J Inf Process Syst, Vol.15, No.5, pp.1036~1054, October 2019
https://doi.org/10.3745/JIPS.01.0045

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Privacy-Preservation Using Group Signature for Incentive Mechanisms in Mobile Crowd Sensing

Mihui Kim*, Younghee Park**, and Pankaj Balasaheb Dighe**

Abstract

Recently, concomitant with a surge in numbers of Internet of Things (IoT) devices with various sensors, mobile crowdsensing (MCS) has provided a new business model for IoT. For example, a person can share road traffic pictures taken with their smartphone via a cloud computing system and the MCS data can provide benefits to other consumers. In this service model, to encourage people to actively engage in sensing activities and to voluntarily share their sensing data, providing appropriate incentives is very important. However, the sensing data from personal devices can be sensitive to privacy, and thus the privacy issue can suppress data sharing. Therefore, the development of an appropriate privacy protection system is essential for successful MCS. In this study, we address this problem due to the conflicting objectives of privacy preservation and incentive payment. We propose a privacy-preserving mechanism that protects identity and location privacy of sensing users through an on-demand incentive payment and group signatures methods. Subsequently, we apply the proposed mechanism to one example of MCS—an intelligent parking system—and demonstrate the feasibility and efficiency of our mechanism through emulation.

Keywords

Incentive Method, Internet of Things (IoT) Model, Mobile Crowd Sensing (MCS), Privacy-Preserving, Using Group Signature

1. Introduction

Physical objects in industrial, mobile, and home environments are increasingly being transformed from isolated systems to networked Internet-enabled devices through information communication technology (ICT) to further expand the Internet of Things (IoT). Sensing data from IoT devices are shared to provide many new services in our lives. Cisco forecasts that 50 billion devices will be connected to the Internet by 2020 [1], and IDC reports that these devices and sensors will generate about 180 zettabytes of data by 2025 [2]. Most IoT devices are resource-constrained; thus, IoT technologies are supported by cloud systems. These cloud-assisted IoT technologies have made a new business model, mobile crowdsensing (MCS) [3-7]. In MCS, sensing data providers supply data obtained from mobile devices through publishing sensing data. Then, the crowdsourced big data are gathered and processed on the MCS infrastructure. Sensing data consumers acquire the sensing data from the cloud MCS infrastructure on-

Manuscript received August 31, 2018; first revision December 27, 2018; second revision February 27, 2019; accepted April 18, 2019; onlinefirst July 15, 2019.

Corresponding Author: Younghee Park (younghee.park@sjsu.edu)
* Dept. of Computer Science & Engineering, Computer System Institute, Hankyong National University, Anseong, Korea (mhkim@hknu.ac.kr)
** Dept. of Computer Engineering Department, San Jose State University, San Jose, CA, USA ({younghee.park, pankajbalasaheb.dighe}@sjsu.edu)

demand. For example, a person near a ski resort may share road condition pictures taken with their smartphone via a cloud computing system and the mobile crowd sensing data may aid other consumers that plan to go to that ski resort.

The most important factors for a successful MCS business are real-time and adequate data supply from providers. Sufficient data can provide meaningful information to sensor consumers through analysis and processing of the data in the cloud. Accordingly, appropriate incentive mechanisms are essential to encourage the active participation of providers. Much incentive researches thus have been conducted [8-10], but the main obstacle to sharing sensing data is the privacy issue [11]. Many sensing data reveal personal information, e.g., location, life pattern, health status, and movement route. Therefore, if an appropriate privacy protection mechanism is not provided for MCS, potential users who are concerned about privacy invasion may not join in. On the other hand, if users conceal their identity and the sensing data via simple encryption algorithms, the encrypted message makes it impossible to provide an incentive. Therefore, this problem caused by the conflicting objectives of privacy protection and incentive payment should be carefully resolved.

Researchers have proposed various methods to solve the above problem, i.e., auction-based mechanisms with only winner information exposure, auction-based mechanisms without immediate information exposure, and mechanisms by anonymity authentication. First, earlier studies [12,13] exposed only the information of winners in the auction. However, the private information of winners also should be also kept private. Second, other studies that utilize auction processes include [6,14,15], who provide mechanisms for protecting the shared sensing data or the information for the auction process (e.g., bidding, sensing, and reputation data in the auction process) such as time-lapse cryptography (TLC) [6,14] and differential privacy [15]. However, the server managing the security keys in [6,14,15] can trace the private information of all bidders. Third, methods have also been proposed that both guarantee anonymity during participation and incentive payment [16-19] by applying *K*-anonymity [17], mix-zone [18], group signature [19], or identity-committable signatures (ICS) [16]. However, these complicated processes can hinder the real-time data gathering or on-demand incentive compensation. Our proposal pursues a simple on-demand incentive mechanism with privacy-preservation.

In this paper, we address the problem arising from the conflicting objectives of incentive payment and privacy preservation and design an on-demand incentive mechanism with privacy preservation (called oIMP) for MCS to reveal identity and location information in shared data. The mechanism enhances MCS with the following features: (1) incentive for providers, it boosts crowd sensing sharing without anxiety about identity and location privacy invasion; (2) it differentiates the authentication time in sharing data from incentive payment time; thus, it provides both identity privacy and location privacy protections; and (3) it can support resource-constrained IoT devices that share the sensing data with a simple privacy preservation mechanism. Our mechanism is based on a general MCS infrastructure consisting of providers, consumers, and provisioning server. Providers register on the provisioning server, obtain the key materials for group signatures, and prepare their own temporal IDs using a hash chain. Group signatures enable participants to sign messages on behalf of a group in an anonymous manner and then the verifier can verify whether the signer is a member in the group without revealing the identity of the signer [20]. When a consumer requests a service of the provisioning server, the server announces it to providers. Then, providers who can provide the requested sensing data publish the data together with a temporal ID instead of the real identity and signature to the server. The server subsequently authenticates the signature, i.e., ascertains whether it is generated from a member in the group or not. If yes, the server uses the data for service provisioning. The server then stores the provisioned data amount with temporal ID. When the provider requests incentive payment with the seed of temporal IDs, the server can calculate all temporal IDs from the seed and retrieve the data amount provided by the provider. Finally, the server offers the provider the total incentive calculated from the provided data amount. As our previous work [21], we briefly sketched the incentive mechanism with privacy-preservation in MCS. In this paper, we design our oIMP in detail based on [21] and evaluate it through emulation with developing two types of provider devices and MCS servers. The contributions of this work are as follows:

- Comparison of incentive mechanisms considering privacy issue in MCS.
- Design of an on-demand incentive mechanism after the provider share their data, and apply group signature to authenticate the providers while concealing their identity and location.
- Emulation of the proposed mechanism to ensure the feasibility and evaluate the performance. We use two types (i.e., high and low specifications) of systems: a laptop and an Intel edition board as provider devices, and a workstation system and Amazon Web Service (AWS) cloud system as MCS servers.
- Evaluation of system performances, i.e., initialization time, sign/verification time for group signature, and scalability.

