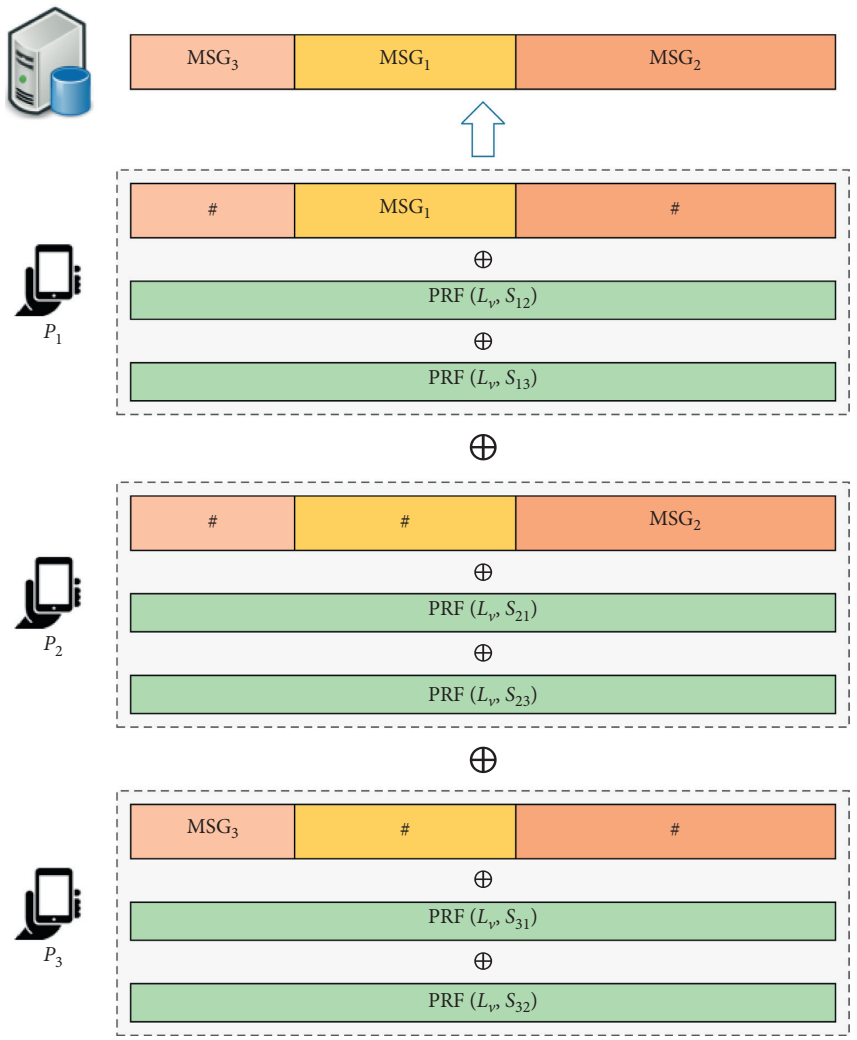

Figure 2: Illustration of modified bulk transfer for group of 3 participants using permutation [1–3].

link between the message and its own is only known by the owner if the order of the permutation is shuffled. Comparing with the original bulk transfer protocol, this reduces the time of pseudorandom generating maintaining the same anonymity.

4.2. Protocol Description. This protocol is mainly based on work by Yao et al. [14] and our previous work [8] (called Yao's scheme and ADR scheme, respectively); the differences among Yao's ADR and OADR scheme are that: in Yao's scheme, each participant receives and sends N slot reservation messages with N times encryption and decryption. In the ADR scheme, each participant gets only its predecessors' slot reservation messages and encrypts its own slot reservation messages with successors' public keys, and decrypts once for each participant during the whole slot reservation stage. For OADR scheme, it uses slots random selection in bulk transfer instead of the encrypted

pseudonyms permutation to reserve the slots, thereby reducing the latency of the slot reservation stage, and for the data submission, the process is simplified due to the modification of bulk transfer. Furthermore, the incentive mechanism supports dynamic rewards and enhances anonymity based on blind signature scheme and bulk transfer. The OADR scheme is formalized into five main sequential stages as ADR scheme including: setup stage, slot reservation stage, token requesting stage, data submission stage, and token deposit stage. Except for the setup stage, all the stages leverage the modified bulk transfer mentioned (see the Section "modified bulk transfer") to submit data.

4.2.1. Setup Stage. For the new participants, they need to register to the CA and the SP using their real identities. Considering the sensing data reports may involve participants' location or the task may request data from certain locations, and it is best to select participants in the same group in the same geographic area. There are many researches to prevent the spatial information fromthreatening user privacy. For example, work by [53] shows a way to divide the space into subregions, and the user's location is hidden in l possible sub-regions. Without loss of generality, the N group members are denoted as $P_1, P_2, \ldots, P_N$.

At last, the group members exchange their public keys pk_i, such as using a digital certificate. Using the public key, each participant P_i shares a secret seed and a random function with every other participant P_j for generating the secret S_{ij} between P_i and P_j, which makes S_{ij} the same as S_{ji}, which is only known between them and could generate pseudorandoms for multiple rounds. Furthermore, the format and length of the slot reservation message, and the number of redundancy slots used in the slot reservation stage should be agreed upon by all the participants and SP.

4.2.2. Slot Reservation Stage. The goal of this protocol is to submit the reported data from a whole group of members together, in which the permutation of the different participants' data is disordered to break the link between the reporter and corresponding data. In other words, the participant P_i just knows its own position of the data but the others' position is ambiguous to P_i. In the slot reservation stage, the position of each participant will be determined. Unlike Yao's scheme and ADR scheme using random permutation of encrypted pseudonym to determine the submission order for ensuring anonymity, the slot reservation stage in OADR is optimized which leverages the idea of DC-net and bulk transfer for selecting the slot in redundancy slots. The whole process of the slot reservation stage for a group of three members is illustrated in Figure 3.

At the beginning of this stage, each participant P_i needs to generate a λ-bits fresh random secret pseudonym denoted as PN_i. Obviously, the pseudonym is only known by the participant itself. As the pseudonym is randomly generated, the collisions among all group members are possible. The collisions' probability reduces with the growing of λ and the decreasing of the number of participants in one group. Nevertheless, OADR no longer needs to care about whether the SRM itself will conflict, it only needs to focus on whether the slot conflicts.

Once the pseudonym generation was completed, each participant P_i prepares a slot reservation message (SRM). In addition, the format and length of the slot reservation messages should be decided early by all participants and SP in order to produce the unique format and length SRM. Otherwise, the vector of SRM will not be able to be recognized or identified. The basic SRM is shown in the following equation:

$$\mathrm{SRM}_i = \langle \mathrm{PN}_i | L_i \rangle, \tag{5}$$

where L_i is the length of data with the length of an encrypted token used later for incentives (blinded token) which P_i will report to the task requester (TR) anonymously. Obviously, the data length (L_i) should be the length of the encrypted data with $\mathrm{pk}_{\mathrm{TR}}$ if the data content need to be kept confidential from SP.

The next step is that each participant P_i transfers its SRM_i to SP by bulk transfer. Each participant P_i uses each secret seed S_{ij} to generate $N-1$ pseudorandom bit stream vectors C_{ij} of length $M \cdot L_{\mathrm{SRM}}$, as shown in the following equation:

$$C_{ij} = \mathrm{PRF}\left\{M \cdot L_{\mathrm{SRM}}, S_{ij}\right\}, \quad i \neq j, \tag{6}$$

where PRF represents a pseudorandom vector of length $M \cdot L_{\mathrm{SRM}}$ generated by a pseudorandom function with S_{ij} as a secret seed. L_{SRM} is the length of each SRM. M is the total number of slots, which is greater than the total number of participants N. Note that M should be negotiated and decided by all participants and SP when deciding the format and length of SRM. Therefore, the vector can be divided into M slots of equal length, enough to accommodate all participants' slot reservation information, and some unused slots will remain. Then, each participant P_i selects a random slot (let the position be pos_i) to hold its SRM and generatesC^{poi_i}, as shown in equation (7):

$$C^{\mathrm{pos}_i} = \#_1 | \#_2 | \cdots | \#_{pos_i - 1} | \mathrm{SRM}_i | \#_{\mathrm{pos}_i + 1} | \#_M, \tag{7}$$

where $\#_1$ represents the placeholder of the i^{th} slot (that is, all zeros), $i \in [1, N]$ and $i \neq \mathrm{pos}_i$. Finally, P_i XOR all C_{ij} and C^{poi_i} to complete the generation of vector C_i:

$$C_i = C^{\mathrm{pos}_i} \oplus C_{i1} \oplus C_{i2} \oplus \cdots \oplus C_{i(i-1)} \oplus C_{i(i+1)} \oplus \cdots \oplus C_{iN}. \tag{8}$$

