Privacy Enhanced Location Sharing for Mobile Online Social Networks

Junggab Son, Donghyun Kim, *Senior Member*, IEEE, Md Zakirul Alam Bhuiyan,
Rahman Tashakkori, *Senior Member*, IEEE, Jungtaek Seo, Dong Hoon Lee, *Fellow*, IEEE

Abstract—As a primitive function of location-based services (LBSs), the location sharing aims to provide a user's current location information to other designated users. In recent years, LBSs have become one of the most popular services provided by mobile online social networks (mOSNs). As LBSs actively exploit the users' identity and current location information, appropriate approaches have to be utilized to protect the location privacy of the users. Several recent reports have discussed the significance of friendship privacy protection with the goal of hiding the friendship relation of users from unintended entities. However, to the best of our knowledge, there hasn't been an approach for protecting the location sharing with complete privacy of location and friendship connections. To address this issue, we propose a new cryptographic primitive, functional pseudonym, for location sharing in mOSNs that ensures both of them. Unlike many of the existing solutions, our approach does not require a fully trusted server and does not assume pre-established secrets among friends, and therefore is highly practical. Also, the proposed approach significantly reduces computational overhead of users by delegating part of the computations for location sharing to a server, therefore is highly sustainable. Our primitive can be widely used in many mOSNs to enable LBSs with improved privacy and sustainability. Consequently, it will contribute to proliferate LBSs by eliminating users privacy concerns.

Index Terms—Location sharing, location privacy, friendship privacy, mobile online social networks, functional pseudonyms.

1 Introduction

The recent years have witnessed the adoption of a variety of Internet of Things (IoT) end-user devices such as smartphones, tablets, smartwatches, etc., which can serve as a platform for mobile online social network (mOSN) applications. Today, mOSNs have become popular as they can provide a medium for exchanging various information such as opinions, believes, knowledge, current status, and location with designated recipients in a timely manner regardless of their current time and location [2], [3], [4]. Location sharing is the most important primitive to realize the promise of location based services (LBSs) over mOSNs such as local recommendation service, tracking children, proximity notification among users [5], [6], [7], [8]. Despite the advantages, LBSs are not widely used because of users' privacy and trust concerns as they actively expose their identity and current location information [9], [10]. Clearly, a LBS without an appropriate privacy protection mechanism could lead users into unpleasant situations such as being in an unwanted location resulting from fake advertisements or spams. Also, this can result in damaging the users' social reputation, financial loss, and being a victim of blackmail and target of stalking or physical violence [11], [12].

In recent years, numerous approaches have been proposed to preserve user privacy for location sharing. SmokeScreen [13], SMILEs [6], [14], and MobiShares [15], [16], [17], [18] utilize privacy-preserving mechanisms for this purpose which are not suitable for mOSNs. SmokeScreen uses an unreasonable assumption of a trusted third party or pre-established secret for each pair of users to help the proximity notification among users. SMILEs require users, who would like to detect the presence of desired partners, to be in the same geographic location at least once beforehand in order to share some secrets. Since in mOSNs, many of the online friends do not have a prior physical contact with each other and can establish a relationship with strangers and share location information any time, such an assumption is not realistic. In MobiShares, the location service providers can figure out the users' spatiotemporal relationship by linking their queries. Frequently, LBSs rely on mOSNs for a user to conveniently obtain information on designated partners. Unfortunately, this makes the mOSN server to learn about the relationship among users and is of another great privacy concern, which is known as friendship privacy problem [19], [20], [21], [22]. Our investigation of the related literature indicates that there is no location-sharing primitive with both location and friendship privacy protections signifying the research presented on this paper.

For the smooth operation of mOSNs, users need to exchange their information regarding establishing a friendship and sharing location information with friends [23]. However, the leakage of the information will directly affect their privacy. Thus, an effective way for secure information

The preliminary version of this paper was published in the proceedings of ICCCN'16 [1]

- *J. Son and D. Kim are with the Department of Computer Science, Kennesaw State University, Marietta, GA, 30060. E-mail: {json,donghyun.kim}@kennesaw.edu.*
- *M.Z.A. Bhuiyan is with the Department of Computer and Information Sciences, Fordham University, Bronx, NY, 10458. E-mail: mbhuiyan3@fordham.edu.*
- *R. Tashakkori is with the Department of Computer Science, Appalachian State University, Boone, NC, 28608. E-mail: tashakkorir@appstate.edu.*
- *J. Seo is with the College of Information Security, SoonChunHyang University, CheonAn, South Korea. E-mail: seojt@sch.ac.kr.*
- *D.H. Lee is with the Graduate School of Information Security, Korea University, Seoul, South Korea.*
  E-mail: donghlee@korea.ac.kr, Corresponding Author.

exchange is needed. A fully trusted server would be a straightforward solution for allowing users to exchange their information securely, but this is generally treated as a strong assumption in that service providers can interfere [24], [25]. In addition, assuming that two users have a pre-established secret is inefficient and does not make sense in mOSN where users freely establish friendship with strangers. Therefore, providing location-sharing services in mOSNs while preserving two types of privacy creates a problem, thus an efficient and effective solution is required to address the problem.

Contribution of This Paper We propose a new cryptographic primitive for location sharing in mOSNs, called *functional pseudonym*, which design goals are to achieve:

- Location privacy protection: The current location of a user must not be tracked by any unintended entity, including the service provider.
- Friendship privacy protection: The friendship information of a user must not be revealed from unintended entities even with a large set of data related to location sharing.

To make it more practical, we develop the proposed scheme with the following three requirements, which are less demanding than those from the existing work.

- Semi-honest-but-curious server [26]: A server is an ability-limited active attacker which performs some of the pre-defined attacks only while following given protocols correctly.
- No pre-established secret: Users establish friendship relations through the server without any pre-established secret.
- Delegation of computation: The process of location sharing should be delegable to support resource-constraint mobile devices and to achieve high sustainability.

Organization of This Paper. This paper is organized as follows. Section 2 introduces related work, Section 3 provides the system model, problem description, adversary models, and security goals that considered in this paper. We describe our main contribution, privacy enhanced location sharing based on the functional pseudonym, in Section 4. Section 5 provides the security analysis of the proposed scheme, and Section 6 provides the efficiency analysis of the proposed scheme. In Section 7 we demonstrate that the proposed scheme is suitable for mobile usage. Finally, we provide the conclusions for this paper in Section 8.

2 Related Work

Privacy problem is one of the emerging issues in location-based services. Some applications treat all users as trusted entities, and thus they provide users' identities as well as location information without a protection scheme [27], [28]. By collecting the location information, an attacker can infer various information such as a moving route, a position in a certain time, a current location, and a prediction of future location.

In order to address this problem, anonymity schemes have been proposed. Gruteser et al. proposed a location privacy mechanism through spatial and temporal cloaking [29]. To hide the location information based on the time, this scheme uses an approximate time and location information instead of a real value. Since this approach has a trade off between achieving accurate location information and user's privacy, it is challenging to provide an application that offers perfect privacy. Jiang et al. resolved this problem using a pseudonym with silent time [30], however their approach has a drawback as the service provider is assumed to be a trusted entity. Also, since the system has no effective solution to protect users' location privacy, it could easily reveal the information to the attackers.

Although user privacy in location-based applications is important, information is still needed to provide better services [31]. For example, users must disclose their location information to get a location-based service. Users can get more precise services if they disclose more information, however privacy problem will also get worsen. In order to satisfy this condition, several schemes were proposed to provide a location information in limited circumstance. In 2007, SmokeScreen was proposed to provide flexible presence-privacy controls for presence-sharing on applications with the identities of co-located users, while user's location information is never revealed without the explicit permission [13]. SmokeScreen also enables presence-sharing among trusted social relationships, as well as untrusted strangers, through trusted brokers which coordinate anonymous communications between them. The MobiShare proposed by [15] enables a flexible location sharing between trusted social relations as well as untrusted strangers. MobiShare is vulnerable to attacks generating fake identities, thus a location information will be potentially leaked to a service provider. In order to address this problem, J. Li et al. proposed MobiShare+ which uses dummy queries and a private set intersection protocol to prevent leakage of location information from the service provider in OSN [16]. However a social relation can be easily exposed by a service provider, and location information can be easily exposed by eavesdropping wireless communication section between mobile devices and cellular tower.