The remainder of this paper is organized as follows: Section 2 discusses existing work related to privacy-preserving incentive approaches for IoT. Section 3 presents an overview of MCS architecture and state intrusion model in MCS to address in this paper. Section 4 proposes the privacy-preserving incentive mechanism. Section 5 presents the emulation results for operation and performance. Finally, Section 6 states our conclusions and discusses future work.

2. Related Work

The most important factor for successful MCS is numerous participants for sensing data provision. The proper incentive mechanism is indispensable to boost participation. However, in general, sharing the sensing data (e.g., identity, location, and life pattern) causes a privacy exposure problem, and thus the tradeoff problem between incentive payment and privacy preservation exists. More specifically, revealing the identity of the participant at least in sharing the sensing data is necessary to pay the incentive, but the exposure can incur privacy invasion. Therefore, various mechanisms to resolve the tradeoff problem have been proposed. To address the problem, existing work can be divided into three categories: auction-based mechanisms with only winner information exposure, auction-based mechanisms without immediate information exposure, and mechanisms by anonymity authentication.

Auction-based mechanisms with only winner information exposure include [12,13], who assumed that the winners in the auction are compensated for private information exposure. The mechanism in [12] induces the price at which participants are willing to expose their location, utilizing experimental economics and psychology. In a case study, students were required to conduct seal bidding for location exposure, and the n students with the lowest bids were chosen as winners and they were given the lowest price of the bidders who were not chosen. This auction structure can infer a suitable price for location exposure, but the price may differ according to the user types or applications. The system in [13] prevents privacy leak of losers without compensation in an auction. It has a platform in which users in a bidding process expose ambiguous locations and only winners expose the real location information after the auction is finished. However, the individual information has to be exposed because the incentive has already been obtained, and the exposure of privacy even with the compensation may incur more damages, such as burglary. Moreover, some personal information cannot be exchanged even with monetary compensation (e.g., health information).

Auction-based mechanisms without immediate information exposure have also been proposed. In auction processes to decide the winners, i.e., workers for sensing, with these mechanisms, some researchers try to protect the information in the auction process, e.g., bidding, sensing, and reputation data. In [6,14], TLC is employed to conceal the information. In TCL, the public key used for the encryption is first disclosed, and the private key for decryption is exposed in the end stage. Similarly, to protect the bid information of workers against honest-but-curious workers, the authors [15] design a differentially private incentive mechanism that makes it difficult for curious workers to infer information about the bids of other workers from the outcome (i.e., payment profile). However, the server managing the security keys in [6,14,15] can trace the private information of all bidders.

Mechanisms that ensure both participation anonymity and incentive payment, i.e., mechanisms with anonymity authentication, have also been proposed [6,16,18,19]. The authors [17] enhance *K*-anonymity, the most common and classic solution for location-based service (LBS) privacy protection, by utilizing a reputation mechanism based on fuzzy logic. Their mechanism prevents participants with poor practicability from forming the *K*-anonymity set through reputation values. The authors [18] utilize mix-zone, another common method used to provide participant anonymity. To ensure incentive mechanism untraceability, they employ an e-cent exchange process involving participants in a specific mix-zone and an e-cent renewal process through a server. E-cent, like unit bearer currency, is used as a pledge and a reward for participation, and a credit for service usage. However, these complicated processes can hinder the real-time data gathering. The authors [16,19] proposed authentication mechanisms without identity that apply whether the providers are proper users or not, but utilize group signatures for proper users and ICS, respectively. The mechanism in [16] also has too complicated architecture (i.e., group manager, identity provider, and pseudonymous certification authority) for authentication. The mechanism in [19] applies two cloaking methods to prevent other customers from inferring the real identity of a customer by mining the correlations between rewards and withdrawals, but the customer cannot freely withdraw the rewards on-demand due to these cloaking rules. Table 1 summarizes the existing researches and compares them with our proposed oIMP.

It is clear that none of these existing mechanisms resolves both anonymity authentication in sharing the sensing data to preserve privacy against the server and other users, and simple on-demand incentive payment. Our proposed on-demand incentive mechanism with privacy preservation for MCS, called oIMP, resolves these two problems. As in [16-19], we propose to authenticate a genuine member of a group through group signatures, but the users are compensated on-demand. That is, before the users require compensation for sharing their sensing data, their private information is protected from other users and even the provisioning server.

Table 1. Comparison of incentive mechanisms considering privacy issue in MCS

Type	Ref.	Exposed object & time	Protected data	Methods used for privacy protection
Auction-based mechanisms with only winner information exposure	[12]	All	Location	-
	[13]	All	Location of loser	-
Auction-based mechanisms without immediate information exposure	[6]	All after completion of auction	Bidding, sensing, reputation data	TLS
	[14]	All after completion of auction	Bidding, selection preferences, and identity	TLS
	[15]	Server	Bidding	ϱ-differential privacy
Mechanisms by anonymity authentication	[17]	Neighbor	Location	K-anonymity
	[18]	Server	Sensing data	Mix-zone
	[19]	Server in incentive payment with strict rule	Identity and metering data in smart grid	Group signature
	[16]	Server after finishing a sensing task and paying incentive	Identity and sensing data	ICS
	oIMP	Server in on-demand incentive payment	Identity and sensing data	Group signature

3. Background

This section presents an overview of MCS architecture based in presenting our oIMP. We introduce a problem of conflicting objectives between revealing the user's private information in sharing the sensing data and incentive payment. We state intrusion model in MCS to address in this paper.

3.1 Overview of MCS Architecture and a MCS Example

This subsection describes the basic MCS system architecture, which is divided into three main parts: Provider, Consumer, and Cloud Infrastructure, as shown in Fig. 1 (refer to [21] for details). It then discusses a MCS example to which oIMP can be applied.

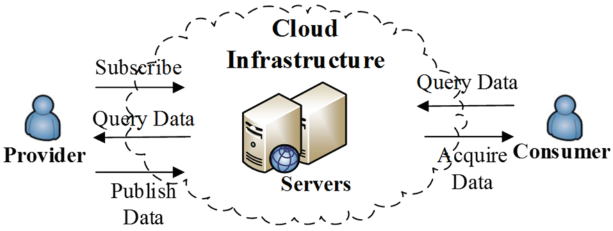

Fig. 1. Basic MCS system architecture.

To concretely explain oIMP, we utilize an intelligent parking system MCS. This service can be made a MCS without installation of expensive sensors. In recent intelligent parking systems, a sensor is installed per parking lot for sensing whether it is empty or not, or each installed closed-circuit television (CCTV) monitors parking lots in a narrow area to check its vacancy status. However, the installation of such systems is too expensive; thus, to date, the transition to intelligent parking system is slow. This bottleneck can be addressed through MCS with crowd sensing.