After that, each participant P_i sends C_i to the service provider (SP) by using bulk transfer protocol. Finally, the SP

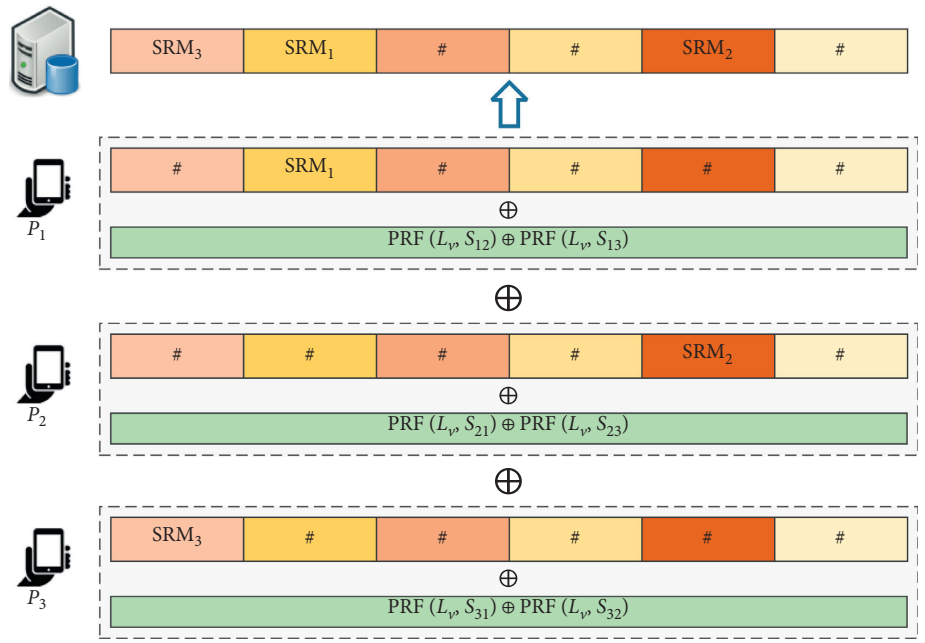

Figure 3: Illustration of a slot reservation by six slots for a group of three participants.

obtains the slot reservation vector C by performing XOR operation on all received C_i, as equation (9) shows:

$$C = C_1 \oplus C_2 \oplus \cdots \oplus C_N = \underbrace{\#|\cdots|\mathrm{SRM}_{\pi N(1)}|\cdots|\#|\cdots|\mathrm{SRM}_{\pi N(N)}|\cdots|\#}_{M}, \tag{9}$$

where $\mathrm{SRM}_{\pi N(i)}$ is the i^{th} element in the random permutation of $\{\mathrm{SRM}_1, \mathrm{SRM}_2, \ldots, \mathrm{SRM}_N\}$. SP gets the vector C and checks the number of SRMs in the vector to confirm whether there is a slot collision (i.e., whether multiple participants have selected the same slot), and then publishes C. Each participant checks whether the SRM of its selected slot in the published vector C is correct. When a time slot collision is found, the slot reservation stage needs to rerun again from the step of pseudorandom generation (starting from equation (6)). Each participant generates a new round of pseudorandom and finally builds a new vector C. For the slot of the participant P_j that does not collide (i.e., P_j checks whether its SRM_j in the vector C is correct), P_j places its SRM_j in the same time slot as before; on the contrary, the slot that has collided participant P_k needs to select another unused slot to carry its SRM_k. If all participants do not find their own SRM errors but the total number of SRMs is still insufficient, all participants must reselect their own slots. Then, all participants sends the newly generated vector C_i to SP to check again. The process of SP checking time slot collision will repeat until all participants confirm that their SRMs are correct, and the stage is successfully completed. After that, each participant P_i determines its own reserved slot, which is the position of its SRM_i, without a placeholder. P_i calculates the data length declared by all SRMs to confirm the starting position of its reserved slot in the data submission stage and the total length of the data vector. Note that when the participant P_i calculates its time slot, it needs to ignore the placeholder and only calculate the length declared in all SRMs.

In addition, the slot collision is incidental when the redundancy slots are not many. The probability of collision is shown in the following equation:

$$P_{M(N)} = 1 - \frac{M}{M} \cdot \frac{M-1}{M} \cdot \cdots \cdot \frac{M-(N-1)}{M} = 1 - \prod_{k=1}^{N-1}\left(1 - \frac{k}{M}\right). \tag{10}$$

Since each redundancy slot brings extra XOR operations, the increase of redundancy slots will reduce the probability of collision as well as increase the operation time dramatically . In the case of 800 slots ($M = 800$) for 40 members ($N = 40$), the probability of collision is 0.63. But with 100 slots for 5 members, the probability of collision is less than 0.1. Therefore, it is suggested to divide participants to multiple groups instead of one group for many participants in order to keep a low probability of collision for fewer slots. On the other hand, the collision is inevitable for a large number of participants. But, fortunately, the probability of collision is usually low in the second round of slot

reservation. According to the experiments, the average probability of collision for the second round is less than 0.14 even for $M = 200$ and $N = 40$.

4.2.3. Token Requesting Stage. Once the slot reservation stage is completed, participants request micropayment tokens from the Credit Authority (CA). Firstly, participants generate N secret seeds (denoted as PI_i) and encrypt the pseudonyms (let be $\mathrm{PI}_i^{\mathrm{pk_{CA}}}$). In this stage, participants' identities and tokens need to be unlinked; thus, all participants send $\mathrm{PI}_i^{\mathrm{pk_{CA}}}$ to CA together using the similar approach as the slot reservation stage to transfer the tokens. At the same time, one of the participants P_k sends the vector with the task id, session number, and round number to CA. CA confirms the session number and the round number with the SP. This slot reservation for tokens request still needs to protect from collision. Therefore, CA publishes the received vector containing encrypted pseudonyms for confirmation by participants as the slot reservation stage does, until all participants confirm the submissions. At last, CA removes all the placeholders and publishes the vector for determining the order token transfer.

After each participant ensures its own slot, CA generates N fresh random integers of δ-bits called incentive tokens, denoted as τ. The token is generated based on the hash function by the seed including random secrets, task identity, session number, round number, participant identities, and the current time, in order to make sure that the token is unique and fresh for each request. Next, CA decrypts all secret seeds $\mathrm{PI}_i^{\mathrm{pk_{CA}}}$ and uses them to generate N pseudorandoms $\mathrm{PRF}\{L_\tau, \mathrm{PI}_i\}$. After that, CA xor τ and $\mathrm{PRF}\{L_\tau, \mathrm{PI}_i\}$ are put it into the corresponding slot to build the final token vector V_T, as shown in equation (11).

$$V_T = \langle \mathrm{PRF}\{L_\tau, PI_{\pi N(1)}\} \oplus \tau_{\pi N(1)} \rangle | \langle \mathrm{PRF}\{L_\tau, PI_{\pi N(2)}\} \oplus \tau_{\pi N(2)} \rangle | \cdots | \langle \mathrm{PRF}\{L_\tau, PI_{\pi N(N)}\} \oplus \tau_{\pi N(N)} \rangle. \tag{11}$$

Finally, CA publishes the vector V_T, and the tokens are fetched back by each participant from its slot. The participant P_i uses PI_i to generate the pseudorandom and then XOR the pseudorandom with the fetched tokens $\mathrm{PRF}\{L_\tau, \mathrm{PI}_i\} \oplus \tau_i$ to obtain real token τ_i.

Next, the participant P_i collects a random secret integer r_i which needs to satisfy equation (12):

$$\begin{cases} 1 < r_i \le 2^\delta - 1, \\ \gcd\left(2^\delta, r_i\right) = 1. \end{cases} \tag{12}$$

After that, P_i generates a series of blind tokens using the secret factor r_i and different TR's incentive public keys based on the RSA blind signature scheme without loss of generality:

$$\tau_i^* = \tau_i \cdot (r_i)^{\mathrm{pk_{TR}}} \mathrm{mod} n_{\mathrm{TR}}. \tag{13}$$

For simplicity, τ_i^* is called the blind incentive token of the participant for the rest paper.

