A Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

MUNMUN BHATTACHARYA[1], SANDIP ROY[2], KAMLESH MISTRY[3],
HUBERT P. H. SHUM[4], (Senior Member, IEEE),
AND SAMIRAN CHATTOPADHYAY[3], (Senior Member, IEEE)

[1]Department of Information Technology, Jadavpur University, Kolkata 700106, India
[2]Department of Computer Science and Engineering, Asansol Engineering College, Asansol 713305, India
[3]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.
[4]Department of Computer Science, Durham University, Durham DH1 3DE, U.K.

Corresponding author: Samiran Chattopadhyay (s.chattopadhyay@northumbria.ac.uk)

ABSTRACT The rapid development of mobile internet technology and the better availability of GPS have made mobile online social networks (mOSNs) more popular than traditional online social networks (OSNs) over the last few years. They necessitate fundamental social operations such as establishing friend relationship, location sharing among friends, and providing location-based services. As a consequence, security and privacy issues demands the utmost importance to mOSNs users. The first stream of existing solutions adopts two different servers to store locations-based and social network-based information separately, thereby sustaining large storage and communication overhead. The second stream of solutions aims at integrating the social network server and the location-based server into a single entity. However, as these approaches exploit only one single server, they may face several performance issues related to server bottlenecks. Moreover, such schemes are found to be vulnerable to various active and passive security attacks. In this paper, we propose a privacy preserving, secure and efficient location sharing scheme for mOSNs, which shows both efficiency and flexibility in the location update, sharing, and query of social friends and social strangers. The security of the proposed scheme is validated using random oracle based formal security proof and Burrows-Abadi-Needham (BAN) logic based authentication proof, followed by informal security analysis. Additionally, we have used ProVerif 1.93 to verify the security of the system. The efficiency and practicability of the proposed scheme are demonstrated through experimental implementation and evaluation.

INDEX TERMS Mobile online social networks, privacy, location sharing and query, BAN logic, random oracle.

I. INTRODUCTION

The advancement of mobile internet technology over the last few years have shifted online social networks (OSNs) users towards its more flexible and dynamic version, namely mobile online social network (mOSNs). In general, mobile users keep their mobile devices in online mode anytime, anywhere. This allows the mobile device to use the current location information, thereby providing support to a range of location-based services such as current location sharing,

The associate editor coordinating the review of this manuscript and approving it for publication was Young Jin Chun.

social friend or stranger's location query, etc. Nowadays, mOSNs users can use location-based services to recommend good social friend, search various intended Points of Interests (PoIs) such as restaurants, movie halls and hospitals.

Online Social Networks (OSN) is an online platform which people use to build social networks or social relationships with other people who share similar personal or career interests and activities [1]. Normally, people use PC or laptop to use and access online social network services.

Mobile online social networking (mOSN) involves the interactions between participants with similar interests and objectives through their mobile devices and/or tablet within

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

virtual social networks [1]. mOSN leverages mobile communication networks and social networks, as mobile applications can use existing social networks. In mOSN, social networks can take advantage of mobile features and ubiquitous accessibility. Moreover, an mOSN can readily exploit mobile networks to support the concept of real-time web [2], which is at the forefront of the emerging trends in social networking. mOSNs enhance conventional social networks with additional features, such as location-awareness, tag media [3], etc.

mOSNs can take advantage of the additional capabilities of modern mobile devices such as smartphones or tablets. People can access mOSNs applications anywhere and anytime. These capabilities, such as global position system (GPS) receiver, sensing modules (cameras, sensors, etc.), and multiple radios (third/fourth generation cellular, WiFi, Bluetooth, WiFi Direct, etc.), enable mOSNs to enhance conventional social networks with additional features, such as location-awareness [5], location-based service, the ability to capture and tag media [3]. In general built-in GPS is not that much available in laptops. Moreover, it does not exploit 4G or the current standard of cellular networks. Hence, location-based services cannot be accessed using laptops [1].

Location sharing through mOSN may end up in catastrophic failure, especially when privacy and security measures are not implemented properly. On the one hand, the popularity and usage of mOSN based applications are increasing every day. On the other hand, different malicious users and attackers continuously engineer innovative attacks to unlawfully access and modify various social and physical information of the registered mOSN users. The implementation of a secured, privacy-preserving location sharing strategy while sustaining the modern-day facilities of various mOSN applications is a serious research challenge.

Detail security analysis reveals the vulnerability of the existing related schemes against many security attacks, such as the denial-of-service (DoS) attack [6], replay attack [6], [7] and privileged insider attack [6], [7]. A recent study reveals that two attacking tricks, namely Regional Statistical Attack (RSA) [8] and Long-term Statistical Attack (LSA) [8], give more opportunity to the attackers.

In this paper, we propose a new location sharing scheme for mobile online social network applications, in which the limitations of the earlier schemes concerning security and functionality are overcome. The proposed system adopts a model, where the social network server (SNS) and the location-based server (LBS) are integrated into one single entity. To share privacy-preserving locations, the proposed scheme exploits dummy locations, a dedicated mapping protocol among the Cellular Tower (CT) and a set of location-storing social network servers. Various security attacks including the strong replay attack, man-in-the-middle attack, etc., which are prevalent in existing schemes, can be successfully overcome by our scheme.

Formal security validation of the proposed scheme is achieved through ProVerif 1.93 simulation tool. The Real-or-Random (ROR) model based on the random oracle model is employed to verify the security of the proposed scheme formally. Moreover, BAN logic is used to prove authentication of the proposed system. We logically explain how the proposed scheme defends various active and passive security attacks by analyzing it informally. Experimental implementation and evaluation results demonstrate the efficiency and practicality of the proposed scheme.