Most people, especially those with their own cars, now possess smartphones with high quality cameras. Moreover, many smart cars have a rear camera for reverse parking or are equipped with a black box in preparation for traffic accidents. This means that providers with individual devices including cameras can provide pictures taken around parking lots to one-time or periodically upload to servers in MCS, after they park their cars in a specific parking lot. Thus, the following MCS based parking system scenario is possible: Provisioning server gathers real-time parking status from providers in MCS and analyzes them with real-time and past information. Consumers query the parking information when they arrive at the entrance to the specific parking garage or they need the information. The provisioning server provides the consumers with the parking information, e.g., how many vacant parking lots exist on each level, where the vacant parking lots exist, and the shortest route to the nearest vacant parking lot.

In order to successfully operate the intelligent parking system through MCS, gathering as much real-time data as possible is essential; thus, an incentive mechanism is invaluable in this scenario. We designed the following incentive mechanism. This incentive is paid to providers whenever they provide accurate parking lot information, and the consumers need the incentive to utilize the intelligent parking system. This circular incentive flow can boost the voluntary participation of users. Fig. 2 shows the sequence of activities in a parking system scenario based on MCS infrastructure. Providers subscribe to the provisioning server before participating in the service. When a consumer requests parking information related to a specific parking garage from the provisioning server, the server informs the providers of the request. The providers in the requested vicinity then publish the pictures taken by their devices with camera to provisioning server. The provisioning server processes the pictures to get the vacant information of parking lots, and then the provisioning server provides the information to the requesting consumer. Finally, the provisioning server informs incentive management server of the process with the provided data amount or data quality, in order to reward the incentive to providers and to charge consumers for the proper fee (i.e., subtract the proper fee among the possessing incentive).

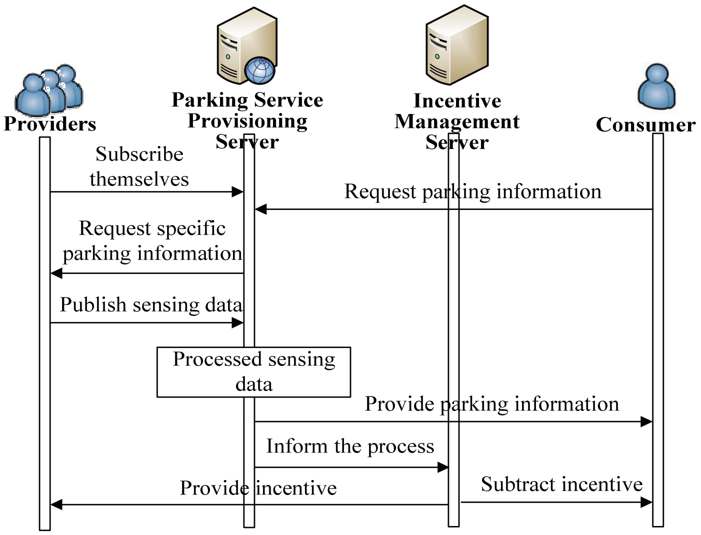
Providers
Parking Service Provisioning Server
Incentive Management Server
Consumer
Subscribe themselves
Request parking information
Request specific parking information
Publish sensing data
Processed sensing data
Provide parking information
Inform the process
Provide incentive
Subtract incentive

Fig. 2. The process sequences of a parking MCS scenario.

3.2 Intrusion Model

In the parking MCS scenario considering in this paper, there are some problems from the view of security issue. First, when providers publish the information for parking lot, the process reveals their identity and location to provisioning server. The intruder can monitor the current location for providers by threatening the server or by eavesdropping the communication between providers and provisioning server, i.e., privacy problem of provider. Second, consumers reveal their location when requesting the specific parking information. The location of consumers can be also exposed to intruder, i.e., location privacy problem of consumer. However, this second problem can be addressed by revealing obscure location information when consumers request the service [22], by anonymizing the consumer in a group [23], or by using cloaking areas instead of the exact coordinates [24].

In this paper, we focus on resolving the first problem utilizing group signature and on-demand incentive payment to providers.

4. On-demand Incentive Method with Privacy Preservation

This section presents an overview of our oIMP mechanism with MCS architecture and applies group signature to authenticate the provider. In MCS, there is a problem of conflicting objectives between revealing the user's private information in sharing the sensing data and incentive payment. The novelty of oIMP is that it differentiates the authentication time in sharing the sensing data and the incentive payment time, thereby resolving the conflict problem.

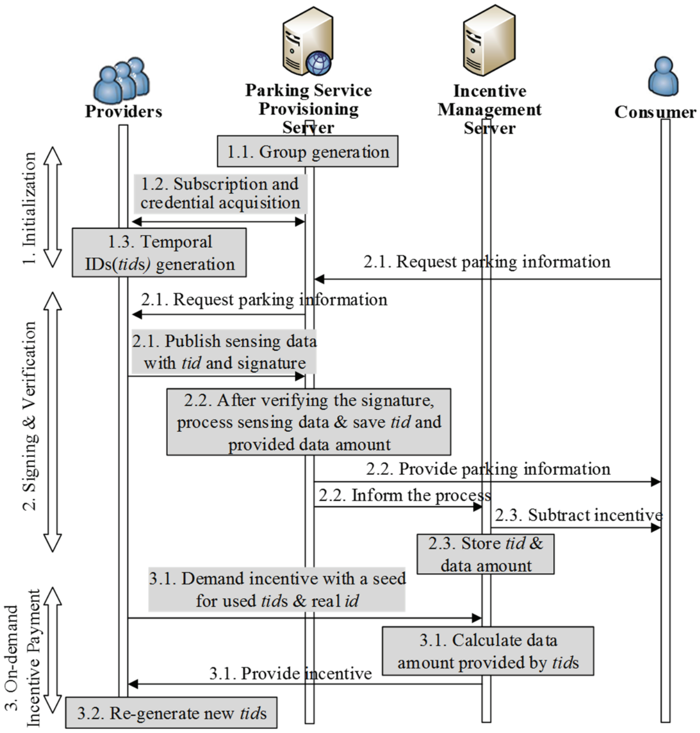
Providers
Parking Service Provisioning Server
Incentive Management Server
Consumer
1. Initialization
2. Signing & Verification
3. On-demand Incentive Payment
1.1. Group generation
1.2. Subscription and credential acquisition
1.3. Temporal IDs(*tids*) generation
2.1. Request parking information
2.1. Request parking information
2.1. Publish sensing data with *tid* and signature
2.2. After verifying the signature, process sensing data & save *tid* and provided data amount
2.2. Provide parking information
2.2. Inform the process
2.3. Subtract incentive
2.3. Store *tid* & data amount
3.1. Demand incentive with a seed for used *tids* & real *id*
3.1. Calculate data amount provided by *tids*
3.1. Provide incentive
3.2. Re-generate new *tids*

Fig. 3. Process sequences of parking MCS based oIMP.