In 2009, Manweiler et al. proposed SMILE, a privacy-preserving "missed-connections" service, establishing a connection between users who do not have a pre-established social relationship through an untrusted service provider [14]. In SMILE, the trust is established based on the shared encounters that has passively exchanged with the nearby peers. However, to share encounters, users must be located at the same place and at the same time for at least once.

The location-based services were extended to geosocial networks [6]. With location-aware capabilities, a geosocial network can offer different types of services, such as location sharing, tracking friends, and local recommendation services. Also, a proximity service was proposed that would alert the user when any of his/her friends would come into the specified geographical range of the user [8], [32], [33]. However, the convenient functionality comes with privacy problems which include exposing location, absence, co-location, and identity that are all sensitive information [7].

In the application proposed on this paper, two different types of privacy issues exist. The first issue is location privacy, and an approach that an attacker can only obtain location information without identity or identity without precise location was considered [8], [33]. The second issue is identity privacy and a quasi-identifier scheme was used to deal with this type of privacy [34], [35]. Mascetti et al. proposed a proximity service with complete privacy [8]. When a proximity service satisfies both the location privacy and identity privacy, it is said to support complete privacy. Their scheme assumes untrusted service providers and curious buddies. For this, Mascetti et al. proposed two new protocols: providing complete privacy with respect to a service provider, and controllable privacy with respect to friends. However, that is not suitable for the system model presented on this paper because they assume that the user's friends are pre-determined, while this paper sssumed no pre-established trust. In addition, their approach has the drawback of being challenging for the user to obtain a precise location of friends.

In recent years, the privacy issues in mOSNs have been among active topics and has attracted researchers' attention, thus many interesting solutions have been proposed to address them. Tang et al. proposed a verifiable location query based on homomorphic encryption with a searching index and a trapdoor [36]. Sun et al. proposed a privacy-preserving LBS scheme to deal with the problem in which a friend can be an attacker [37]. Peng et al. employed a new entity, function generator, which periodically distributes location parameters [38] that is used to allow users and a service provider transform a real location into a pseudo-location. This contributes to eliminate a trusted entity from the system. Schlegel et al. also proposed a new encryption scheme to eliminate a trusted entity and to provide efficient query processing [39]. However, none of these schemes was suitable to preserve users' privacy in mOSNs as they have a limitation of either (a) a fully trusted entity or (b) did not consider the friendship privacy. Li et al. proposed a location-sharing system that could partially preserve the friendship privacy in a way that users separately make partial lists of friends and send them to multiple servers [40]. Since their scheme could not guarantee the friendship privacy completely, it is difficult to say that it is a suitable solution for our environment. Therefore, a better solution with enhanced privacy is required for the location sharing in mOSNs.

3 Problem Description

This section describes the system model, problem definition, adversary models, and security goals that are considered on this paper.

3.1 System Model

The proximity notification service is a well-known and widely used concept which notifies users of other nearby users with their distances [41]. The location sharing in mOSN is a similar concept to the proximity notification, but the crucial difference exists that the LBS in mOSN can use social information to establish a location information sharing group.

Figure 1 illustrates the system model considered on this paper. According to the location-based social networks (LBSNs) classification [11], our system model can be classified as category I: LBSNs with Exact Location Sharing, and Sub-type II: User Authorized Location Sharing in which users have the control to choose with whom they wish to share their exact location information. In a nutshell, the system model consists of two entities: a server and users. The server provides geosocial services including location sharing, and users can establish a real-time connection with other users and share various information among friends through the server. We define the location sharing as a sub-service or a sub-application of a general SNS, where a user allows friends to access the location information in a real time manner. Users mostly use the SNS provided by the server, and whenever they want to share location information with friends, they operate the sub-application with a pseudonym as an anonymous identity to ensure privacy preservation.

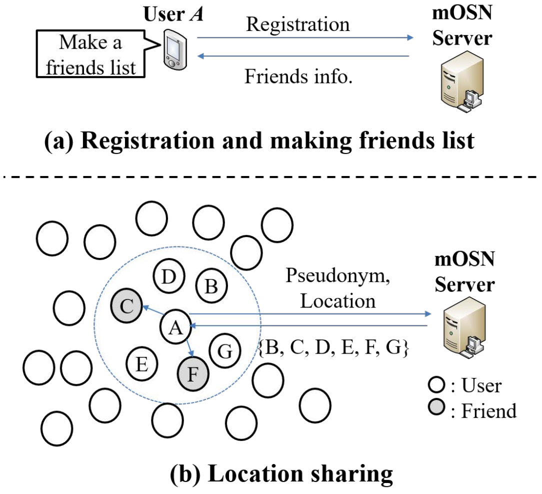


Fig. 1: System Model

As shown in Fig. 1 (a), every user should be registered to the sub-application first. Each user then makes a list of designated friends as a subset of friends list. Then, only users in the designated friends list can access the location information. Since we need to mention the designated friend in the rest of this paper, the term "friend" is used instead of "designated friend" from this point on. Suppose there is a registered user U_i, and another user U_j who belongs to the friend list of the U_i. The U_j can identify which pseudonyms are generated by U_i and check U_i's current location. In this case, we say that U_i and U_j have a friendship relation. Fig. 1 (b) illustrates the process of finding nearby friends. If the user uploads the friendship list along with location information to utilize the location sharing service, the server sends a set of users' information in the user's requested range. The user also can broadcast the information to find the nearby friends. However, it is more desirable that the user gets a nearby user's information through the server for several reasons that includes loss of signal at a location. This also delegates computation to the server and hides information from unintended users [42].

3.2 Problem Definition

Given a set of tuples that are comprised of a pseudonym and location information $\{(P_1, L_1), (P_2, L_2), \ldots, (P_n, L_n)\}$ within a pre-defined range, the problem addressed in this paper lies in finding a subset of tuples $\{(P_i, L_i)\}_{1 \leq i \leq n}$

generated by friends while preserving two types of privacy: location privacy and friendship privacy. In addition, such a location sharing scheme should be done with neither a pre-established secret among friends nor through a fully trusted server for practical reasons.

One simple yet effective way to preserve identity privacy, as well as location privacy, is adopting an anonymous communication scheme based on pseudonyms. By employing pseudonyms which are random strings, instead of a real identity, users' identity could not be revealed to unintended entities. At the same time, this approach could preserve location privacy even when the users' location information is provided. In such a case, adversaries could obtain information that someone is at a certain location but could not know exactly who that person is.

The anonymity helps preserve the privacy, however, it disrupts a smooth exchange of information to establish a friendship relation and to find nearby friends. To improve this, users must provide some information for the location sharing which can be an ammunition for the invasion of privacy. Therefore, there is a issue associated with providing information and preserving privacy.

3.3 Adversary Models

On this paper, an unintended entity is whoever do not have a friendship relation and access permission to the location information, including a service provider as a potential adversary. Similar to [11], the adversaries in this study have limited capability that they can only access the publicly available information and have the same computing power as normal users in mOSNs. They have an ability to eavesdrop messages transferring between a server and users. We assume a semi-honest-but-curious model for the server [26], which follows the given protocols correctly but may perform some predefined attacks only. Specifically, we consider the following attacks that can be attempted by adversaries:

(a) **Attacks on pseudonym**: From a given set of pseudonyms, an attacker may try to obtain a piece of information on a user's identity. Also, the attacker may try to distinguish a subset of pseudonym issued by the same user. This information may be abused to track the user and/or predict his/her future location. This attack is directly linked to the invasion of location privacy.

(b) **Attacks on location information**: Some of the previous work show the possibility of location discovery and tracking, which are applicable to our system model. In [41], attackers can expose the users' location information by trilateration attacks. Also, an automated location tracking system has been developed by [11]. We assume that an adversary could use these tools.

(c) **Attacks on friendship relation**: By persistent monitoring, the attacker can obtain information about a friendship relationship between two users. Furthermore, the attacker can infer a user's friendship list.

In this paper, we do not consider the adversaries hacking the server's database to obtain valuable information. We only consider a limited collusion attack such that adversaries can collude with other entities (e.g., server-user or user-user) to infer a third user's friend list or any other secrets, without directly exchanging their secret keys (e.g., a relation value). A collusion attack to exchange information other than the relation value, such as a pseudonym with/without location information, will not leak any information about users to adversaries.