4.2.4. Data Submission Stage. After the slot reservation stage, each participant P_i has a slot position, $\mathrm{pos}_i \in [1, N]$. And P_i holds the sensing data and blind token $m_i | \tau_i^*$ (mt_i for short) of length L_i willing to report. Due to the order of the slots and the lengths of the data being determined, all participants could use bulk transfer protocol to report the data to SP directly. Finally, SP obtains the vector C:

$$C = mt_{\pi N(1)} | mt_{\pi N(2)} | \cdots | mt_{\pi N(N)}. \tag{14}$$

SP publishes the received data C for each participant and checks its own data including sensing data and the blinded token. Next, each participant reports a flag bit to the SP to confirm the data using the same data submission protocol. The flag could be a decided bit like the flag is set to 1 when data content is correct; otherwise, the flag bit is zero. The SP receives the confirmed report vector and checks whether there is any error. If all the data are correct, SP sends the data vector C to the task requester (TR). On the contrary, if there is one or more zero in the flags vector, the SP needs to execute the error recovering step for repairing the data.

The error recovering step requires all participants to repeatedly submit the reported data, and the pseudorandoms used in the process need to be regenerated. Considering about malicious members, some blame mechanisms should be run if the error is detected multiple times, like work by [10]. In addition, selective ordering can be used to detect malicious members.

4.2.5. Token Deposit Stage. After the data submission stage, the TR receives the final message vector C from the SP. For the data mt_i in each slot, TR decrypts the sensing data $m_{\pi N(1)}$, then checks the freshness and content of the data with the task agreed terms. According to the quality of the data, TR decides the amount of the incentive for the sensing data m_i (run some data evaluation mechanisms to estimate the data). Then, it collects a random secret integer for representing the amount where the process is similar to r_i collection, denoted as R_i. *TR* signs the incentive token (τ_i^*) with the private key of the corresponding level $\mathrm{sk_{TR}}$ as shown in the following equation:

$$\begin{aligned} \mathrm{sign}\left(\tau_i^*\right) &= \left(R_i^{\mathrm{pk_{CA}}} \cdot \tau_i^*\right)^{\mathrm{sk_{TR}}} \mathrm{mod}\, n_{\mathrm{TR}} \\ &= \left[R_i^{\mathrm{pk_{CA}}} \cdot \tau_i \cdot (r_i)^{\mathrm{pk_{DC}}}\right]^{\mathrm{sk_{TR}}} \mathrm{mod}\, n_{\mathrm{TR}} \\ \mathrm{sign}\left(\tau_i^*\right) &= \left(R_i^{\mathrm{pk_{CA}}} \cdot \tau_i\right)^{\mathrm{sk_{TR}}} \cdot r_i \mathrm{mod}\, n_{\mathrm{TR}}. \end{aligned} \tag{15}$$

Meanwhile, TR encrypts R_i by CA's incentive publish key (denoted as $R_i^{\mathrm{pk_{CA}}}$) and concentrates it with $\mathrm{sign}(\tau_i^*)$. Next, it

concentrates all $\text{sign}(\tau_i^*)|R_i^{\text{pk}_{\text{CA}}^R}$ to be a vector according to the order of slot reservation. The vector is sent back to SP for publishing. If the vector needs to remain confidential from other entities, TR could use a part of the data to mask each slot of the feedback vector as CA sends tokens to participants in token request stage.

Each participant P_i fetches its own signature tokens from the published vector. P_i will unblind signature incentive token as follows:

$$\begin{aligned}\text{sign}(\tau_i) &= \text{sign}(\tau_i^*) \cdot r_i^{-1} \bmod n_{\text{TR}} \\ &= \left(R_i^{\text{pk}_{\text{CA}}} \cdot \tau_i\right)^{\text{sk}_{\text{TR}}} \cdot r_i \cdot r_i^{-1} \bmod n_{\text{TR}} \\ \text{sign}(\tau_i) &= \left(R_i^{\text{pk}_{\text{CA}}} \cdot \tau_i\right)^{\text{sk}_{\text{TR}}} \bmod n_{\text{TR}}.\end{aligned} \tag{16}$$

Finally, all the participants ally to request micropayment from CA by the incentive deposit message, which uses the same order in token requesting stage to generate the vector that each slot contains encrypted message $\tau_i|\text{sign}(\tau_i)|R_i^{\text{pk}_{\text{CA}}^R}|\text{account}_i$ (one of participants sends the vector to CA with task id, session number, and round number). The CA decrypts and removes R_i from $\text{sign}(\tau_i)$ as shown in equation (16), checks again the token freshness with round and session number, signature validity, and whether the token τ_i belongs to the corresponding slot. Then, CA performs a micropayment transaction from TR's account to the corresponding member account after confirming the specific incentive represented by different R_i. The participant's account (account_i) should not be associated with the participant's real identity. Furthermore, it is suggested that the participant changes the account when the members change, or the CA uses an untraceable digital currency payment (such as "Monero") to preventthe participant from being tracked when using the incentive. But this is not the focus of this paper.

In addition, the protocol execution may include multiple sessions where each session may include multiple rounds (Figure 4). A round represents the data submission stage using the same transmission plan decided in the previous slot reservation. In contrast, a new session requires execution of the slot reservation stage. The token requesting and deposit stage is independent of the data submission stage; the order of token transmission can also be performed in multiple rounds and sessions using the same principle. Furthermore, the setup stage only needs to run once for a group if the members do not change.

5. Protocol Analysis

This protocol is used to ensure the anonymity of data reporting with incentive and reputation mechanism. In this section, the analyses of the protocol will be discussed, including anonymity, integrity, and efficiency.

5.1. Anonymity. The protocol preserves anonymity in a group containing at least two honest participants, where other malicious members colluding with system entities and global eavesdroppers cannot determine which honest members submitted which sensing data with nonnegligible probability.

Firstly, we need to analyze the unobservability of OADR's bulk transfer. In the bulk transfer of OADR, the real data are hidden under the pseudorandom number generated by S_{ij} as the secret seed. If a malicious attacker wants to get the real data, it must know the pseudorandom number. Part of the pseudorandom number is mixed with the real data (by XOR operation), and only P_i and P_j know S_{ij} and the corresponding pseudorandom generation function. When the pseudorandom number generation function and the secret seed are uncertain, it is very difficult to obtain a complete pseudorandom, which is guaranteed by the security of the pseudorandom generation function. Therefore, the bulk transfer protocol of OADR is unobservable.

Next, we analyze the anonymity in each stage. In the setup stage, the participant's real identity is used to prevent the same user from faking to be multiple participants for earning the incentive. But the real identity will not appear with the data submission, so it does not expose the sensing data anonymity. On the other hand, the key pair of each participant is binding to its real identity, which the certification authority can discover. However, the key pair is only used for exchanging secrets. The anonymity in data submission does not rely on public keys but relies on the bulk transfer. Even some entities can discover the identity of participants, but still cannot link their submitted data. In addition, the certification authority could verify the identity to reduce the risks of impersonated multiple members. In slot reservation and data submission stages, the modified bulk protocol still ensures unobservability, which is stronger than unlinkability and anonymity. Before proofs, we repeat the definition of semantically secure cryptosystem, which has been stated in the previous work.

5.1.1. Definition. A cryptosystem is semantically secure if any probabilistic, polynomial-time algorithm that is given the ciphertext of a certain message m (taken from any distribution of messages), and the message's length, cannot determine any partial information on the message with probability nonnegligibly higher than all other probabilistic, polynomial-time algorithms that only have access to the message length (and not the ciphertext) [54].

In other words, for two encrypted equal-length messages m_0, m_1, an adversary knowing m_0, m_1 cannot determine which is the cipher corresponding to m_0 and which one corresponds to m_1 with a probability significantly greater than 0.5.

The random slot reservation selected by the participant ensures that the position is only known to the participant. Because it is selected at random, the probability of the participant selecting any slot is $1/M$ (M is the number of slots). Therefore, if a malicious adversary wants to know the position chosen by the participant, it needs to obtain the position of the real data sent by it. Suppose an attacker controls all members except the two members whose submitted data are to be anonymized, at least one seed between the two honest members should be unknown. The SRM of an honest member must be encrypted with a one-time

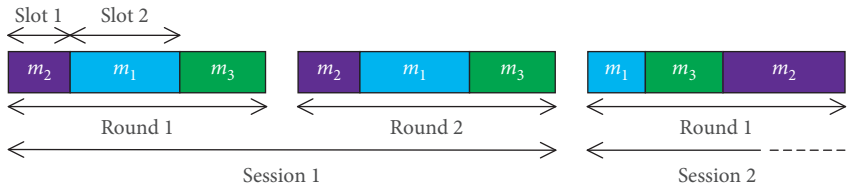

Figure 4: Example of transmission schedule for three anonymous group members.

pseudorandom generated by a seed only known to the honest member. In this case, the unobservability of the participant's SRM is guaranteed through XOR of the pseudorandom. If the malicious attacker wants to know the slot selected by the participant, it must know the pseudorandom generated by the honest participants. As analyzed above, the attacker cannot obtain a complete pseudorandom, when the pseudorandom number generation function and the secret seed are unknown. Similarly, the secret seed cannot be calculated. The attacker cannot intercept messages sent by honest participants, and tries to remove the pseudorandom to determine which slot the participant puts its SRM into. Even if the attacker compares the bit streams of two honest participants, it can only find that the two slots are different, but still cannot determine which participant selected which slot. If it is a slot collision that causes the re-run time slot reservation, the pseudorandom will be regenerated, and the attacker still cannot cancel the pseudorandom bit by comparing the pseudorandom in the two slot reservation stages. The only way to get the message content is to get all the ciphertexts generated by all members. Therefore, the attacker only knows which slots these honest participants are using but cannot determine which slot corresponds to which honest participant and the probability is greater than $1/K$ (K is the number of honest participants). In the end, the attacker cannot learn anything about the arrangement from the message. Since slot reservation ensures the anonymity of the arrangement of honest participants, an attacker cannot link participants and their reported data during the data submission stage.