4.1 oIMP Mechanism

This subsection explains the oIMP with MCS based parking system, as outlined in previous subsection.

The unique difference with the oIMP mechanism is its anonymous authentication and on-demand incentive system, in comparison with the basic parking MCS. Our oIMP anonymously authenticates the identity of the provider with group signature scheme in sharing the sensing data. That is, this authentication does not reveal each identity, but verifies one of service users in the MCS. When the provider requests the incentives, oIMP pays incentives for the data the provider has provided so far. Only then exposes the identity of user requesting the incentives.

Group signature scheme [25] allows participants to anonymously sign messages on behalf of a group. The signature verifier can verify whether the signer is a true member in the group. The verification process does not reveal the identity of the signer and thus it preserves privacy [20].

To provide the identity and location privacy, our oIMP mechanism utilizes three steps: Initialization, Signing & verification, and On-demand Incentive Provisioning, as shown in Fig. 3.

4.1.1 Initialization

- Group generation: To provide group signature, a provisioning server creates group information. The server may create a group or multiple groups per a MCS service, and the number of groups depends on the number of users in the MCS service. For example, in case of MCS smart parking service for a building, limitary parking users exist and thus a group is sufficient. However, in case of MCS smart parking service at the shopping mall with many branches, a large number of transient users parks. Depending on the MCS participant level or main parking branch, each user may assign a group among multiple groups.
- Provider subscription & credential acquisition: Provider subscription makes the provisioning server register each provider as a member of one of the groups and provide a credential for the provider.
- Temporal IDs generation: Based on a hash function H and a randomly selected seed, s_i, the generated temporal IDs are used to connect with the real ID of the provider i in the later incentive payment but in the meantime conceal the real ID. These temporal IDs are used backward (i.e., $tid_i^n \rightarrow tid_i^{n-1} \rightarrow \cdots$) every time the provider shares the sensing data, and stored with the information of providing data in provisioning server for on-demand incentive payment afterward. In on-demand incentive payment, the new chain of temporal IDs is generated:

$$tid_i^n \overset{H}{\leftarrow} tid_i^{n-1} \overset{H}{\leftarrow} \cdots \overset{H}{\leftarrow} tid_i^1 \overset{H}{\leftarrow} s_i$$

4.1.2 Signing & verification

- Request information & publishing: When the consumer requests the parking information, the provisioning server forwards the request to providers. The providers who are in the vicinity of parking lots and can provide the information (e.g., take photos using their devices with camera) publish the pictures with current tid_i^j and signature created with their credential. At this point, their identity and location are not revealed because their temporal ID for the information publishing is used instead of their real ID.

- Verification & information provision: The provisioning server can verify the signature based on the credential of the providers, i.e., authenticate the provider as one of the members of a group. The provisioning server provides processed parking information to the consumers.
- Update incentive information: The provisioning server updates the incentive information at the incentive server by charging the proper fee for the consumers. This process is like subtracting the decided amount from possessing incentive. Subsequently, the incentive management server stores the provided information amount (or calculated incentive) along with tid_i^j.

4.1.3 On-demand incentive provisioning

- Demand incentive & provision: When users need their own incentive (i.e., to obtain parking information as consumers), they request the current cumulated incentives to the incentive management server with real ID i and temporal ID's seed s_i. The incentive server makes up the previous temporal IDs generated from the seed and retrieves the incentive for the temporal IDs. Finally, the incentive server provides the calculated incentive to the user. Even though the server has unnecessarily stored the past location information of the user, the disclosed location information at this point is not current valid data.
- Re-generate new temporal IDs: After obtaining the incentive, the provider regenerates the new TIDs from a new seed value s_i'. For a period using a temporal IDs chain, the providers can conceal their locations in sharing the sensing data with the MCS. That is, the period of demanding incentive can be controlled according to the duration of privacy preservation.

Next, we explain location privacy preservation for consumers. When the consumer requests parking information, our oIMP provides obfuscated location information with the notion of differential privacy [26,27]. Differential privacy is a statistical technique, and the goal is to provide means to maximize the accuracy of queries from statistical databases while measuring and minimizing the privacy impact. Thus, it guarantees strong theoretical privacy with a bounded accuracy loss. These mechanisms in [26,27] are proposed to preserve location privacy in LBS. Users report an obfuscated point of interest (POI) instead of a real POI to the LBS server. Through differential privacy, exact location information is protected by adding a controlled amount of noise according to the desired level of privacy. As a result of the added noise for the location, the LBS server gives the user both the data in which the user is interested and the unnecessary data, i.e., a bounded error of results. Due to the unnecessary data, the communication amount and processing time at user system increase, but the data amount can be controlled. The user can filter the useful information from the received results. For example, if consumers want to know the parking information for a south garage in San Jose State University, consumers in oIMP can provide an obfuscated POI, i.e., the zip code CA95192 covering the south garage to the server. After obtaining the received results, the consumer can filter out parking information in CA95192 except the south garage.

4.2. Applied Authentication Scheme

To authenticate the provider in sharing the sensing data, we apply group signatures. Unlike conventional digital signatures, a group signature scheme ensures the anonymity of the signing participants when the signature is verified. That is, the verifier identifies that the signer is indeed a valid member of a group while the identity of the specific signer is not revealed. It is a good feature for our provider authentication scheme because the identity and location of the provider should be concealed during authentication.

The group signature scheme, first introduced in [25], allows participants to sign messages on behalf of a group in an anonymous manner. Subsequently, other extended versions [20,28,29] have been developed with various properties: soundness and completeness, anonymity, unforgeability, and traceability. Because of the anonymous signing feature in a group, the applications of group signatures are various, such as e-voting, e-bidding, online payment, and anonymous attestation [20]. The group signature scheme comprises four processes: group establishment, joining, signing, and verification. These processes are explained below and showed in Fig. 4:

- **Group establishment**: As an initialization process, an issuing authority set up a group and creates its public key and the corresponding group membership issuing key.
- **Joining**: When an applicant requests to join a group with commitment for a secret value, the issuing authority adds the new member to the group and issues a membership credential based on the group membership issuing key and the received commitment, including the secret value. The new member has its own credential for signing at the end of the joining process.
- **Signing**: A valid member can sign some data with the secret value and the membership credential, then the member can be authenticated for data (i.e., be identified as a member of the joined group with the real identity being concealed).
- **Verification**: A verifier can verify the signature using the group public key, whether the signature was issued by a valid member of the group or not.