3.4 Security Goals

From the observations of the system and adversary models, we define the following security goals:

(a) **Pseudonym indistinguishability**: an attacker cannot distinguish whether pseudonyms have come from the same user or not;

(b) **Identity/Location privacies**: an attacker cannot invade a user's location privacy regardless of the session. The user's current, past, as well as future location information cannot be obtained by the attacker. In addition, an attacker who is not a friend of a user cannot obtain the user's identity and location information such as GPS coordinates.

(c) **Friendship relation privacy**: an attacker cannot obtain any information about a friends list and a friendship relation from published pseudonyms, even from a large set of data.

4 The Proposed Functional Pseudonym based location sharing scheme

4.1 overview

A randomly generated string as pseudonym can be a simple yet effective solution to preserving users' privacy. However, the pseudonym provides almost no information, thus establishing a friendship relationship and sharing location information is almost impossible under this circumstance. To address this problem, we propose a new cryptographic primitive, named functional pseudonyms, which holds a functionality that makes it possible for users to exchange information for the location sharing purpose. They can distinguish whether a functional pseudonym was generated by a friend or not. At the same time, the functional pseudonyms still have randomness to provide anonymity. Beyond the functionality, we have designed the pseudonym changeable for the robustness against statistical disclosure and intersection attacks [43], [44]. Whenever a user tries to find nearby friends, he/she must pick a new random number using a pseudo-random generator (PRG) to generate a new functional pseudonym. We will prove the robustness of changeable functional pseudonym against the statistical attacks in Section 5.

Fig. 2 shows the flow of the proposed scheme which largely consists of two processes: functional pseudonym generation and location sharing. A user U_i who wants to share his/her location with another user U_j has to establish a friendship relation first. Without loss of generality, we assume that U_i can browse every other users' public information through the server. Obtainable information includes the user name, pictures, self-introduction messages, and so forth. In addition, we consider the location sharing as a sub-application, and every user generates a designated friend list from the set of friends in general SNS. For this reason, it is not a strong assumption that communication at this level

TABLE 1: Notations.

Notation	Description
q	k-bit prime number
$\mathbb{Z}_q$	Integers modulo q
$\mathbb{G}$	Cyclic group with prime order q
g	Generator of $\mathbb{G}$
e	bilinear pairing
U_i	User i
pk_i, sk_i	Public/private key pair of U_i
P_i	Pseudonym of U_i, $P_i = \{p_{i1}, p_{i2}, p_{i3}\}$
L_i	Current location information of U_i, e.g., GPS coordination
$H(\cdot), H_1(\cdot)$	A collision free hash functions

can be protected by a public key cryptosystem. Thus, each user securely exchanges a relation value, which will be used for location sharing, with the designated friends as if he/she is sending and receiving text messages. The exchanged relation values are computed by Lagrange polynomial, and the set of relation values will be expressed by a single value. On the basis of this single value, adding a randomly generated value makes the pseudonym indistinguishable under decisional Diffie-Hellman problem. To find U_i's current location, U_j (a friend of the user U_i) sends a pseudonym along with the relation values to the server. The server cannot obtain identities and relations from the values but can identify pseudonyms in friendship relations by applying Lagrange Interpolation. The identification results will be sent back to the U_j. Finally, the U_j can confirm the U_i's real identity and current location depending on the previously exchanged relation value.

In our preliminary work, we have developed a functional pseudonym scheme in [1], which preserves location privacy and friendship relation privacy and provides a functionality of the location sharing. However, it has a drawback of weak sustainability due to a high computational overhead as each user needs to compute n times of identity checking processes to find friends, where n is the number of adjacent users. To overcome this weakness, we propose a novel delegable functional pseudonym. A major advantage of our scheme is that heavy computations for location sharing can be computed by a service provider. This advantage contributes to significantly reducing the battery consumption on local devices or computers, hence resulting in higher sustainability.

4.2 Setup

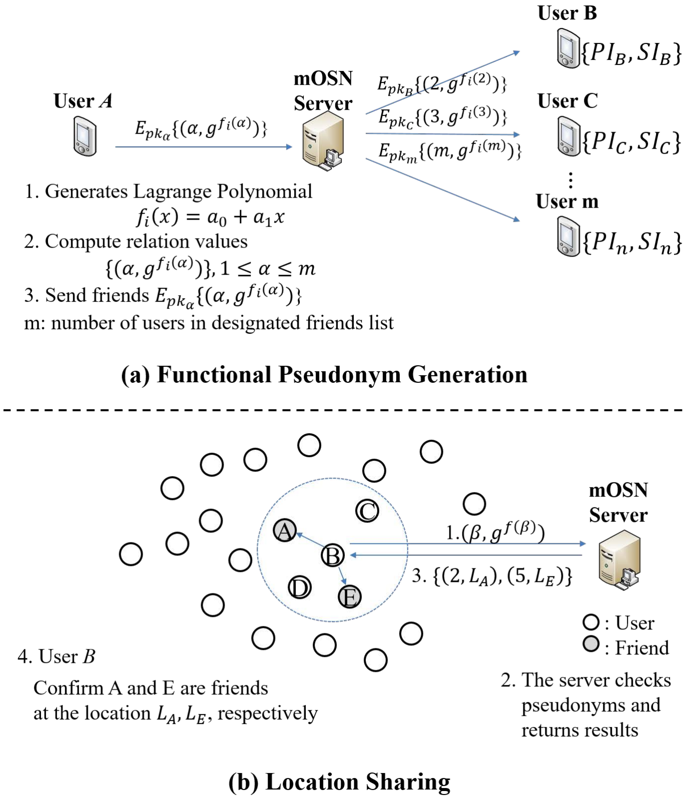
Table 1 illustrates the notations used in this paper. On input a security parameter 1^k, the setup process first determines a large prime q, $\mathbb{Z}_q$ which are represented by integers modulo q and a cyclic group $\mathbb{G}$. Also, the process selects a generator $g \in_R \mathbb{G}$ and a bilinear function e which can be defined as follows:

Definition 1 (Bilinear map)**.** *A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties [45], [46].*

(a) Computable: there exists an efficiently computable algorithm for computing e,

(b) Bilinear: for all $h_1, h_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$, and

(c) Nondegenerate: $e(g, g) \neq 1$, where g is a generator of $\mathbb{G}$.


User A
mOSN Server
User B
User C
User m
$E_{pk_\alpha}\{(\alpha, g^{f_i(\alpha)})\}$
$E_{pk_B}\{(2, g^{f_i(2)})\}$
$E_{pk_C}\{(3, g^{f_i(3)})\}$
$E_{pk_m}\{(m, g^{f_i(m)})\}$
$\{PI_B, SI_B\}$
$\{PI_C, SI_C\}$
$\{PI_n, SI_n\}$
1. Generates Lagrange Polynomial
$f_i(x) = a_0 + a_1 x$
2. Compute relation values
$\{(\alpha, g^{f_i(\alpha)})\}, 1 \leq \alpha \leq m$
3. Send friends $E_{pk_\alpha}\{(\alpha, g^{f_i(\alpha)})\}$
m: number of users in designated friends list

(a) Functional Pseudonym Generation

A
B
C
D
E
mOSN Server
1.$(\beta, g^{f(\beta)})$
3. $\{(2, L_A), (5, L_E)\}$
○ : User
● : Friend
4. User B
Confirm A and E are friends at the location L_A, L_E, respectively
2. The server checks pseudonyms and returns results

(b) Location Sharing

Fig. 2: Functional Pseudonym Generation and Location Sharing Protocol

The process then determines two cryptographic hash functions $H(\cdot)$ and $H_1(\cdot)$ that satisfy $H(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{G}_T$, respectively. For simplicity, we assume a symmetric bilinear map, however advanced bilinear maps can easily be applied to the proposed scheme. Finally, the global parameters are defined as $\{q, \mathbb{G}, \mathbb{G}_T, g, H(\cdot), H_1(\cdot), e\}$. Once the global parameters are available, each user generates a public/private key pair. The user U_i then picks $sk_i \in_R \mathbb{Z}_q^*$ as his/her private key, and computes g^{sk_i} as public key. The key generation algorithm outputs the public/private key pair (pk_i, sk_i).