In the token request stage, the anonymity of the participant is not exposed due to unknown relations between the token and the participant going by bulk transfer protocol. Furthermore, work in Lucre [55] for anonymous tokens can be adapted to overcome some attacks, such as private key extraction attacks. When the participant retrieves the token, the unobservability of the token is guaranteed by XOR of the token with a pseudorandom. The secret seed and pseudorandom generation function are only known to the participants in the corresponding slot and CA. Therefore, a malicious attacker cannot learn the content of the token through observation, and thus cannot pretend to be the participant by the token. The participants themselves can easily calculate the corresponding pseudorandom and obtain real tokens. Furthermore, the methods of keeping the confidentiality of the token are various, pseudorandom generation and XOR operation are more efficient.

In the last stage, the anonymity is mostly derived from the blind signature scheme. Here, the blind signature mechanism is used twice, once to verify the legitimacy of the token and then to carry specific incentives. Blind signature gives a signer the ability to sign a document without knowing the contents. Moreover, the signer cannot determine when or for whom he signed a given pair of signatures, even though the signer can verify whether the signature is valid. Blind signature is mostly used to preserve the anonymity of the signature requester. On the other hand, TR and CA possibly collude with each other. A possible attack is to sign different tokens or use dynamic incentives for the purpose of de-anonymizing members' contributions. But, the final beneficiary account or digital cash is irrelative to the user's identity, which is derived from bulk transfer and one-time account. Therefore, TR and CA with other malicious members just link the data to the account and not the user's identity. Even SP is malicious to select a certain honest member to participate in multiple sensing tasks with different other members; however, no entity can obtain the relation between the member and the data if the member changes its account.

This protocol preserves the anonymity of K honest participants in each group. It establishes an out-of-order submission schedule by using random slot selection; then, it submits the out-of-order encoded data by all the participants in a group, so that the relationship between the data and the identity of the data contributor is cut off, and the identity of each participant is hidden among the K honest participant. Therefore, any malicious entity cannot distinguish any two honest participants with a nonnegligibly high probability, i.e., it satisfies the semantic security of the cryptosystem. This anonymity is irrelevant to the number of honest participants. Even if there are only two honest participants, a malicious entity still cannot determine who sent which data. However, more honest participants can significantly reduce the probability of a malicious entity identifying the participant.

In addition, if all honest participants have submitted the same data, the SP or TR can determine the content of the data and the identities of all honest participants. Therefore, this protocol is not suitable for applications where the diversity of the reporting data is too low. However, this situation is not common for participatory sensing applications, especially when data need to be collected continuously. Furthermore, if the diversity of the data is low, a data diversity check should be conducted before submitting the data. First send a message digest of the data that will be submitted; the SP will publish all the message digests and all participants will check them, and then consider whether to submit the data. However, adding the diversity check will lead to an extra latency of sending data each time. Or, add privacy protection for data publishing, such as local

differential privacy protection. In this case, this protocol may not be able to submit the original individual data.

5.2. Integrity. The protocol is based on an honest but curious adversary model. Therefore, all system entities and participants must follow the protocol scheme but can passively try to destroy the anonymity of participants. If all members run the protocol scheme exactly, in the first stage, each participant has a secret seed with and the public key of each other participant, and also public keys of TR, SP, and CA.

In the slot reservation stage, each participant selects a slot randomly to reserve the slot for data submission. They update the slot reservation message encrypted with a series of one-time pseudorandoms generated from the seed sharing with other members, and then, the XOR operation will cancel all pseudorandom bits and the slot reservation message is anonymously recoverable. The process is the same as performing in the data submission stage. In the token request stage, the CA generates random nonspecial tokens and keeps records in a table associated with the round number and the session number. Furthermore, CA links the slot to the token but cannot know who obtains which token. In the last stage, the TR blindly signs members' tokens; the unblind signature can be easily calculated by the token owner but is difficult for other members. We reversely use the blind signature mechanism to carry a specific amount of incentives, which makes it difficult to forge. Because it cannot be observed, even if participants collude, it is difficult to disrupt the operation of the system by exchanging encrypted incentive tags. Finally, each member gets its incentive just by checking the availability of the token and the correctness of the slot. In addition, the token is kept secret and since it is randomly generated, it is hard to guess [40]. On the other hand, the slot reservation stage and the data submission stage use similar strategies. The former is to establish a verifiable transmission plan to facilitate subsequent anonymous data transmission. If all participants submit the data of the same length only once, it is possible to use the slot reservation stage strategy for data transmission. For larger data or data that are submitted multiple rounds, determining the transmission plan can effectively reduce the extra overhead caused by the submission of redundant data and the collision of slots.

If the system entity or participant is dishonest, that is, it can perform without following the requirements of the protocol, the system is vulnerable to attack or even unable to operate, but it is easy to detect many malicious behaviours, e.g., the protocol can detect double use of any token, even the if the participant uses tokens of others or puts its token to another slot, because these operations lead the token verification to fail by the principal of blind signature mechanism. On the other hand, the bitstream in bulk transfer is possible to be forged or members can put a malicious message in an inadequate slot. These malicious behaviours are still easily detectable. Selective order could remove a part of members including the malicious ones. Similarly, misbehaviour of TR or SP can be lead to participants' leaving. Furthermore, participants may accidentally submit wrong data, causing TR or SP to be unable to obtain the information submitted by all participants. This situation is the same as members put a malicious message in an inadequate slot. These errors will be detected during the error recovery stage after all data have been submitted. This type of problem can usually be solved by resubmitting the data. If there is indeed a malicious member, the blame mechanisms in work [10] or selective order can be used to exclude the malicious participant.

In addition, members leaving is a serious problem in all distributed anonymity preserving schemes, whether it is a participant who accidentally exits the system or goes offline halfway. Generally, a member who leaves for a long time requires the restart of the protocol; the maximum leave time must be estimated early and set as a parameter depending on the application (participatory sensing application). The whole stage will be halted if a member leaves before completing its job; depending on the participatory sensing application, the group decides whether to wait for the member or to restart the protocol without considering it. All the stages use the bulk transfer except the setup stage; a member leaving means that the whole group submitted data is incomplete. In that case, the SP depending on participatory sensing application decides to wait for that user data or to restart the protocol.

5.3. Efficiency. In the setup stage, each participant needs to generate N–1 secret seeds to share with each other member, which is the same as work [8].

In the slot reservation stage, each participant needs to generate N–1 pseudorandom bits streams. And then, this protocol performs N^2 XOR operations instead of previous $(N \cdot (N-1))/2$encryptions and $(N \cdot (N-2))/N$ decryptions to permute the slot randomly. However, the traffic overhead is also increasing $(M-1) \cdot L \cdot N$ from $N^2 \cdot L$, where M is the number of redundancy slots and the L is the SRM predefined length (all SRMs must have the same length). Even if the traffic overhead increases, the efficiency improvement brought about by replacing encryption with fewer XOR operations is huge. And, in reality, each participant could transfer the message at the same time, leading to lower transfer latency unlike when each participant needs to wait for the previous participant's message to decrypt and permute in previous work. It is similar to the data submission stage; the number of pseudorandom bits stream generations is reduced from $N \cdot (N-1)$ to $N-1$ for one participant, but the number of bits XOR is the same even if the times of XOR operations reduce to N. Once received by SP, it needs to compute N times the XOR operations for $\sum_{i=1}^{N} L_i$ bits.