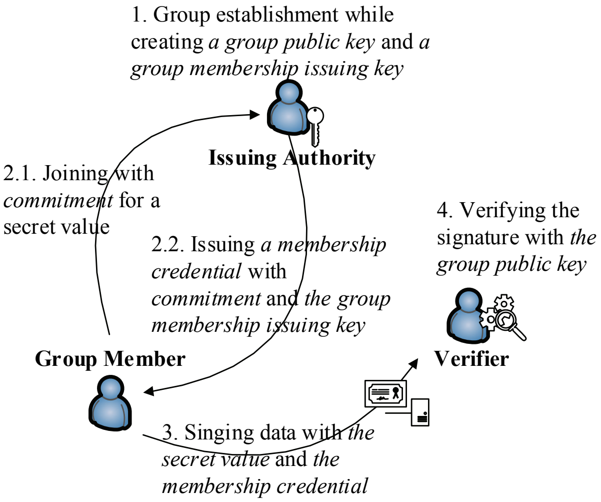
1. Group establishment while creating *a group public key* and *a group membership issuing key*
Issuing Authority
2.1. Joining with *commitment* for a secret value
2.2. Issuing *a membership credential* with *commitment* and *the group membership issuing key*
4. Verifying the signature with *the group public key*
Group Member
Verifier
3. Singing data with *the secret value* and *the membership credential*

Fig. 4. Processes of group signature scheme3.

For performance evaluation, we applied a pairing-based elliptic curve cryptography-direct anonymous attestation (ECC-DAA) scheme [28] as the group signature because the competitive performance of the ECC-DAA scheme has been verified on mobile devices, unlike other group signature schemes [20]. DAA was originally introduced as a mechanism to remotely authenticate a trusted platform—i.e., trusted platform module (TPM)—in a privacy-preserving manner. DAA consists of issuers, signers, and verifiers. An issuer verifies the legitimacy of signers and issues a DAA credential to signers. A signer is originally divided into two signers: less powerful principal signer (TPM) and powerful assistant signer (Host) into which the TPM is embedded. However, we used the codes that apply them to a signer in [28,29] for emulation. The signer provides the signature to a verifier in order to prove a membership with a valid DAA credential. The verifier can verify the membership credential but cannot know the identity of the signer. The protocols of the pairing-based ECC-DAA scheme are outlined below and showed in Fig. 5 (see [28] for more detailed information).

- **Setup protocol (the same as group creation):** As initialization in this protocol, the system generates four types of parameters: commitment parameter par_C, signature and verification parameter par_S, issuer parameter par_I, and TPM parameter par_T. It then publishes $(par_C, par_S, par_I, par_T)$.
- **Join protocol:** Signer (i.e., working together with TPM and Host) requests to join to issuer through the *createJoinRequest* process (TPM commits with a TPM's secret value sk_T), and the issuer responds through the *respondToJoin* process (Issuer checks the received commitment and creates a credential for the TPM). Finally, the signer obtains a credential cre from the response message received from the issuer through the *createKeyFromCredential* process (TPM acquires a credential and Host verifies the credential).
- **Sign protocol:** TPM and Host sign a message *msg* with its own credential *cre*. That is, they work together to produce a signature σ of knowledge on some values: a secret value sk_T and a valid credential *cre* that was computed for the same value sk_T.
- **Verify protocol:** With signature σ, message *msg*, and an issuer public key ipk_k, as inputs, the verifier checks three things: (1) whether the signature σ proves knowledge of a secret value sk_T; (2) whether the signature proves knowledge of a valid credential issued on the same value of sk_T; (3) whether sk_T is not on the list of rogue values.

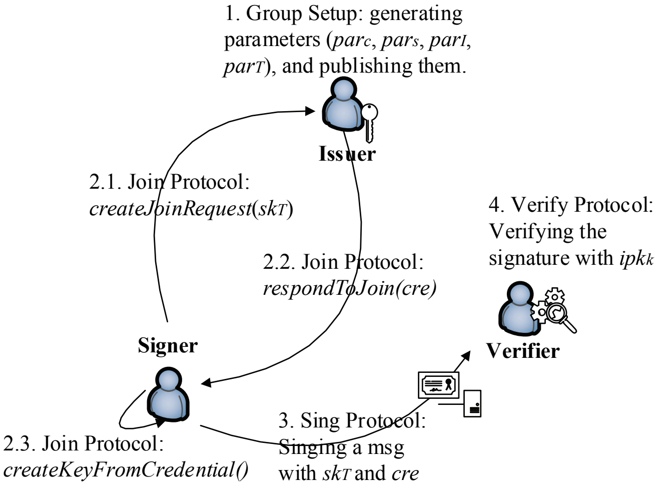
1. Group Setup: generating parameters (par_C, par_S, par_I, par_T), and publishing them.
Issuer
2.1. Join Protocol: *createJoinRequest*(sk_T)
4. Verify Protocol: Verifying the signature with ipk_k
2.2. Join Protocol: *respondToJoin(cre)*
Verifier
Signer
3. Sing Protocol: Singing a msg with sk_T and *cre*
2.3. Join Protocol: *createKeyFromCredential()*

Fig. 5. Processes of applied group signature scheme (pairing-based ECC-DAA).

Lastly, we explain the group management more detail. In our mechanism, members in a group do not have a special relationship like e-voting or e-bidding. The bigger the group size is, the larger the degree of anonymity is but the bigger the computation overhead (i.e., revocation) is [20]. Therefore, the proper group size should be considered according to the capacity of servers or user devices. If a member wants to join and a group being not full exists, previously described Join protocol would be performed. However, if all existing groups are full, Setup protocol to generate a new group should be first performed.

The Revocation protocol should be provided for the leave process of existing member, i.e., private key revocation, blacklist revocation, signature revocation, and credential revocation. ECC-DAA [28] based on our mechanism supports four revocations [20].

5. Performance Evaluation

We implement oIMP and evaluated our system in a MCS testing environment consisting of AWS EC2 [30], a workstation with high capacity, a laptop, and an embedded board. The AWS EC2 and the workstation are considered as a MCS server. The laptop and the embedded board (i.e., Intel Edison board [31]) are regarded as sensor providers. We evaluate the performance of oIMP in terms of setup time, authentication time, and scalability with the following processes: group creation on server, temporal ID generation on client, join on client and server, sign on client, and verification on server. Moreover, we analyze our oIMP from the view point of security.

5.1 Testing Environment

Table 2 summarizes each specification for each system in our testbed. We utilized the workstation as a high-capability server and AWS EC2 as a low-capability server. Because we used a free account for AWS EC2, the assigned specification was not very high, as shown in Table 2. For clients, we employed the laptop as a high-capability client and Intel Edison board as a low-capability client. We implemented our system by utilizing the published group signature code designed by the authors in [20,28,29]. We also used SHA with 256 and 512 key sizes.