4.3 Establishment of a Friendship Relation

Suppose the user U_i tries to include one of his/her friends U_j in the designated friend list for the location sharing purpose. The U_i and U_j should exchange relation values with each other. The relation value can be generated using (t, n) secret sharing scheme which splits a secret into n separate values and reconstructs it again when minimum t number values are collected [47]. Recently, several advanced secret sharing schemes have been proposed [48], however for simplicity, we use Shamir's secret sharing [49]. Since we only need to check who the friend of U_i is, 2 is a sufficient factor for t. The Lagrange interpolation polynomial, which is a fundamental theory of the secret sharing, can be defined as follows:

Definition 2 (Lagrange Interpolating Polynomial)**.** *The Lagrange interpolation polynomial [49] is the polynomial*

$f(x)$ of $\tau - 1$ degree that passes through the τ points $(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)$, and is given by

$$f(x) = \sum_{\ell=1}^{t} y_\ell \cdot \Delta_{x_\ell, S}(x),$$

where $\Delta_{x_\ell,S}(x)$ is Lagrange coefficient and a set S of elements in $\mathbb{Z}_q$:

$$\Delta_{x_\ell,S}(x) = \prod_{x_m \in S, m \neq \ell} \frac{x - x_m}{x_\ell - x_m}.$$

To generate the relation value, U_i randomly picks $a_0, a_1 \in_R \mathbb{Z}_q^*$. Here a_0 is a y-intercept, which will also be the secret to be split, and a_1 is a coefficient for the Lagrange polynomial:

$$f_i(x) = a_0 + a_1 x.$$

U_i computes tuples $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m))\}$ based on $f_i(x)$, where m is the number of friends. U_i then randomly picks a pair $(\alpha, f_i(\alpha))$, computes $g^{f_i(\alpha)}$, and sends the tuple $I_j = (\alpha, g^{f_i(\alpha)})$ to U_j, which is a friend of U_i, through a secure channel, where $2 \leq \alpha \leq m$. Note that $f(0)$ is a_0 and $f(1)$ will be used as a pseudonym verification factor in further process. U_j then stores the tuple along with an identification value of the U_i (e.g., name). These values will be used to find U_i. In addition, U_j performs the same process to allow U_j to find his/her location.

4.4 Pseudonym Generation

In most privacy-preserving applications, pseudonyms generally used only to provide anonymity to the users. However, pseudonyms in the proposed scheme have a functionality of identifying friends as well as the indistinguishability for users who are not friends of the user who has generated the pseudonym. Technically, the U_i picks a secret, splits it into m pieces, and sends the pieces to each friend as a relation value. Then, U_i generates pseudonyms with the chosen secret and publishes it along with the current location information. A GPS Coordination is an example of the location information. Friends of U_i, who hold the split pieces, can confirm that the initiator of the pseudonym is generated by a friend and can identify that friend as U_i. To do so, U_i who is willing to share the location information with friends performs the following two procedures to generate a functional pseudonym.

(1) User U_i picks a uniformly random number $r \in_r \mathbb{Z}_q^*$ and computes g^r.
(2) U_i computes $g^{r \cdot a_0}$ and $g^{r \cdot f_i(1)}$

When U_i wants to use the location sharing, he/she periodically uploads his/her current location L_i along with a pseudonym P_i to the mOSN server, where

$$\begin{aligned} P_i &= \{p_{i1}, p_{i2}, p_{i3}\}, \\ p_{i1} &= g^{r \cdot H(L_i)}, \\ p_{i2} &= g^{r \cdot f_i(1)}. \\ p_{i3} &= H_1(L_i) \cdot e(g,g)^{r \cdot a_0}, \end{aligned}$$

For the indistinguishability of the pseudonym, U_i has to pick a new random number r every time when he/she sends P_i to the server. The pseudonym in our proposed scheme consists of two parts: randomness and functionality. The randomness assigns indistinguishability while functionality enables the location sharing. Whenever a user needs to upload the current location, he/she generates a new random value and a new pseudonym based on it for enhanced privacy. This does not affect the location sharing function, and thus users will be able to freely share their location information with other authorized users even after changing the random value.

4.5 Location Sharing

After the pseudonym generation process, U_i can participate in location sharing applications. The users' mobile devices are usually resource-constraint and battery-powered. Therefore, we design the location sharing to help improve the efficiency of mobile nodes by letting the server performs the computations for finding friends. To do so, U_j, who is the friend of U_i, sends the server a list of tuples, $\{I_k\}_{1 \leq k \leq m_j}$, where $I_k = (k, g^{f(k)})$ was obtained from friends. Suppose U_i uploads the pseudonym P_i, then the server first computes the following:

$$\begin{aligned} \chi &= e(g^r, g^{f_i(k) \cdot 1/(1-k)}) \cdot e(g^{H(L_i)}, p_{i2}^{k/(k-1)}) \\ &= e(g,g)^{r \cdot f_i(k) \cdot 1/(1-k) + r \cdot f_i(1) \cdot k/(k-1)} \\ &= e(g,g)^{r(f_i(k) \cdot 1/(1-k) + \cdot f_i(1) \cdot k/(k-1)} \\ &= e(g,g)^{r \cdot a_0}. \end{aligned}$$

The server can confirm that the owner of the pseudonym P_i is a friend of U_j by checking the following equation:

$$H_1(L_i) \overset{?}{=} \frac{p_{i3}}{\chi}.$$

If true, the server sends (k, L_i) to U_j. By checking the saved identification information, U_j can ensure that his/her friend U_i is at the location L_i. The server periodically checks to find friends of U_j until he/she terminates the location sharing application. This process leaks no information regarding the identities and relation between U_i and U_j. The only information the server can obtain from the process is that the two anonymous users are friends.

The proposed scheme can be utilized for the location sharing in two ways: (a) the server provides computation results regardless of his/her current location and (b) given a current location of U_j, the server provides computation results only in a specific geographical range. Both of which are based on U_j's list of tuples. We do not consider a location forgery attack by friends, where a forged GPS coordination is submitted for the purpose of tracking a specific user. Since U_i has agreed to share its location information by establishing a friendship relation, then U_j can check where he/she is located.

4.6 Pseudonym Update

Whenever the U_i adds or removes friends, his/her pseudonym must be updated as well. We consider two cases: add a friend and remove a friend. In case of adding a friend, the update process is quite simple. The Lagrange polynomial has already been computed, hence all the user U_i needs to do is to generate a new tuple $(\alpha', f_i(\alpha')$. By sending $(\alpha', g^{f_i(\alpha')}))$, U_i can share his/her location with the new friend. The newly generated tuple will not affect the pseudonym. The proposed scheme can guarantee the users' privacy if a new random value is generated and applied for the pseudonym every time. In case of removing a friend, the proposed scheme needs to repeat the process of establishing a friendship relation again. This is an essential process to preserve the users' privacy and to prevent unexpected entities from obtaining information.

5 Security Analysis

The goal of the proposed scheme lies on location sharing in mOSNs while preserving both location privacy and friendship relation privacy. In this section, we prove the proposed scheme in three folds to show that the proposed scheme provides the privacies and security against plausible attacks. First, since the pseudonyms in the proposed scheme are developed to have functionality, we prove that the functional pseudonym still has the indistinguishability, which is the major property for pseudonyms beyond the functionality. Second, we prove that an entity who does not have a friendship relation cannot obtain the originator's identity. Finally, we prove that an attacker cannot obtain a friendship relation from a given set of pseudonyms.

5.1 Indistinguishability of Functional Pseudonyms

To address the security in our proposed scheme, we first define decisional Diffie-Hellman (DDH) problem as follows:

Definition 3 (Decisional Diffie-Hellman Problem). *Considering a cyclic group $\mathbb{G}$ of order q, with the generator g, the Decisional Diffie-Hellman (DDH) problem [50] is that, given g^a and g^b for uniformly and independently chosen $a, b \in \mathbb{Z}_q$, the value g^{ab} looks like a random element in $\mathbb{G}$. The following two uniform and independent probability distributions are computationally indistinguishable in the security parameter, $n = \log q$:*

(a) g^a, g^b, g^{ab}, where a and b are randomly and independently chosen from $\mathbb{Z}_q$

(b) g^a, g^b, g^c, where a, b, c are randomly and independently chosen from $\mathbb{Z}_q$.