In the token request stage, the CA needs to execute a random token generation function N times,and participants run a process similar to the slot reservation stage to obtain the tokens anonymously. On the other side, each participant generates one secret random number, and runs a blinding operation (equivalent to an encryption). In the last stage, the TR and the CA needs to encrypt and decrypt each incentive tag, blind and unblind sign all the tokens, and verify them. Encryption and decryption theoretically consume a lot of

resources, but it is necessary for privacy considerations. And because the fields that need to be encrypted and decrypted are short, the consumption in CA and TR is actually limited compared to the data transmission process; the subsequent experiments have also proved it.

6. Protocol Implementation

In order to prove the feasibility and efficiency of our protocol, we built and tested a prototype of the protocol. In this section, the implementation overview and the evaluation results will be illustrated and discussed.

6.1. Implementation Overview. The prototype uses java for both Android mobiles and servers. The involved cryptographic primitives are implemented using bouncy castles (sponge castles for Android). For the public key cryptosystem, all the server entities opt RSA-OAEP with a key length of 1024-bit and the participants use different keys.

For experiments, we ran the server on a PC (Intel XEON E3-1230, 3.3 GHz, 16 GB RAM, and windows 10) and an android phone (Google nexus 3). The network is 100 m WiFi channel for ensuring communication between the mobile phone and the server. The server runs three independent entities: SP, TR, and CA. For the N group members, first we run the setup stage to generate the full system parameters, and then we run N members on the same mobile phone where the execution is running in an independent fashion; therefore, we take only significant execution time. The communication between N members is simulated via 100 m WiFi channel where the server acts as a message repeater to send messages back to the phone.

The energy consumption of the phone battery is measured using an app named "iTest," which allows the extraction of energy consumption for each app on android 4.3. And, we use a formatted phone to reduce the error due to other app or system consumption. We also eliminate our own app's basic consumption (the energy consumption when the app is running without doing anything), which consumes a mean of 1.71193×10^{-2} J per second.

6.2. Performance Evaluation. The evaluation results of all stages are illustrated and discussed in this section. It is important to mention that privacy-related researches still lack metrics, which allow the comparison of different approaches. We can only compare similar approaches as comparing communication anonymity in our case where we can only compare protocols' latencies and energy consumptions.

6.3. Setup Stage. First, the setup stage is evaluated. Each member generates $N-1$ secret seeds of 32 bytes length. A secure pseudo random generation function based on SH1PRNG uses about 3.3×10^5 ns, and it uses about 25 ms to generate secrets for all the members of group size $N = 40$.

6.4. Slot Reservation Stage. Firstly, we evaluate the slot reservation stage latency of group size $N = 40$ for different numbers of slots, as shown in Figure 5. Because slot collisions may occur, the slot reservation possibly needs to repeat. For a certain number of slots, the bars illustrate the average time of submitting the slot reservation message once and finishing the slot reservation stage (i.e., all members hold a slot without any collision), respectively. Generally, to use more slots, it is necessary to not only bring a lower probability of collision, which reduces the risk to repeat to submit slot reservation message, but also increase the latency of pseudorandom generation and XOR operation. Therefore, the time of signal submission is rising with the growth of the number of slots, and the completion times are irregular. The ratio of completed time to single time shows the average counts of repeat. As expected, the average counts are 3 and 1.6 times for group size $M = 80$ and $M = 1600$, respectively. Nevertheless, the shortest completion time is 0.758 s when M = 200.Even though the single submission time is 0.361 s, which is longer than M = 80, the completion time is 0.0581 s faster.

As shown in Figure 6, the latency of the full slot reservation stage for different group sizes. For the OADR scheme, the number of slots is 5 times the group size (e.g., using 25 and 200 slots for 5 and 40 members, respectively). As shown, the increasing efficiency of the OADR scheme is dramatic. Even for a group size of 40, the average latency of the OADR scheme is less than 1 s (0.796 s). In the same condition, the latencies of Yao's and ADR scheme are 107.7 s and 26.8 s, respectively. It is worth noting that the error ranges of OADR scheme latency are relatively large. Among them, the longest experimental time is 1.152 s and the shortest is only 0.64 s in the case of 40 members. The reason is also obvious, the slot reservation stage needs to be rerun once the slot collision occurs in OADR scheme; this will not happen in other schemes. Also, the error ranges of other schemes' latencies depend only on the operations, while that of the OADR scheme is multiplied by the probability of slots collision occurring.

Next, we illustrate the traffic data overhead of slot reservation while the SRM length is 512 bits (64 bytes). As shown in Figure 7, for an SRM of 64-byte length, the traffic overheads are 435.156 KB and 165.761 KB for a group of 40 members in the Yao's and ADR schemes, respectively, while the traffic overhead of the same group under the OADR scheme increases to 509.4 KB. Due to the use of redundant slots, the proposed optimization increases the traffic overhead. But, the encryption operation is not used, the traffic overhead does not increase too much, and the runtime of the slot reservation is greatly reduced.

To measure the mean energy consumption of the one slot reservation stage and compare it to the other two slot reservation schemes, we run all schemes in similar conditions many times (10 times for each group size), and measure the total consumption. Since the app is the same, we do not have to substrate the base energy use. The mean energy consumption of three schemes for each participant for different group sizes is shown in Figure 8. Obviously, OADR scheme's consumption is largely lower than others to the same group size proportionally; As latencies, the error

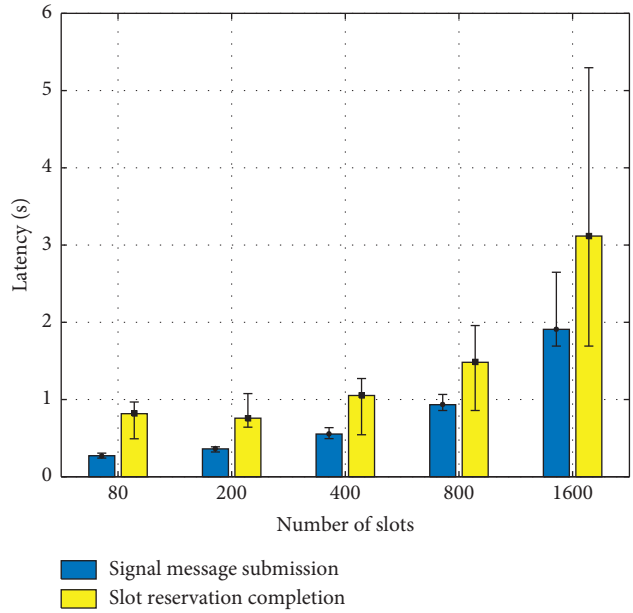

Figure 5: Slot reservation stage latency of group size $N = 40$ for different numbers of slots.

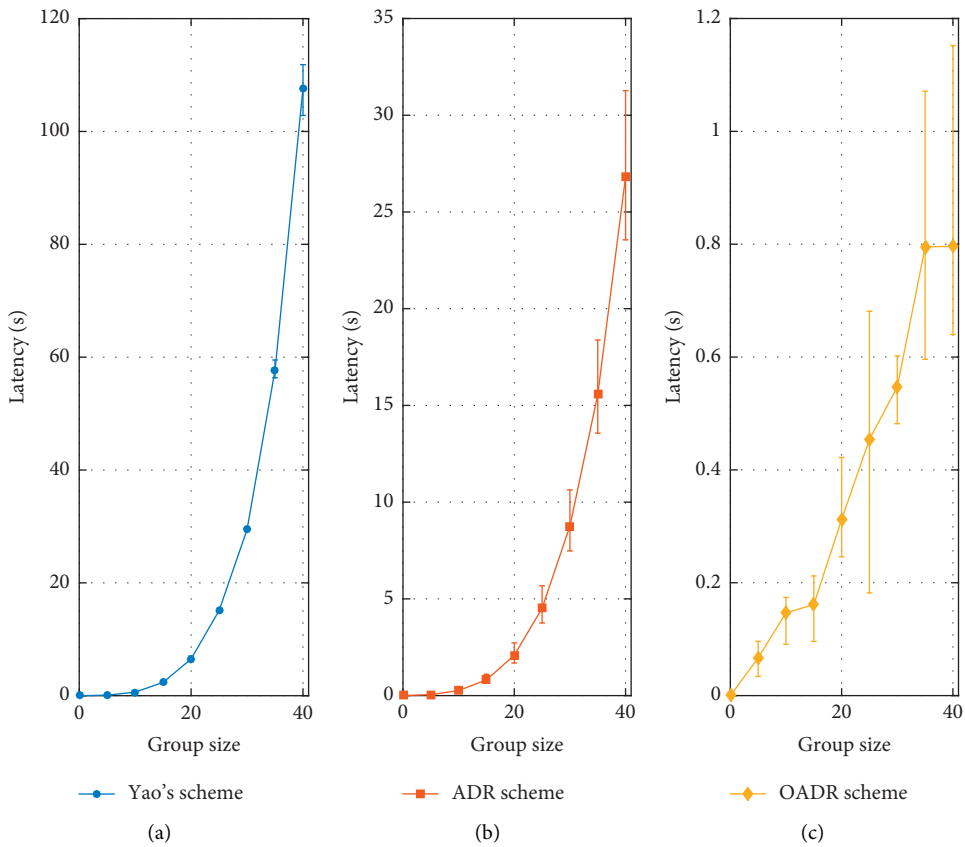

Figure 6: Latency of slot reservation stage for different group sizes.

ranges of OADR scheme consumption is still relatively large as the slot reservation stage needs to rerun if slot collision occurs. Nevertheless, it incurs 0.1537 J for a group of 40 participants, while the consumptions of ADR and Yao's scheme are about 3.7926 J and 0.8961 J, respectively, for the same group size.