Table 2. System specification for our testbed

Role	System	Specification description
High Spec. Server (H-Srv)	Workstation	Model: Dell Precision Tower 3420 CPU: Intel Core i7-6700 CPU @ 3.40 GHz Memory: 8014 MB Storage: 106 GB OS: Linux Ubuntu v.4.2.0-27-generic NIC Speed: 1 Gbps LAN
Low Spec. Server (L-Srv)	AWS EC2	Model: t2 micro service (free) CPU: Intel Xeon Processors @ 3.30 GHz Memory: 1 GB Storage: elastic block storage, 5 GB OS: Linux Ubuntu v.3.13.0-68-generic
High Spec. Client (H-Clt)	Laptop	Model: Dell Inspiron 15 5000 Series CPU: Intel Core i5 processor @ 2.80 GHz Memory: 8 GB Storage: 1 TB NIC Speed: IEEE 802.11ac OS: Linux Ubuntu v.4.4.0-22-generic
Low Spec. Client (L-Clt)	Embedded board (Intel Edison)	Model: Intel Edison Grove Starter Kit Board CPU: Genuine Intel CPU 4000 @ 500 MHz Memory : 983100 kB Cache : 1024 kB NIC Speed: 56 Mbps WiFi OS: Linux 3.10.17-poky-edison+ i686

5.2 Experimental Results and Analysis

We measured the process times on two processes: 'Initialization' and 'Signing and verification'. In this evaluation, we did not evaluate the 'On-demand Incentive Payment' process because it requires a simple process time on a server with round trip time. All results were computed by averaging the values of 30 emulation runs for statistical results.

5.2.1. Experimental results for initial setup

In the initialization stage of our system, a server is required to create groups for the verification of group signatures. Then, clients request to join the server and obtain a key for signing from response messages. Finally, Clients generate their temporal IDs with hash function.

- **Group generation time**: First, the server creates groups for group signatures. The group size determines the degree of anonymity because anonymity resides in the verifier not identifying the exact members in a group [20]. After clients subscribe (join) to the server, the clients obtain credentials for signing messages. Finally, the clients generate their temporal IDs. Fig. 6 shows the group creation time on servers when the number of groups increases from 5 to 100. In the case of a small number of groups, up to 30 groups, both servers have similar creation times. However, from 35 groups, the group creation times begin to show a big difference as the group size increases. The creation time of the low-capability server (L-Srv) is greater than that of the high-capability server (H-Srv). The time of L-Srv increases linearly with the number of groups.

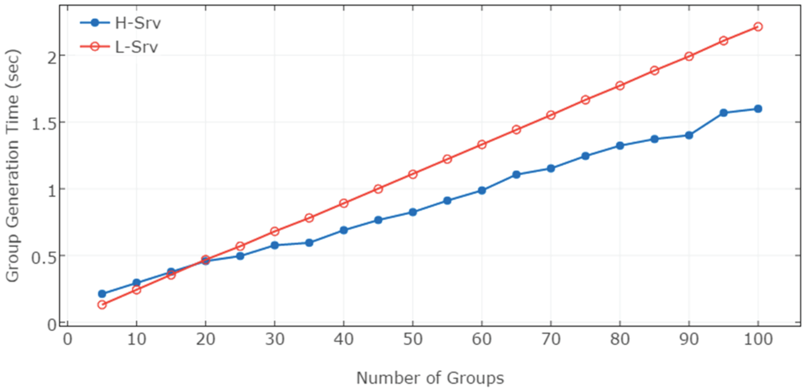

Fig. 6. Group generation time vs. number of groups.

- **Joining time:** The joining process consists of five sub-processes with three main function calls: (1) *createJoinRequest* on client, (2) request transmission from client to server, (3) *respondToJoin* on server, (4) response transmission from server to client, and (5) *createKeyFromCredential* on client. In order to compare each sub-process time for joining, we measured each function call to separately evaluate each sub-process for a joining process. Fig. 7 shows the times of each sub-process for joining on servers and clients. We also show the transmission times—i.e., related with sub-processes (2) and (4)—as a round trip time (RTT) between client and server. As a result, the time for *respondToJoin* on server is the greatest among three functions and it can decrease by utilizing a high-capacity server. Because there is a long distance between L-Srv (i.e., AWS) and L-Cli (i.e., Intel Edition Board), the transmission time (RTT) is higher than the RTT between H-Srv and H-Cli.

In general, a server may perform concurrent join processes because of many joining requests from clients. To show the concurrent *respondToJoin* processing time on a server, we increased the number of *createJoinRequests* from 5 to 100, as shown in Fig. 8. Even though the processing time for *respondToJoin* is quite large, the time increases linearly. For example, H-Srv requires 0.054 seconds per *respondToJoin* for 5 joining requests, but it has a similar value, 0.051 seconds per *respondToJoin* for 100 joining requests. This means that increases in joining requests does not affect the performance of the H-Srv.

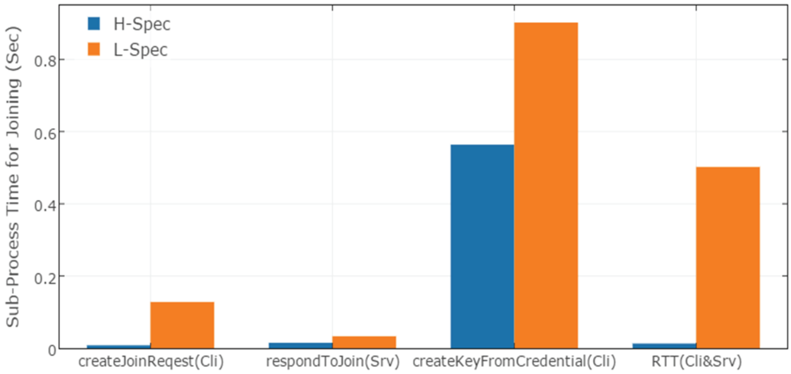

Fig. 7. Processing time for each sub-process for joining.

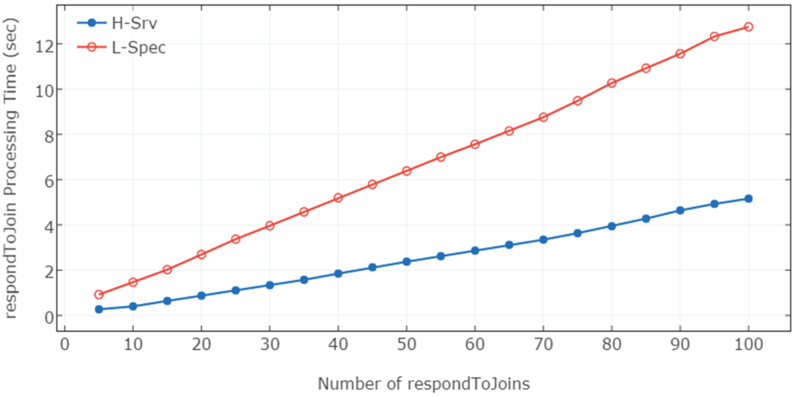

Fig. 8. Processing time for *respondToJoin* on server for concurrent joining requests.

- **Generation time for temporal IDs**: After clients obtain their key from their credentials during the joining process, they generate their temporal IDs using a hash function and a random seed. We simulated the generation of 100 temporal IDs with two well-known hash functions: SHA-256 and SHA-512, as shown in Fig. 9. Because of the longer key size for SHA-512 than SHA-256, the generation time for temporal IDs with SHA-512 is higher than that for SHA-256. However, the 100 temporal IDs were generated within only 0.053 seconds.