By combining Definition 1 and 3, the following definition, DDH problem on bilinear maps, can be driven:

Definition 4 (DBDH). *The decisional bilinear diffie-hellman (DBDH) [51] problem in groups $(\mathbb{G}, \mathbb{G}_T)$ is: given a tuple $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$ with unknown $a, b, c \in_R \mathbb{Z}_q$, whether $Z = e(g, g)^{abc}$. A polynomial-time algorithm $\mathcal{B}$ has the following advantage of negligible probability ϵ in solving the DBDH problem in groups $(\mathbb{G}, \mathbb{G}_T)$, if*

$$\begin{aligned}|(&Pr[(g, g^a, g^b, g^c, \mathbb{Z} = e(g, g)^{abc}) = 1] \\ &- Pr[(g, g^a, g^b, b^c, \mathbb{Z} = e(g, g)^d) = 1])| \leq \epsilon,\end{aligned}$$

where the probability is taken over the random choices of $a, b, c, d \in \mathbb{Z}_q$, the random choice of g in $\mathbb{G}$, and the random bits consumed by $\mathcal{B}$.

Next, we will define the indistinguishability of pseudonyms as follows:.

Definition 5 (Polynomial-time indistinguishability). *Suppose there exist PRGs that have robustness against polynomial-size circuits. Then, a random number that is generated using the PRG has polynomial-time indistinguishability [52].*

Two random numbers $X \overset{def}{=} \{X_n\}_{n \in \mathbb{N}}$ and $Y \overset{def}{=} \{Y_n\}_{n \in \mathbb{N}}$, which are uniformly distributed over $\{0, 1\}^n$ by the PRG, are indistinguishable in polynomial time if for every probabilistic polynomial-time algorithm D, every positive polynomial $p(\cdot)$, and all sufficiently large n's, this relation holds:

$$|Pr[D(X_n, 1^n) = 1] - Pr[D(Y_n, 1^n) = 1]| < \frac{1}{p(n)}.$$

Based on the Definitions 3 and 4, we prove the indistinguishability of functional pseudonyms, which is our main contribution.

Theorem 1 (Indistinguishability of functional pseudonyms). *Suppose there exist pseudo random generators (PRGs) that have robustness against polynomial-size circuits, and let A be an adversary for which its advantage can be defined as follows:*

$$Adv_{Ind}(A) = |Pr[D(fp) = 1] - Pr[D'(r) = 1]|,$$

where D is a probabilistic algorithm that halts in polynomial time with output 1 when the input fp is a functional pseudonym. Otherwise with output 0; for D' as another probabilistic algorithm that halts in polynomial time, with output 1 when the input r is generated by a PRG, or with output 0 in polynomial time.

A functional pseudonym that is created with a random number generated by the PRG has polynomial-time indistinguishability and the advantage must be negligible under DDH assumption.

Proof. Suppose a set of pseudonyms $P = \{P_1, P_2, ...P_m\}$ were obtained by an attacker. We are going to prove any of two given pseudonyms, P_i, P_j from P, are indistinguishable while providing the functionality.

Specifically, from the p_{i1} and p_{j1}, we can say with confidence that the values are indistinguishable under the DDH as the two values are generated using random numbers. The $P_{i2} = g^{r \cdot f_i(1)}$ can simply be written as $g^{r_i \cdot \gamma_i}$, where r_i is a randomly chosen number in $\mathbb{Z}_q^*$ and γ_i acts as a secret part. From p_{i2} and p_{j2}, each of these values can also be represented as $\widehat{g}^{r_i}, \overline{g}^{r_j}$. Since g is a generator of cyclic group $\mathbb{G}$, $\widehat{g}$ and $\overline{g}$ are also a generator of $\mathbb{G}$. Similar to this, the p_{i3} and p_{j3} also have indistinguishability under the Definition 4, i.e. DBDH. All the three values of pseudonyms are generated using two uniform random numbers r_i and r_j, thus it has polynomial-time indistinguishability by Definition 5. Therefore, the $Adv(A) \leq \epsilon$, where ϵ is a negligible probability. □

In the proposed scheme, users can easily generate a new pseudonym by changing a random number which has no effect on the functionality of pseudonyms. Users can use a new pseudonym whenever they need to communicate with the mOSN server, therefore it is not possible for attackers to obtain a piece of information from pseudonyms in polynomial-time. Also, because of indistinguishabilitity, is not possible to obtain a correlation between functional pseudonyms even with a statistical attack.

In addition, an attacker may try to infer a piece of information from pseudonyms and relation values, which are public to the server. Thus, we are going to prove the semantic security of the proposed scheme, in which the attacker cannot infer any information from pseudonyms in polynomial time.

5.2 Security of Functional Pseudonyms

To prove the security of the functional pseudonym, we will demonstrate that only users in friendship relation can identify the owner of pseudonyms.

Theorem 2 (Security of Pseudonyms)**.** *Anyone who is not a friend of U_i cannot identify the originator of P_i. Specifically, an adversary A has the following advantage:*

$$Adv_{Sec}(A) = |Pr[B(\chi, P) = 1]|,$$

where B is a probabilistic algorithm that halts in polynomial time with output 1 when the P contains the randomly selected value χ, otherwise output 0, χ is a pair of random values ($\chi = (\chi_1, \chi_2) \in \mathbb{Z}_q$), and P is a functional pseudonym.

Proof. To verify this, we apply a security analysis of the secret sharing scheme that uses $Vandermonde\ matrix$ [47].

The secret χ of a pseudonym derived by $(2, n)$ secret sharing can be represented as $f(x) = a(i,0) + a(i,1)x \in \mathbb{Z}_q$, and $f(0) = \chi$. The solution to recover χ can be described by multiplication of the following matrices:

$$\begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix} = \begin{bmatrix} 1 & y_i \\ 1 & y_t \end{bmatrix} \times \begin{bmatrix} \sum_{i=1}^{n} a(i,0) \\ \sum_{i=1}^{n} a(i,1) \end{bmatrix}$$

The second matrix of the equation is the well-known $Vandermonde\ matrix$ which has the non-zero determinant of the matrix. Thus, the coefficients of the matrix $\{\sum_{i=1}^{n} a(i,0), \sum_{i=1}^{n} a(i,1)\}$ have a unique solution over $\mathbb{Z}_q$.

If a user U_k who does not belong to the list of friends of U_i tries to recover the secret χ, he/she has to obtain that by solving the following linear equations: $\chi_1 = \sum_{k=1}^{n} a(k,0) + y_k \cdot \sum_{k=1}^{n} a(k,1) \in \mathbb{Z}_q$, and $\chi_2 = \sum_{k=1}^{n} a(k,0) + y_t \cdot \sum_{k=1}^{n} a(k,1) \in \mathbb{Z}_q$. Since the coefficient matrix of the $Vandermonde\ matrix$ has a unique solution, the two equations can derive a unique solution such that $\chi' = f'(0)$. Hence, there exist $q \times q$ cases for the pair (χ_1, χ_2). In addition, q exponential operation time is needed to confirm that a secret value of the P is χ, and the advantage is: $Adv_{Sec}(A) = 1/q^{2q}$ which is negligible with a sufficiently large q. Therefore, any user who is not in friends list cannot recover the secret of a functional pseudonym, hence cannot identify the originator of it. □

By the Theorem 1 and Theorem 2, Polynomial-time Indistinguishability and Security of Pseudonyms, we can argue that an attacker cannot obtain any information from pseudonyms. Even though an attacker succeeds to recover χ from eavesdropped message, it has to decrypt an encrypted message in order to obtain a correlation between χ and a friendship relation, for which its security is proven by $Theorem$ 3.

5.3 Friendship Privacy

Although the server may find a corresponding pseudonym for users' relation values, it needs other values to identify the owner of a pseudonym or discover the relation between users. The one and only option for this is eavesdropping the encrypted data from the relation value exchange process. The server can obtain identification factor through cryptanalysis of the encrypted data. For the friendship relation privacy, we prove the semantic security of functional pseudonym.

Theorem 3 (Semantic security of functional pseudonyms)**.** *The functional pseudonym proposed in this paper has semantic security, which means that no information can be obtained from functional pseudonyms, thus attackers cannot obtain information from that in a polynomial time. Specifically, an adversary A has the following advantage:*

Proof. The only way for the server (or other attackers) to identify friendship relations through location sharing is to obtain identification factors from the process of establishing a friendship relation. Suppose an attacker can obtain encrypted relation value $\mathcal{E} = E_{pk_i}\{(\alpha, g^{f_i(\alpha)})\}$, where $E_k\{m\}$ is a public key encryption (e.g., ElGamal encryption) using the key k. In addition, let's assume $\mathcal{O}$ to be an oracle which can solve the proposed scheme in polynomial time. The $\mathcal{O}$ is represented as: $\mathcal{O}(q, g, r, \mathcal{E})$, where q is k-bit prime number, g is a generator of cyclic group generated by q, and r is a random number picked by the attacker. The oracle outputs $true$ if $\mathcal{O}$ can distinguish whether r is used to encrypt the $\mathcal{E}$ in polynomial time, otherwise outputs $false$.