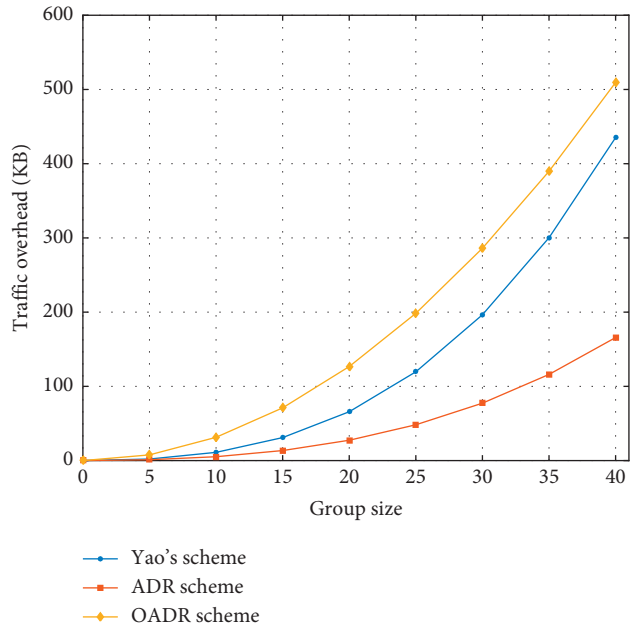

Figure 7: Traffic data overhead for slot reservation stage.

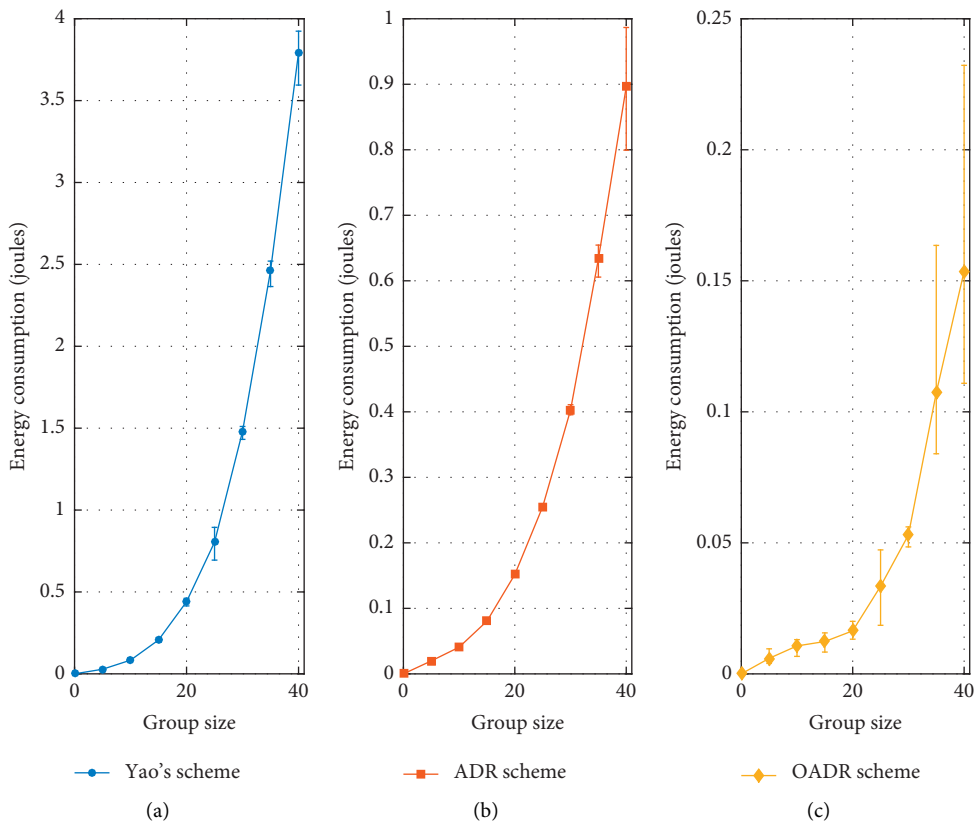

Figure 8: Consumption of slot reservation for different group sizes.

6.5. Data Submission Stage. We evaluate the data submission stage efficiency in terms of latency. For simplicity, we use the same data length for all members, and then we compute the latency of data submission for increasing group size starting from 0 to 40 with a 5-step increment. Because the ADR and Yao's schemes use the same data submission, only ADR and OADR schemes are compared here. Figure 9 shows the data submission latency for different schemes. The data lengths

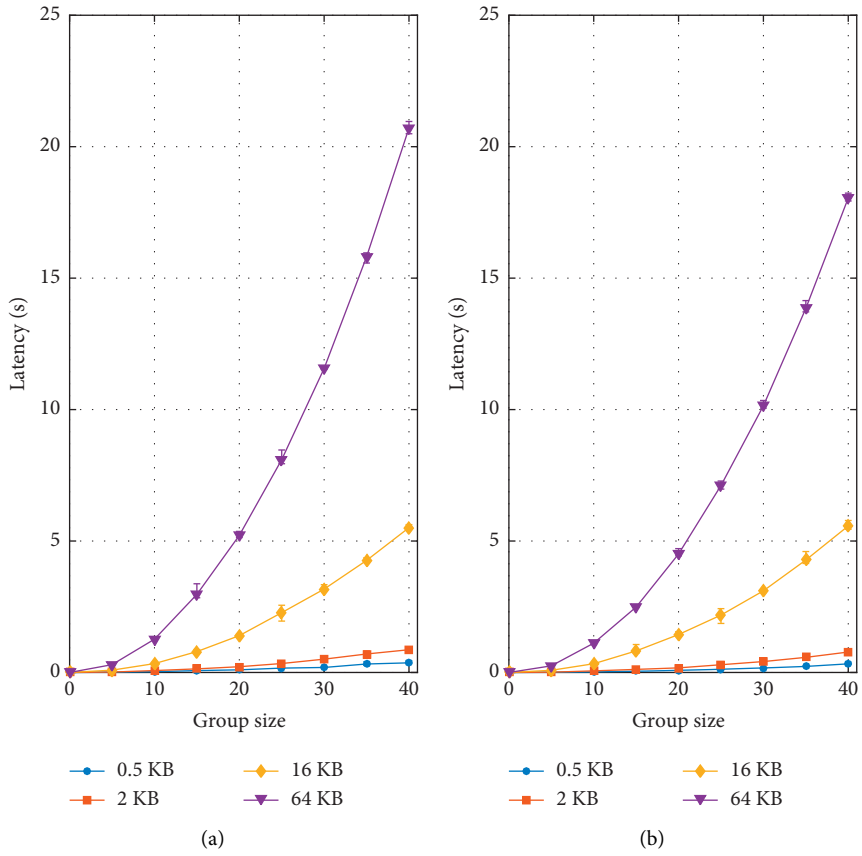

Figure 9: Latencies of data submission for different group sizes. (a) ADR scheme. (b) OADR scheme.

are 0.5 KB, 2 KB, 16 KB, and 64 KB, respectively. When the data lengths are short (0.5 KB and 2 KB), the latencies increase slowly as the group sizes increases, which for a 40-member group are 0.366 s and 0.865 s for 0.5 KB and 2 KB data lengths, respectively, using the ADR scheme. In the same condition, the OADR scheme uses 0.327 s and 0.777 s to submit the data. They are acceptable latencies for low-latency applications. For 64 KB of data length, the proportion of N Pseudo random and XOR function computing latency is significantly increased, and the overall latencies increase dramatically as the group increases. The ADR and OADR schemes grow to reach 20.71 s and 18.05 s, respectively, when considering a group of 40 members. The latency is reduced by approximately 10%. Although OADR does not reduce the number of bits in XOR operations, the number of generations of pseudorandom is reduced to $1/N$ (N is the group size) of the ADR scheme.