The above experimental results are related to an initial setup time that is required once. With the exception of the joining process, these initial setup processes can be performed as background processes. The server time for the joining process can be decreased by utilizing a H-Srv.

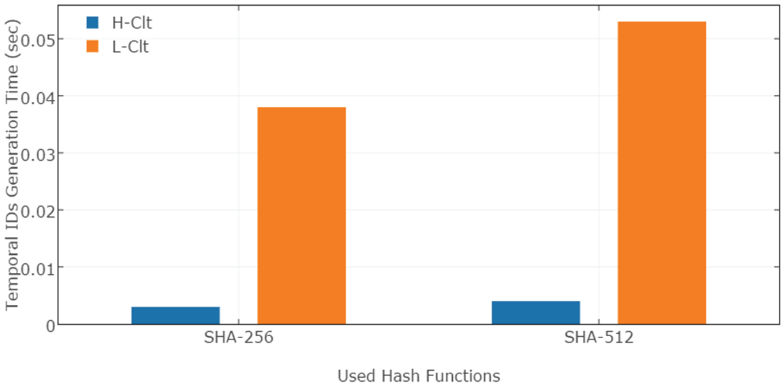

Fig. 9. Generation time for 100 temporal IDs vs. used hash functions.

5.2.2 Results for signing and verification

The clients (i.e., sensor providers) provide their sensing information with signatures to a server. The signatures are signed with the keys created from the credentials obtained during the joining process. The server can verify whether the sensor providers are members of a group. The signing process time may depend on the message length and client capacity, and thus we vary the message length, as shown in Fig. 10. The Signing and verification processing time for long messages increase, but amount of increase is very small. In the case of the high-capability system, the verification time is greater than the signing time. However, in the case of the low-capability system, the signing time increases considerably because it is affected by the very low specification Intel Edison board. Our conclusion is that the processing times for the signing and verification are not significantly influenced by message sizes. Further, the verification time does not affect the acknowledgement of signing or incentive payments. Clients only inform the server of their sensing information, and the sensing information only need be verified before the information is provided to other consumers.

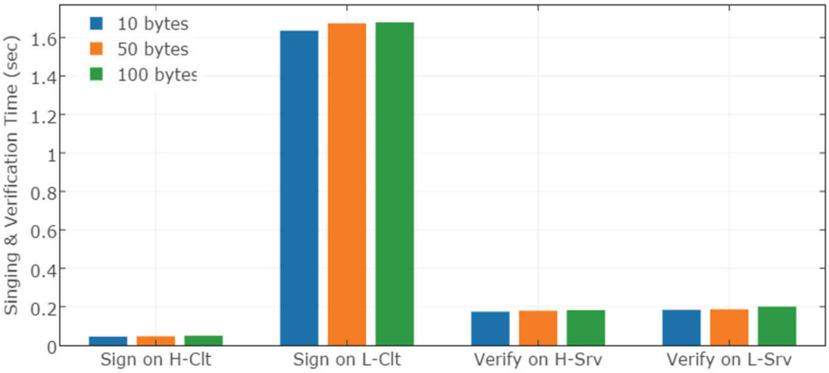

Fig. 10. Signing and verification time vs. message size.

In general, servers may receive many sign messages concurrently. Thus, we increased the number of concurrent sign messages and measured the verification time on a server, as shown in Fig. 11. As expected, the increase is slightly linear to the number of verified messages. In particular, H-Srv gives a more marginal increase than L-Srv. For example, H-Srv requires 0.076 seconds per verification for 5 messages and it decreases to 0.046 seconds per verification for 100 messages.

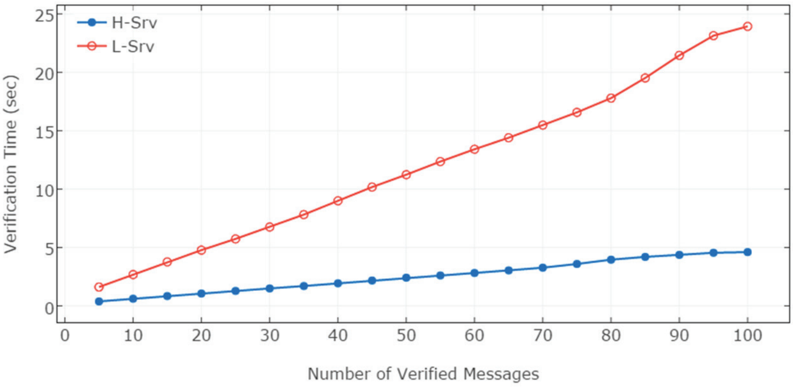

Fig. 11. Verification time vs. number of verified messages.

5.3. Security Analysis

In this subsection, we discuss the security aspects of our mechanism. Many location privacy preservation protocols for LBSs have utilized obfuscation, *K*-anonymity, or cloaking approaches which use inaccurate location or delay the service [22-24]. However, those general approaches are difficult to be applied to our scenario, crowdsensing service on smart parking system, because sensing data provider on our assumed service is required to provide sensing data for parking spaces with exact location and real-time data is useful.

Instead, we utilized group signature and temporal ID to authenticate provider with anonymity, when the provider publishes the sensing data to server. Group signature generally provides the following security properties: non-members cannot sign, signatures are unlinkable and anonymous, and the group manager cannot falsely accuse member [20]. That is, at the time of authentication, the location of provider would not be revealed. Only the past location information that the provider published the sensing data would be exposed when she requests the reward for data providing until now. However, the exposed locations are not current ones.

The location of consumer could be safe utilizing by revealing obscure location information when consumers request the service [22], by anonymizing the consumers in a group [23], or by using cloaking areas instead of the exact coordinates [24]. In cases of obfuscation or cloaking, the consumers just need to filter the parking lots information provided from server to find the necessary parking spaces information. Anonymizing approach can be used in the case of periphery consumers are generally many.

6. Conclusion

Solving the conflicting problem of privacy preservation and incentive payment in MCS is not a trivial task. We proposed an on-demand incentive payment mechanism with privacy preservation, especially location and identity privacy, called oIMP. The proposed mechanism authenticates the provided sensing information through group signatures, and pays the incentive on demand manner. It is possible to preserve the privacy that our mechanism separates the authentication time for providing sensing information and the incentive payment time. In the incentive payment, the revealed information is only past information, even though the server has stored the location information. Our emulation evaluation showed the operation and feasibility of our mechanism and verified the system performance. As future work, we will implement a prototype of oIMP for the intelligent parking system and create a general framework to provide an incentive mechanism that preserves privacy. Moreover, we will show the superiority of our privacy-preservation with the incentive payment on the intelligent parking system, comparing with the existing approaches.

Acknowledgement

This research was supported by Science Research Programs through the National Research Foundation of Korea (NRF) funded by the Korea government (the Ministry of Education and MSIT) (grant numbers 2015R1D1A1A01057362 and 2018R1A2B6009620).