Now, we prove that the DDH assumption is solvable under the assumption of $\mathcal{O}$. Given g^a and g^b, the attacker picks r randomly and sends queries $\mathcal{O}(q, g, r, g^a)$ and $\mathcal{O}(q, g, r, g^b)$. Since the oracle can return the result in polynomial time, the attacker could distinguish g^a over g^b with q queries in the worst case scenario.

Thus, if the proposed scheme is solvable using $\mathcal{O}$, then DDH is also solvable using $\mathcal{O}$. However, DDH is a well-known difficult problem and is already proven that it cannot be solved in a polynomial time. Therefore, the problem of identifying a friendship relation in the proposed scheme is as hard as DDH by a reduction. □

By the Theorem 2 and 3, we can say that an attacker cannot obtain a relation information from pseudonyms. Therefore, the proposed scheme preserves the friendship relation privacy.

6 Efficiency Analysis

Unlike the previous schemes, the proposed scheme has enhanced privacy as it can provide the location privacy and friendship privacy concurrently. Thus, a direct comparison of the efficiency at the same level between the previous schemes and the one presented on this work is unfair. Instead, we will provide computational costs of the proposed scheme, then will provide a comparison among different schemes as fair as possible.

6.1 Computational Costs

First, we measure the computational cost of three important functions that are used in the proposed scheme: establishing friendship relations, generating a pseudonym, and checking adjacent friends (location sharing). Table 2 shows the computational costs of the main functions based on the major operations; exponential (Exp.), multiplication (Mux.), bilinear pairing (Pairing), hash function (Hash), and comparison (Comparison).

TABLE 2: Computational cost based on four major operations (m : number of friends, n : number of adjacent users).

	establishing friendships	Pseudonym generation	Checking friends (Server)	Checking friends (user)
Exp.	m	3	$n * m$	0
Mux.	m	3	$3 * n * m$	0
Pairing	0	1	$2 * n * m$	0
Hash	0	1	n	0
Comparison	0	1	n	m

As shown in Table 2, the proposed scheme is highly efficient for mobile devices in terms of computational costs. For m as the number of friends, it has a time complexity of $O(c_{exp} \cdot c_{mul} \cdot m)$ for establishing friendship relations, where c_{exp} and c_{mul} are constants on the computational cost of exponential and multiplication, respectively. Also, for n as the number of users in a given range, it has a time complexity of $O(c_1 \cdot n \cdot m)$ for checking friends, where c_1 is a constant on the computational cost of checking friends.

Here, we would like to emphasize that the proposed scheme can delegate most of the location sharing related computations to the server. The server conducts the heavy computations upon a user's request and sends the results back to the user. Then, the user will be able to check identities of friends through a simple comparison of computations. In terms of communication overhead, our scheme is highly efficient. Each request contains current location information and a number of tuples (anonymous identities, location information), and a list of adjacent friends will be returned. There is no extra information to obfuscate attackers, and therefore it is highly efficient.

Our scheme was developed to achieve sustainability for mOSNs which are resource-constraint and battery-powered mobile devices. Especially, one of our major contribution, the delegable location sharing that is conducted on the server, can significantly improve sustainability of mobile devices.

6.2 Comparison with Previous Schemes

Comparing the proposed scheme with the SMILE in terms of storage and communication overhead. SMILE establishes a social relationship based on the encounter shared among users located at the same proximity at the same time. Regardless of the number of friends, a user in SMILE has to store the encounter keys for every visited location and their corresponding times of visit. The number of stored keys piles up as the time of usage increases. This also causes a problem of searching the encounters shared with the target participants. In comparison, the proposed scheme needs to store the number of public identities which are the same as the number of friends in the list. Moreover, the proposed scheme does not need to find the shared secrets.

For the purpose of comparing the necessary communications, SMILE needs more communication overhead than our proposed scheme. Suppose there are n number of users. Whenever a user sends a message, it will broadcast the geographical area where the user is located through the server. Thus, every user in that area must receive the sent message and check whether it is coming from one of their friends. At this point, the problem of searching encounter occurs again. If k users $(1 \leq k \leq n)$ is sending messages to find friends in a location, the server has to broadcast k messages, and each user has to receive and check these k messages. However, in the proposed scheme, a sent message does not influence other users in the designated location. The proposed scheme requests a set of user information from the server and checks where the friends are located. Therefore, the proposed scheme can drastically reduce the communication overhead which is a burden on mobile devices.

TABLE 3: Comparison of characteristics and overheads in terms of users. m, number of friends, and n, number of users inside the area of interest.

Features	SMILEs	Mobishares	Our scheme
Location privacy	O	O	O
Friendship privacy	X	X	O
No encounter	X	O	O
No pre-established secret	O	X	O
No fully trusted entity	O	X	O
Search request	m	m	1
Key storage	m	m	1
Friends confirmation	1	1	0
Communication	m	m	1
Delegable computation	X	X	O

Situations are similar in series of MobiShare systems [16], [17], [18], [53]. Each pair of users needs to share a symmetric key, and this reduces the key management efficiency. In terms of finding friend requests, a user in MobiShare systems needs to make a request message for all friend users. When the user has m number of friends, the user should compute m request messages using the symmetric keys that are shared with friends then sends m messages to the service provider.

Table 3 illustrates the summary of comparison results among location sharing services in terms of characteristics and overheads. Our scheme is designed to provide the location sharing service which can preserve (a) location privacy and (b) friendship relation privacy with no pre-established secret, no trusted server, and no encounter. In addition, our scheme does not need to share and manage a secret with friends for location sharing services. The only secret in our scheme is a link between a relation value and the real identity of the originator.

While the overhead of SMILE and Mobishare schemes depends on the number of user's friends, the overhead of our scheme is dependent on the number of users in the area of interest. Since a user in our scheme always sends a single request message to search the adjacent friends in that area, our scheme is appropriate for the location sharing in mOSNs where users have numerous friends, thanks to the delegable computation.

7 Evaluation

To prove the proposed scheme is suitable for mobile environments, we simulated and demonstrated the functional pseudonym scheme. All of the simulations were performed in the Linux Mint version 17.3 installed in the Oracle VM VirtualBox version 5.1.12 where a 3 GB memory was allocated for the operating system. The VirtualBox was installed

TABLE 4: Execution time of major functions (with 100 friends

Functions	Execution time (msec)
Establishing a friendship relation	122.14
Assigning randomness	6.00
Finding anonymous friends (delegable)	387.7
Checking identities (user side)	0.7

on the desktop computer that has Intel Core i7-4790 CPU 3.6 GHz with 8 GB memory, 64-bit operation system, and 7200rpm SATA 3.0 hard drive. In addition, we applied a restriction that the VirtualBox operates with only one processor to make it as similar environment as those on mobile devices.

All simulation programs were implemented using GNU Compiler Collection (GCC) version 4.8.4. We used Pairing-Based Cryptography (PBC) library [54] and the GNU Multiple Precision Arithmetic Library (GMP) version 6.1.2 [55] for the cryptographic computations. All experimental results are the average of 100 trials.

We simulated the running time of two major processes: the pseudonym generation and the location sharing. The pseudonym generation process were separated into two subprocesses: establishing friendship relations and assigning randomness. The location sharing process were also separated into two subprocesses: finding anonymous friends which can be performed by the server, and checking identities which will be performed by the users. We assumed that a user had 100 friends on a list, which effected the process of establishing friendship relations and location sharing.