The energy consumption increases with the increase of the group size, although there are some fluctuations. The increase of the group size will increase the number of secret seeds and thus the number of calculations of the pseudo-random function. These fluctuations are also due to low consumption, which is easily affected by the particle of the energy monitoring of the Android system. On the other hand, the total stream bit vector length increases relatively to the group size. As shown in Figure 10, it is similar to the latency experiment; the consumptions increase slowly as the group sizes increase for the short data length. They are 1.0391 J and 0.8212 J at 0.5 KB data length for ADR and OADR scheme, respectively. For the larger data lengths, the consumption overhead begins to be large and noticeable. With 64 KB data, they are 36.6649 J and 32.1684 J, respectively. Compared to the ADR, the consumption reduction rate of the OADR scheme is more than 10%.

For participating sensing applications, the amount of data submitted is usually small. Therefore, bulk transfer does not lead to significant additional latency or energy consumption. When group size is 40 to transfer 2 KB data, the latency and energy consumption are still lower than 0.8 s and 1.6 J, respectively. On the other hand, the group size will also significantly affect the latency and energy consumption, so that the latency and energy consumption can be better restricted by reducing the group size. Therefore, the bulk transfer is suitable for low latency tolerant applications of participating sensing .

6.6. Token Request and Deposit Stages. We evaluate the incentive scheme by computing the latency of token signing by TR and token verification by CA for increment group size (Figure 11). The latency of blind signature signing for a group of 40 members is about 61.9 ms (a mean of 1.5 ms per member), which is negligible compared to data submission latency. For the verification of the same number of members, the consumed latency is about 73.5 ms, even it performs

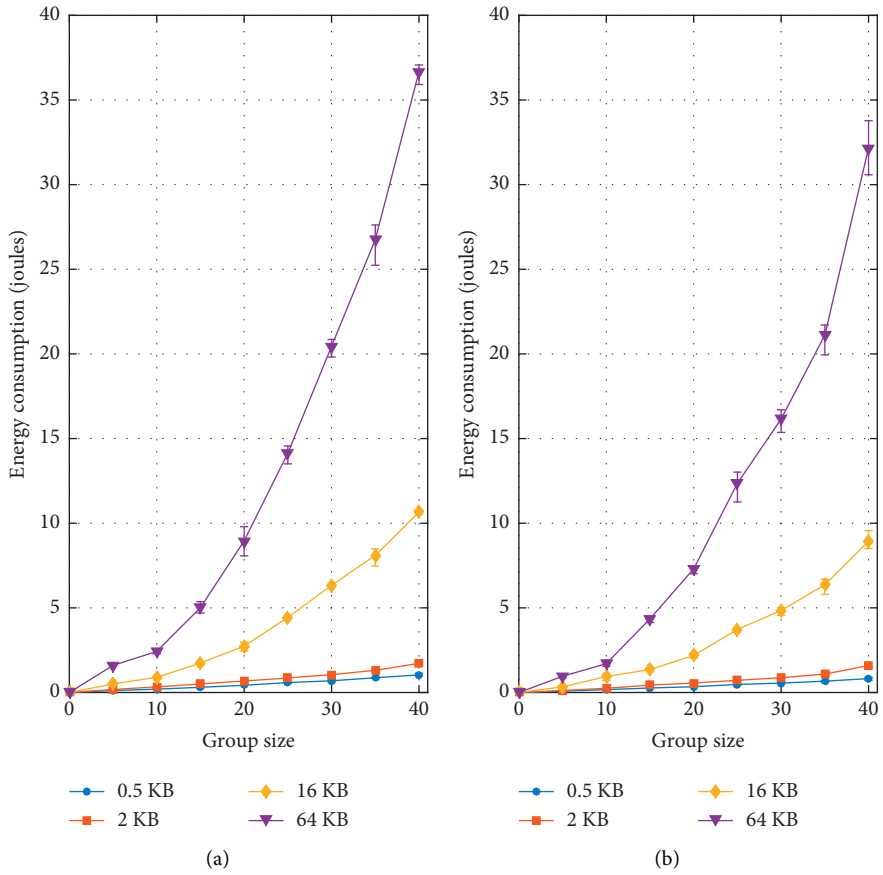

Figure 10: Consumption of data submission for different group sizes. (a) ADR scheme. (b) OADR scheme.

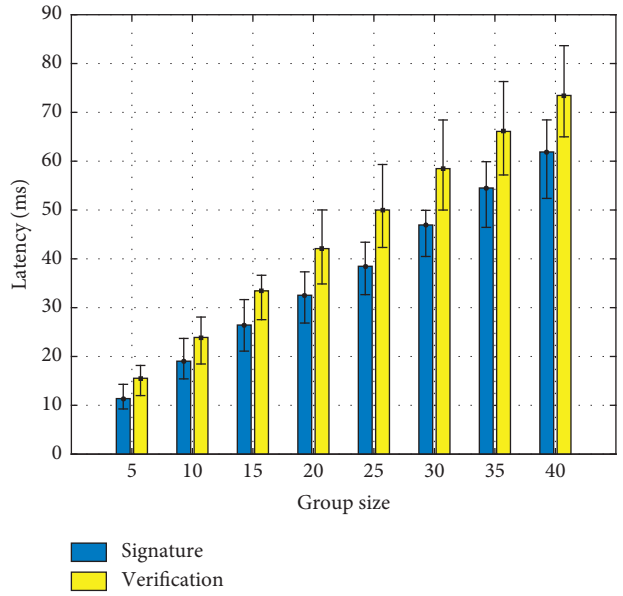

Figure 11: Latencies of blind signature in TR and verification in CA for different group sizes.

encryption, decryption, and an unblind signature. Thus, our proposed micropayment mechanism is efficient. In addition, Table 2 shows the time it takes for the mobile phone to generate and cancel the blind signature, where the process of generating the blind signature includes generating the r factor. The blind signature scheme does not consume a lot of

Table 2: Blind signature related time and consumption in mobile phone device.

Time and consumption	Interval	Average (ms)
Blinding signature time	[3.215 ms, 7.721 ms]	4.062
Blinding signature consumption	[0.052 J, 0.141 J]	0.081
Unblinding signature time	[1.831 ms, 4.6 ms]	2.21
Unblinding signature consumption	[0.024 J, 0.027 J]	0.025

resources, and the signature only spends 4.06 ms for one participant. Therefore, it fits perfectly to be used on phones.

7. Conclusions

In order to solve the privacy exposure by associating participant identities with multimodal information hidden or attached to reported data, we proposed an anonymous data reporting strategy with dynamic incentive mechanism for participatory sensing. The proposed protocol leverages the verifiable random slot selection to establish a transmission plan and then uses multiplayer DC-nets and bulk transfer for anonymous data submission, so that the participant's identity, which is linked to the multimodal information hidden in the data, is confused among a group of participants. The incentive mechanism uses a method similar to data transmission to break the connection between additional information and the participant's identity, and combines blindly signed tokens to anonymously verify the availability of the participant's identity. Furthermore, it reversely uses the blind signature mechanism to carry the dynamic incentives to anonymously complete micropayments transfer. This method is also suitable for the transfer of other similar additional information. In addition, we improve the bulk transfer used in previous works to increase the transmission efficiency while maintaining the same anonymity. The integrity, anonymity, and efficiency of this protocol are proved by theoretical analysis. We implemented and tested the prototype of our protocol. Experimental results on Android phones show that the protocol has advantages comparing with similar anonymous data reporting protocols. These results also show that this protocol is effective for low latency tolerance applications with a certain delay tolerance, which is the case for most participatory sensing applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by National Key R&D Program of China under Grant No. 2020YFB1710200, National Natural Science Foundation of China under Grant No. 62072236, and Fundamental Research Funds for the Central Universities under Grant No. 3072020CFT0603.

References

[1] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, 2011.

[2] Q. Li and G. Cao, "Providing efficient privacy-aware incentives for mobile sensing," *IEEE International Conference on Distributed Computing Systems*, vol. 15, no. 6, pp. 208–217, 2014.

[3] J. Son, D. Kim, R. Hussainy et al., "Privacy aware incentive mechanism to collect mobile data while preventing duplication," in *Proceedings of the IEEE MILCOM*, pp. 1242–1247, Tampa, FL, USA, October 2015.

[4] Q. Li and G. Cao, "Providing privacy-aware incentives in mobile sensing systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1485–1498, 2016.

[5] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

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

[7] H. Wu, L. Wang, and G. Xue, "Privacy-preserving and trustworthy mobile sensing with fair incentives," in *Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC)*, IEEE, Shanghai, China, May 2019.

[8] Y. Li, Y. Zhao, S. Ishak et al., "An anonymous data reporting strategy with ensuring incentives for mobile crowd-sensing," *Journal of Ambient Intelligence & Humanized Computing*, vol. 9, no. B, pp. 1–15, 2017.