References

[1] D. Evans, “The Internet of Things: how the next evolution of the internet is changing everything,” Cisco Internet Business Solutions Group (IBSG), San Jose, CA, 2011.

[2] M. Kanellos, “What's The Big Data?,” 2016; https://whatsthebigdata.com/2016/03/07/amount-of-data-created-annually-to-reach-180-zettabytes-in-2025/.

[3] A. Botta, W. De Donato, V. Persico, and A. Pescape, “On the integration of cloud computing and Internet of Things,” in *Proceedings of 2014 International Conference on Future Internet of Things and Cloud*, Barcelona, Spain, 2014, pp. 23-30.

[4] X. Sheng, J. Tang, X. Xiao, and G. Xue, “Sensing as a service: challenges, solutions and future directions,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3733-3741, 2013.

[5] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32-39, 2011.

[6] X. Jin and Y. Zhang, “Privacy-preserving crowdsourced spectrum sensing,” *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 3, pp. 1236-1249, 2018.

[7] E. Macias, A. Suarez, and J. Lloret, “Mobile sensing systems,” *Sensors*, vol. 13, no. 12, pp. 17292-17321, 2013.

[8] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, “A location-based incentive mechanism for participatory sensing systems with budget constraints,” in *Proceedings of 2012 IEEE International Conference on Pervasive Computing and Communications*, Lugano, Switzerland, 2012, pp. 103-108.

[9] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, “Quality-driven auction-based incentive mechanism for mobile crowd sensing,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203-4214, 2015.

[10] V. S. Pulla, C. S. Jammi, P. Tiwari, M. Gjoka, and A. Markopoulou, “QuestCrowd: a location-based question answering system with participation incentives,” in *Proceedings of 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Turin, Italy, 2013, pp. 75-76.

[11] Q. Xie and L. Wang, “Privacy-preserving location-based service scheme for mobile sensing data,” *Sensors*, vol. 16, article no. 1993, 2016.

[12] G. Danezis, S. Lewis, and R. J. Anderson, "How much is location privacy worth?," in *Proceedings of the 4th Annual Workshop on the Economics of Information Security (WEIS)*, Cambridge, MA, 2005.

[13] A. Singla and A. Krause, "Incentives for privacy tradeoff in community sensing," in *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing*, Palm Spring, CA, 2013.

[14] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *Proceedings of 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, Nara, Japan, 2016, pp. 344-353.

[15] J. Sun and H. Ma, "Privacy-preserving verifiable incentive mechanism for online crowdsourcing markets," in *Proceedings of 2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, Shanghai, China, 2014, pp. 1-8.

[16] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "Security, privacy, and incentive provision for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839-853, 2016.

[17] X. Li, M. Miao, H. Liu, J. Ma, and K. C. Li, "An incentive mechanism for K-anonymity in LBS privacy protection based on credit mechanism," *Soft Computing*, vol. 21, no. 14, pp. 3907-3917, 2017.

[18] X. Niu, M. Li, Q. Chen, Q. Cao, and H. Wang, "EPPI: an e-cent-based privacy-preserving incentive mechanism for participatory sensing systems," in *Proceedings of 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC)*, Austin, TX, 2014, pp. 1-8.

[19] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "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, 2016.

[20] K. Potzmader, J. Winter, D. Hein, C. Hanser, P. Teufl, and L. Chen, "Group signatures on mobile devices: practical experiences," in *Trust and Trustworthy Computing*. Heidelberg: Springer, 2013, pp. 47-64.

[21] M. Kim, "Incentive mechanism with privacy-preservation on intelligent parking system utilizing mobile crowdsourcing," in *Proceedings of 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, Kuta Bali, Indonesia, 2017, pp. 1-4.

[22] R. Huang, B. Ying, and A. Nayak, "Protecting location privacy in opportunistic mobile social networks," in *Proceedings of 2018 IEEE/IFIP Network Operations and Management Symposium*, Taipei, Taiwan, 2018, pp. 1-8.

[23] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, "ILLIA: enabling k-anonymity-based privacy preserving against location injection attacks in continuous LBS queries," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1033-1042, 2018.

[24] B. Ying and A. Nayak, "Social location privacy protection method in vehicular social networks," in *Proceedings of 2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, Paris, France, 2017, pp. 1288-1292.

[25] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptology-EUROSCRIPT'91*. Heidelberg: Springer, 1991, pp. 257-265.

[26] M. E. Andres, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS'13)*, Berlin, Germany, 2013, pp. 901-914.

[27] V. Primault, S. B. Mokhtar, C. Lauradoux, and L. Brunie, "Differentially private location privacy in practice," in *Proceedings of the 3rd Workshop on Mobile Security Technologies (MoST)*, San Jose, CA, 2014.

[28] L. Chen, D. Page, and N. P. Smart, "On the design and implementation of an efficient DAA scheme," in *Smart Card Research and Advanced Applications*. Heidelberg: Springer, 2010, pp. 223-237.

[29] ISO20008-2.2 Group Signature Scheme Evaluation on Mobile Devices, https://github.com/klapm/group-signature-scheme-eval/.

[30] Amazon EC2 Instance Types, https://aws.amazon.com/ec2/instance-types/?nc1=h_ls.

[31] Intel IOT Developer Kit, https://software.intel.com/en-us/iot/hardware/devkit.

Mihui Kim https://orcid.org/0000-0002-4896-7400

She received the B.S. and M.S. degrees in Computer Science and Engineering from Ewha Womans University of Korea, in 1997 and 1999, respectively. During 1999–2003, she stayed in Electronics and Telecommunications Research Institute (ETRI) of Korea. She also received the Ph.D. degree in Ewha Womans University in 2007. She was a postdoctoral researcher of the department of computer science, North Carolina State University from 2009 to 2010. She is currently an associate professor of the Department of Computer Science and Engineering, Hankyong National University in Korea. Her research interests include security and efficient protocol design in IoT and crowd sensing system.

Younghee Park https://orcid.org/0000-0003-0651-2384

She is an assistant professor in Computer Engineering of San Jose State University. Before joining this department, she was a post-doctoral researcher in the University of Illinois at Urbana-Champaign. She was also a post-doctoral research scientist in the Computer Science at Columbia University in New York City in 2011. She received her Ph.D. in Computer Science from North Carolina State University in 2010. Her main research is network and system security with an emphasis on malware detection, insider attacks, botnets, trace back to detect attacks. Currently, she is focusing on security issues in SDN and mobile cloud computing.

Pankaj Balasaheb Dighe https://orcid.org/0000-0003-2734-8683

He received B.S. in Computer Engineering of University of Pune in 2011, and M.S. degree in Computer Engineering of San Jose State University in 2016. He is working as Software Engineer at Autodesk. He has his specialties in Java, JavaScript, Salesforce, and Amazon Web Services