Table 4 presents the execution time of the major functions. The experimental results show that the computations to establish friendship relations with 100 friends was done in 122.14 millisecond (msec), and it exclusively required 100 communications to exchange relation values. In addition, it has taken 6.00 msec to generate a pseudonym. Since the generated friendship relations are reusable unless they are eliminated, the users need to repeatedly compute only a small part of pseudonyms. Checking a functional pseudonym based on relation values in order to distinguish whether the pseudonym was generated by a friend or not has taken 3.87 msec. When a user had 100 friends, it took 387.7 msec which was a great burden on mobile devices. Fortunately, the server is in charge of these computations, and thus users do not have to consider these time overheads. Since the execution time is calculated by our mobile environment, actual computations by the server is much faster. The only thing that users have to do is to retrieve a friend's real identity, which takes only 0.7 msec with the one-by-one comparison strategy. It would be faster if an advanced search algorithm is applied which is out of the scope of this paper.

These results prove that the proposed scheme is highly efficient and is sufficient for mobile usages. In addition, the delegable location sharing significantly reduces the computational overhead of users, and consequently makes LBSs highly sustainable.

8 Concluding Remarks

This paper presented a privacy enhanced location sharing scheme for mOSNs. The goal was to solve security problems associated with 1) the user's location privacy and 2) friendship relation privacy, and securely providing information to share their current location. In order to achieve this goal, we developed a new cryptographic primitive, named functional pseudonym. Using the pseudonym, the user's identity/location as well as his/her friendship relations can be protected from unintended entities while an authorized user can identify the owners of different pseudonyms.

The proposed scheme uses Lagrange polynomial to merge friendship relations (represented by relation values) into a single value for the efficiency reason. Later, the relation values are applied to identify friends in a real-time manner. We developed the proposed scheme in a way that the server performs the heavy computations which is a part of the location sharing process. This significantly improves the efficiency of the computations on resource-constraint mobile devices.

The security analysis proved that (a) the proposed scheme holds indistinguishability while providing the location privacy and friendship relation privacy, (b) an attacker could not obtain identity/location information from the functional pseudonym, and (c) an attacker could not infer and/or obtain friendship relations from the functional pseudonyms. In addition, the efficiency analysis showed that the proposed scheme is one of the most suitable solutions for location sharing in mOSNs, and the evaluation demonstrated that the proposed scheme has sufficient efficiency and sustainability for mobile devices.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01860, A Study on Functional Signature and Its Applications)

References

[1] J. Son, D. Kim, R. Tashakkori, A. O. Tokuta, and H. Oh, "A new mobile online social network based location sharing with enhanced privacy protection," in *ICCCN 2016 Conference Proceedings*. IEEE, August 2016, pp. 1–9.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, December 2013.

[3] S. Peng, G. Wang, and D. Xie, "Social influence analysis in social networking big data: Opportunities and challenges," *IEEE Network*, vol. 31, no. 1, pp. 11–17, January/February 2017.

[4] M. Carman and K.-K. R. Choo, "Tinder me softly how safe are you really on tinder?" in *SecureComm 2016 Conference Proceedings*, June 2017, pp. 271–286.

[5] M. Ficco, F. Palmieri, and A. Castiglione, "Hybrid indoor and outdorr location services for new generation mobile terminals," *Personal and Ubiquitous Computing*, vol. 18, no. 2, pp. 271–285, February 2014.

[6] D. Quercia, N. Lathia, F. Calabrese, G. D. Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," in *IEEE ICDM 2010 Conference Proceedings*. IEEE, December 2010, pp. 971 – 976.

[7] C. R. Vicente, D. Freni, C. Bettini, and C. S. Jensen, "Location-related privacy in geo-social networks," *IEEE Internet Computing*, vol. 15, no. 3, pp. 20–27, February 2011.

[8] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, "Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies," *The international Journal of Very Large Data Bases*, vol. 20, no. 4, pp. 541–566, August 2011.

[9] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understaiding graph-based trust evaluation in online social networks: Methodologies and challenges," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, pp. Article 10:1–35, May 2016.

[10] T. Wang, J. Zeng, M. Z. A. Bhuiyan, H. Tian, Y. Cai, Y. Chen, and B. Zhong, "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, May 2017.

[11] M. Li, H. Zhu, Z. Gao, S. Chen, L. yu, S. Hu, and K. Ren, "All your location are belong to us: Breaking mobile social networks for automated user location tracking," in *ACM MobiHoc 2014 Conference Proceedings*. ACM SIGMOBILE, August 2014, pp. 43–53.

[12] R. Shetty, G. Grispos, and K.-K. R. Choo, "Are you dating danger? an interdisciplinary approach to evaluating the (in)security of android dating apps," *IEEE Transactions on Sustainable Computing*, vol. Early Access, pp. 1–11, December 2017.

[13] L. P. Cox, A. Dalton, and V. Marupadi, "Smokescreen: Flexible privacy controls for presence-sharing," in *ACM MobiSys 2007 Conference Proceedings*. ACM SIGCOMM, June 2007, pp. 233–245.

[14] L. C. J. Manweiler, R. Scudellari, "Smile: encounter-based trust for mobile social services," in *ACM CCS 2009*. ACM SIGSAC, November 2009, pp. 246–255.

[15] W. Wei, F. Xu, and Q. Li, "Mobishare: Flexible privacy-preserving location sharing in mobile online social networks," in *IEEE INFOCOM 2012 Conference Proceedings*. IEEE Computer and Communication Society, March 2012, pp. 2616–2620.

[16] J. Li, J. Li, X. Chen, Z. Liu, and C. Jia, "Mobishare+: Security improved system for location sharing in mobile online social networks," in *IEEE INFOCOM 2012 Conference Proceedings*. IEEE Computer and Communication Society, March 2012, pp. 2616–2620.

[17] N. Shen, K. Yuan, J. Yang, and C. Jia, "B-mobishare: Privacy-preserving location sharing mechanism in mobile online social networks," in *BWCCA 2014 Conference Proceedings*. IEEE, November 2014, pp. 312–316.

[18] Z. Liu, D. Luo, J. Li, X. Chen, and C. Jia, "N-mobishare: New privacy-preserving location-sharing system for mobile online social networks," *International Journal of Computer Mathematics*, vol. Published online, May 2014.

[19] N. Li, N. Zhang, and S. K. Das, "Preserving relation privacy in online social network data," *IEEE Internet Computing*, vol. 15, no. 3, pp. 35–42, January 2011.

[20] S. Preibusch and A. R. Beresford, "Privacy-preserving friendship relations for mobile social networking," in *W3C Workshop on the Future of Social Networking*, 2009.

[21] N. Li, N. Zhang, and S. K. Das, "Relationship privacy preservation in publishing online social networks," in *IEEE PASSAT/SocialCom Conference Proceedings*. IEEE, October 2011, pp. 443–450.

[22] G. Carullo, A. Castiglione, A. D. Santis, and F. Palmieri, "A triadic closure and homophily-based recommendation system for online social networks," *World Wide Web*, vol. 18, no. 6, pp. 1579–1601, November 2015.

[23] M. A. Rahman, V. Mezhuyev, M. Z. A. Bhuiyan, S. M. N. Sadat, S. A. B. Zakaria, and N. Refat, "Reliable decision making of accepting friend request on online social networks," *IEEE Access*, vol. 6, pp. 9484–9491, February 2018.

[24] H. Hacigumus, B. Iyer, C. Li, and S. Mehrotra, "Executing sql over encrytped data in the database-service-provider model," in *ACM SIGMOD 2002 Conference Proceedings*. ACM, June 2002, pp. 216–227.

[25] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, December 2010.

[26] T. Xiang, X. Li, F. Chen, Y. Yang, and S. Zhang, "Achieving verifiable, dyanmic and efficient auditing for outsourced database in cloud," *Journal of Parallel and Distributed Computing*, vol. 112, no. 1, pp. 97–107, February 2018.

[27] E. Snekkeness, "Concepts for personal location privacy policies," in *ACM EC 2001 Conference Proceedings*. ACM SIGecom, October 2001, pp. 48–57.

[28] A. K. Pietilaine, E. Oliver, J. Lebrun, and G. Varghesei, "Mobiclique: Middleware for mobile social networking," in *ACM SIGCOMM 2009 Conference Proceedings*. ACM SIGCOMM, August 2009, p. 68.

[29] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *ACM MobiSys 2003 Conference Proceedings*. ACM SIGMOBILE, May 2003, pp. 31–42.

[30] T. Jiang, H. J. Wang, and Y. C. Hu, "Preserving location privacy in wireless lans," in *ACM MobiSys 2007 Conference Proceedings*. ACM SIGCOMM, June 2007, pp. 246–257.