[9] J. Brickell and V. Shmatikov, "Efficient anonymity-preserving data collection," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 76–85, Philadelphia, PA, USA, August 2006.

[10] H. Corrigangibbs and B. Ford, "Accountable anonymous group messaging," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 340–350, 2010.

[11] H. Corrigangibbs, D. I. Wolinsky, and B. Ford, "Proactively accountable anonymous messaging in verdict," in *Proceedings of the 22nd USENIX Conference on Security*, pp. 147–162, Washington, DC, USA, August 2013.

[12] D. Wolinsky, H. CorriganGibbs, and B. Ford, "Dissent in numbers: making strong anonymity scale," in *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, pp. 179–192, Hollywood, CA, USA, October 2012.

[13] X. Zhao, L. Li, G. Xue et al., "Efficient anonymous message submission," in *Proceedings of the IEEE INFOCOM 2012*, pp. 228–236, Orlando, Florida, USA, June 2012.

[14] Y. Yao, L. T. Yang, N. N. Xiong et al., "Anonymity-based privacy-preserving data reporting for participatory sensing," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 381–390, 2015.

[15] D. Christin, "Privacy in mobile participatory sensing: current trends and future challenges," *Journal of Systems and Software*, vol. 116, pp. 57–68, 2016.

[16] D. Das, P. Mohan, and V. N. Padmanabhan, "Prism: platform for remote sensing using smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 63–76, San Francisco, CA, USA, June 2010.

[17] P. Sui and X. Li, "A privacy-preserving approach for multimodal transaction data integrated analysis," *Neurocomputing*, vol. 30, pp. 56–64, 2017.

[18] F. Zhang, H. Li, W. He et al., "Data perturbation with state-dependent noise for participatory sensing," in *Proceedings of the IEEE INFOCOM 2012*, pp. 2246–2254, Orlando, FL, USA, March 2012.

[19] F. David, F. Kargl, and L. Hans, "PUCA: a pseudonym scheme with strong privacy guarantees for vehicular ad-hoc networks," *Ad Hoc Networks*, 2016.

[20] S. Zhang, G. Wang, Q. Liu, X. Wen, and J. Liao, "A trajectory privacy-preserving scheme based on dual-k mechanism for continuous location-based services," in *Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, pp. 1004–1010, IEEE, Guangzhou, China, December 2017.

[21] H. Zhao, Y. I. Xiao-Ling, and J. L. Wan, "Privacy-area aware all-dummy-based location privacy algorithms for location-based services," *DEStech Transactions on Computer Science and Engineering aice-ncs*, 2017.

[22] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, pp. 37–48, Baltimore, MD, USA, June 2005.

[23] C. Bettini and D. Riboni, "Privacy protection in pervasive systems: state of the art and technical challenges," *Pervasive and Mobile Computing*, vol. 17, pp. 159–174, 2015.

[24] Y. Wang, Z. Cai, Z. Chi, X. Tong, and L. Li, "A differentially *k*-anonymity-based location privacy-preserving for mobile crowdsourcing systems," *Procedia Computer Science*, vol. 129, pp. 28–34, 2018.

[25] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, and Q. Ni, "A *k*-anonymity based schema for location privacy preservation," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 2, pp. 156–167, 2019.

[26] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pp. 160–164, Washington, DC. USA, November 1982.

[27] H. Jiang and Q. Xu, "Advances in key techniques of practical secure multi-party computation," *Journal of Computer Research & Development*, vol. 52, no. 10, pp. 2247–2257, 2015.

[28] M. Blanton and E. Guiar, "Private and oblivious set and multiset operations," *International Journal of Information Security*, vol. 15, no. 4, pp. 493–518, 2016.

[29] J. Zhong, W. Wu, C. Cao et al., "A variable weight privacy-preserving algorithm for the mobile crowd sensing network," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 3053202, 7 pages, 2017.

[30] J. Xiong, R. Ma, L. Chen, Y. Tian, L. Lin, and B. Jin, "Achieving incentive, security, and scalable privacy protection in mobile crowdsensing services," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 8959635, 12 pages, 2018.

[31] J. Zhang, C. Zhe, L. Ting et al., "Design scheme of electronic jury system based on secure multi-party computation," *Journal of Computer Applications*, vol. 40, no. S2, pp. 80–84, 2020.

[32] J. Cheon, A. Kim, M. Kim et al., "Homomorphic encryption for arithmetic of approximate numbers," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, pp. 409–437, Hong Kong, China, December 2017.

[33] S. Fletcher and M. Z. Islam, "Differentially private random decision forests using smooth sensitivity," *Expert Systems with Applications*, vol. 78, pp. 16–31, 2016.

[34] M. Bun and T. Steinke, "Concentrated differential privacy: simplifications, extensions, and lower bounds," *Theory of Cryptography. TCC 2016*, https://arxiv.org/abs/1605.02065, 2016.

[35] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *Journal of Machine Learning Research*, vol. 17, no. 17, pp. 1–51, 2016.

[36] N. Holohan, D. J. Leith, and O. Mason, "Extreme points of the local differential privacy polytope," *Linear Algebra & Its Applications*, vol. 534, pp. 78–96, 2017.

[37] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.

[38] Q. Ye, H. Hu, X. Meng et al., "PrivKV: key-value data collection with local differential privacy," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP) IEEE*, pp. 127–143, San Francisco, CA, USA, May 2019.

[39] L. Sun, J. Zhao, X. Ye et al., "Conditional analysis for key-value data with local differential privacy," 2019, https://arxiv.org/abs/1907.05014v1.

[40] Y. Sei and A. Ohsuga, "Differentially private mobile crowd sensing considering sensing errors," *Sensors*, vol. 20, no. 10, p. 2785, 2020.

[41] T. Ni, Z. Chen, G. Xu et al., "Differentially private double auction with reliability-aware in mobile crowd sensing," *Ad Hoc Networks*, vol. 114, Article ID 102450, 2021.

[42] C. Cornelius, A. Kapadia, D. Kotz et al., "Anonysense: privacy-aware people-centric sensing," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, pp. 211–224, Breckenridge, CO, USA, June 2008.

[43] E. D. Cristofaro and C. Soriente, "Extended capabilities for a privacy-enhanced participatory sensing infrastructure (PEPSI)," *IEEE Transactions on Information Forensics and Security*, vol. 8, pp. 2021–2033, 2013.

[44] R. Shokri, G. Theodorakopoulos, P. Papadimitratos et al., "Hiding in the mobile crowd: locationprivacy through collaboration," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, pp. 266–279, 2014.

[45] F. Qiu, F. Wu, and G. Chen "Slicer, "A slicing-based k-anonymous privacy preserving scheme for participatory sensing," in *Proceedings of the IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 113–121, Hangzhou, China, October 2013.

[46] D. Chaum, "Blind signatures for untraceable payments," in *Proceedings of the CRYPTO'82*, pp. 23–25, Santa Barbara, CA, USA, August 1982.

[47] D. Chaum, "Blind signature system," in *Proceedings of CRYPTO'83, Advances in Cryptology*, pp. 21–24, Santa Barbara, CA, USA, August1983.

[48] S. Gisdakis, T. Giannetsos, and "S. Papadimitratos, "Security and privacy-preserving architecture for participatory-sensing applications," in *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 39–50, Oxford, UK, July 2014.

[49] Y. Qiu, M. Ma, S. Chen et al., "An anonymous authentication scheme for multi-domain machine-to-machine communication in cyber-physical systems," *Computer Networks*, vol. 129, 2017.

[50] T. Gao, Q. Wang, X. Wang, and X. Gong, "An anonymous access authentication scheme based on proxy ring signature for CPS-WMNs," *Mobile Information Systems*, vol. 2017, pp. 1–11, 2017.

[51] Y. Gong, C. Ying, Y. Guo et al., "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1304–1313, 2017.

[52] X. Wang and D. Reeves, *Traceback and Anonymity*, Springer Publishing Company, Incorporated, New York, NY, USA, 2015.

[53] D. Christin, J. Guillemet, A. Reinhardt et al., "Privacy-preserving collaborative path hiding for participatory sensing applications," in *Proceedings of the IEEE Eighth International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 341–350, Valencia, Spain, October 2011.

[54] S. Goldwasser and S. Micali, "Probabilistic encryption how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 365–377, San Francisco, CA, USA, May 1982.

[55] B. Laurie, "Lucre: anonymous electronic tokens," 2008, https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=90F5A5D1F07340A314BB493534C50866?doi=10.1.1.112.7291\&rep=rep1\&type=pdf.