[31] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy protection for users of location-based services," *Annals of telecommunications*, vol. 69, no. 1-2, pp. 47–62, August 2014.

[32] S. Mascetti, C. Bettini, and D. Freni, "Longitude: Centralized privacy-preserving computation of users' proximity," in *VLDB Workshop on Secure Data Management Conference Proceedings*. LNCS, August 2009, pp. 142–157.

[33] S. Mascetti, C. Bettini, D. Freni, X. S. Wang, and S. Jajodia, "Privacy-aware proximity based services," in *MDM 2009 Conference Proceedings*. IEEE, May 2009, pp. 31–40.

[34] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 1–18, January 2008.

[35] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knoledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, December 2007.

[36] J. Li, H. Yan, Z. Liu, X. Chen, X. Huang, and D. S. Wong, "Location-sharing systems with enhanced privacy in mobile online social networks," *IEEE Systems Journal*, vol. 11, no. 2, pp. 439–448, June 2017.

[37] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "Privacy-preserving location sharing services for social networks," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 811–825, September/October 2017.

[38] T. Peng, Q. Liu, and G. Wang, "Enhanced location privacy preserving scheme in location-based services," *IEEE Systems Journal*, vol. 11, no. 1, pp. 219–230, March 2017.

[39] G. Suna, Y. Xiea, D. Liaoa, H. Yua, and V. Changd, "User-defined privacy location-sharing system in mobile online social networks," *Journal of Network and Computer Applications*, vol. 86, no. 2017, pp. 34–45, May 2017.

[40] C. Tang and C. Cai, "Verifiable mobile online social network privacy-preserving location sharing scheme," *Concurrency and Computation: Practice and Experience*, vol. Online published, pp. 1–10, August 2017.

[41] I. Polakis, G. Argyros, and T. Petsios, "Where's wally?: Precise user discovery attacks in location proximity," in *ACM CCS 2015 Conference Proceedings*. ACM SIGSAC, October 2015, pp. 817–828.

[42] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, no. 10, pp. 42–61, February 2017.

[43] D. I. Volinsky, E. Syta, and B. Ford, "Hang with your buddies to resist intersection attacks," in *ACM CCS 2013 Conference Proceedings*. ACM SIGSAC, November 2013, pp. 1153–1166.

[44] G. Danezis and A. Serjantov, "Statistical disclosure or intersection attacks on anonymity systems," in *Information Hiding*. Lecture Notes in Computer Science, May 2004, pp. 293–308.

[45] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *ACM ASIACRYPT01 Conference Proceedings*, 2001, pp. 514–532.

[46] M. Hazewinkel, *Bilinear Mapping*. Springer, 1994.

[47] I.-C. Lin and C.-C. Chang, "A (t,n) threshold secret sharing system with efficient identification of cheaters," *Computing and Informatics*, vol. 24, pp. 529–541, August 2005.

[48] A. Beimel, "Secret-sharing schemes: A survey," in *Proc. of the 3rd International Conference on Coding and Cryptology*, ser. IWCC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 11–46. [Online]. Available: http://dl.acm.org/citation.cfm?id=2017916.2017918

[49] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, November 1979.

[50] D. Boneh, "The decision diffie-hellman problem," in *The 3rd Algorithmic Number Theory Conference Proceedings*. Lecture Notes in Computer Science, June 1998, pp. 48–63.

[51] K. Benson, H. Shacham, and B. Waters, "The k-bdh assumption family: Bilinear map cryptography from progressively weaker

assumptions," in *CT-RSA13 Conference Proceedings*, February 2013, pp. 310–325.

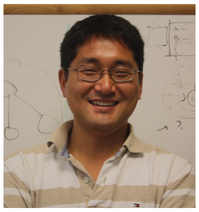
[52] O. Goldreich, "The foundations of cryptography, vol. 1, basic tools," *Cambridge University Press*, 2001.

[53] J. Li, H. Yan, Z. Liu, X. Chen, X. Huang, and D. Wong, "Location-sharing systems with enhanced privacy in mobile online social networks," *IEEE Systems Journal*, vol. Published online, pp. 1–10, April 2015.

[54] B. Lynn, "The pairing-based cryptography library," in *https://crypto.stanford.edu/pbc/*.

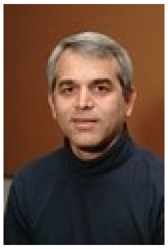
[55] "The gnu multiple precision arithmetic library," in *https://gmplib.org/*. version 6.1.2.

Junggab Son received the BS degree in computer science and engineering from the Hanyang University, Ansan, South Korea (2009), and the MS degree in computer science and engineering from Hanyang University, South Korea (2011). Dr. Junggab Son obtained his Ph.D. in computer science and engineering from Hanyang University, South Korea (2014). Currently, he is an assistant professor in the Department of Computer Science at Kennesaw State University. His research interests include applied cryptography and security/privacy issues in various applications such as cloud computing, internet of things, vehicular ad hoc network, social network services, and bioinformatics.



Donghyun Kim received the BS degree in electronic and computer engineering from the Hanyang University, Ansan, Korea (2003), and the MS degree in computer science and engineering from Hanyang University, Korea (2005). He received the PhD degree in computer science from the University of Texas at Dallas, Richardson, USA (2010). Currently, he is an assistant professor in the Department of Computer Science at Kennesaw State University, Marietta, USA. From 2010 to 2016, he was an assistant professor in the Department of Mathematics and Physics at North Carolina Central University, Durham, USA. His research interests include security and privacy, social computing, mobile computing, cyber physical systems, wireless and sensor networking, and algorithm design and analysis. He is an associate editor of Discrete Mathematics, Algorithms and Applications. He is a member of ACM and a senior member of IEEE.

Md Zakirul Alam, Ph.D. is currently an Assistant Professor of the Department of Computer and Information Sciences at Fordham University, USA. Previously, he worked as an Assistant Professor at Temple University and a Postdoctoral Research Fellow at Central South University, China. His research focuses on dependability, cyber security, big data, and cyber-physical systems. Dr. Bhuiyan has served as an associate/lead guest editor for key journals including IEEE Transactions on Big Data, ACM Transaction on Cyber-Physical Systems, Information Sciences, IEEE IoT Journal, and Int'l Journal of Computer and Application, and cluster computing. He has also served as the general chair, program chair, workshop chair, publicity chair, TPC member, and a reviewer of various international journals/conferences. He is a member of the IEEE and the ACM.



Rahman Tashakkori was born in 1963 in Jahrom Iran. He received his BS degree in Physics from Ahwaz University in 1987. He completed his first and second MS degrees at Louisiana State University in Nuclear Engineering and Engineering Science in 1994 and 1995, respectively. He received his PhD in Computer Science also from Louisiana State University in 2001.Rahman served as a Research Associate and Instructor in the Department of Physics at Southern University Baton Rough from 1996-2000. He joined the Department of Computer Science at Appalachian State University as an Assistant professor in August 2000. He is currently serving as the Chair and Lowe's Distinguished Professor of Computer Science at Appalachian State University and serve as the director of the Visual and Image Processing lab. Rahman has directed several projects funded by the National Science Foundation which include the CSEMS, S-STEM, STEM, RET, CCLI, and IUSE programs. Rahman is a Senior Member of IEEE.

Jungtaek Seo Ph.D. received his degree in information security from the graduate school of Information Security, Korea University, in 2006. From November 2000 to February 2016, he has worked for National Security Research Institute as a senior researcher as well as the head of Infrastructure Protection Research Department. Currently, he is an assistant professor in Department of Information Security Engineering, Soonchunhyang University. He has been a Principal Investigator of several government sponsored research project in SCADA, Smart Grid, nuclear power plants. Recently, he has been actively working in the area of smart grid, in particular with respect to standard and policies. His research interest includes SCADA, Smart Grid, nuclear power plants, Smart City, CPS (Cyber Physical System).

Dong Hoon Lee received a BS degree from the Department of Economics at Korea University, Seoul, in 1985, and MS and PhD degrees in computer science from the University of Oklahoma, Norman, in 1988 and 1992, respectively. Currently, he is a professor of the Graduate School of Information Security at Korea University. Since 1993, he has been with the Faculty of Computer Science and Information Security at Korea University. His research interests include cryptographic protocol, applied cryptography, functional encryption, software protection, mobile Security, vehicle security, and ubiquitous sensor network (USN) security. He is a fellow of the IEEE.