Our study shows that although modern smartphones have privacy and security-based location sharing features, those services requires improvements in the security aspects. Moreover, in current systems, location sharing to a large number of friends may incur substantial security hazards.

First, although popular online social networks provide many facilities to social life, they also increase the danger of user privacy breaches due to direct and indirect location sharing. A few studies have attempted to address the location privacy issues in MSNs [9]–[11]. Recently, H. Li *et al.* presented an empirical research to quantify private information leaking issues arising from location sharing in popular OSNs such as Facebook and Twitter [12]. They conducted a three-week real-world experiment with 30 participants, and discovered that direct and indirect location sharing by popular OSNs could reveal 16% and 33% of the users' real points of interest (POIs) respectively. External adversary was able to attack to infer the demographics (e.g., age, gender, education) after observing the exposed users' location profiles. H. Li *et al.* implemented such an attack in a large real-world dataset involving 22,843 mobile users [12]. Many popular social networks provide location-based sharing functionalities like geolocation tags and check-in services. Based on these functionalities, the attacker can easily obtain the location information shared by the mobile users by crawling the interested information from web pages and extracting POIs from the collected data [10], [13].

Second, It is possible for a privileged insider to execute location spoofing intentionally, providing fake locations on the location-based features of Facebook, WhatsApp and Snapchat (e.g. Nearby Friends and Snap Map). This is done using downloadable apps like FakeGPS, in order to deceive the social friends for malicious purpose [14].

Third, sharing location information is less safe especially when a person has large number of friends or followers whom he/she might not actually know. Location sharing and friend's location query should be done on a restricted basis where the communicating parties can limit the distance threshold by which they can find each other.

In this paper, we address above security drawbacks of the existing location-based features by popular OSNs. According to our proposed scheme, MU_i and $LSSNS$ first separately establish a shared symmetric session key with CT. All location updates and friend's location query messages are encrypted with this session key before transmission. Because of this end-to-end encryption, an adversary $\mathcal{A}$ has little chance to reveal the location information of MU_i. Furthermore, unlike the location-based services of existing OSNs, our proposed scheme allows a user to decide a distance threshold,

IEEE *Access*

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

up to which he/she wants to make himself/herself visible to the social friends. This imposes a much better user controlled restriction on location sharing, as unrestricted location sharing can lead to security vulnerabilities.

A. MOTIVATION

The factors that motivated us to envisage the proposed scheme explained in this paper are as follows.

1) In order to achieve efficiency, the communication cost between the social network server (SNS) and the location-based server (LBS) should be as little as possible. Moreover, less message exchange would give an attacker less exposure to execute attacks in a wireless public channel.
2) The Location sharing mechanism should not depend on a third-party location-based server. This should be done to minimize the chance of privacy leakage and to minimize the establishment cost.
3) The location-based server (LBS) must not be able to discover the topological structures of users' social network. By collusion with the social network server, LBS should not be able to reveal users' social information.

B. RESEARCH CONTRIBUTIONS

The following contributions are made in this paper:

1) The location sharing scheme of the proposed scheme does not depend on any third-party location-based server. This eliminates the possibility of LBS to reveal the social network topology structure of a social user.
2) The proposed scheme integrates LBS and SNS into a set of single entity servers, thereby reducing their internal communication overhead.
3) The proposed scheme has the ability to resist various active and passive security attacks which are present in the existing schemes.
4) The location sharing mechanism is efficient, lightweight and secure. We avoid computation costly operations like bilinear pairing, elliptic curve cryptography, public key infrastructure (PKI), public key cryptography.
5) On top of informal security analysis, we validate security of the proposed scheme through formal security verification using random oracle, and through security simulation using ProVerif 1.93.

C. ORGANIZATION OF THE PAPER

The rest of the paper is as follows. Section II outlines the existing work in brief. Section III discusses mathematical preliminaries, which are necessary to set up the proposed scheme. The system architecture and threat model is explained in Section IV. Section V presents the proposed location sharing scheme for multiserver architecture in mOSNs. Section VI provides various formal security proofs along with informal security analysis. Section VII presents security validation using ProVerif 1.93 simulation

tool. Section VIII presents the computation and communication cost of the proposed scheme. Section IX presents a performance and security comparison of the proposed scheme with the other related existing schemes. Finally, Section X concludes the paper.

II. RELATED WORK

In the field of mOSNs, privacy and security issues have attracted a great deal of research focus. Hence, in recent years, many of privacy-preserving schemes have been proposed with their own merits and limitations. Earlier research focuses on privacy preserving schemes aimed at the achievement of at achievement of information privacy [15], user anonymity [16] and protection of location privacy [17].

In order to sustain location anonymity, a mobile device encrypts the current location before sending it to servers. K-anonymity for location privacy adopts the process of obfuscating the actual location of the user as proposed and used by [18] and [19]. The use of dummy location along with the real location is the next approach for location anonymity [20]. Location encryption is another very effective way to achieve location privacy protection [21]. The pseudonym methods [22], [23], mix zones [24] and the m-unobservability [25] are some well know schemes developed in the past. Rahman *et al.* obtained location obscurity through privacy context obfuscation based on various location parameters [26].

Location sharing while maintaining privacy protection in online social networks has been first primarily addressed in 2007 by SmokeScreen [27], which allowed sharing locations between social friends and strangers. Wei *et al.* enhanced this scheme and proposed Mobishare, where users' social and location information were separately stored into SNS and LBS respectively [28]. Mobishare suffers from the weakness that, in the query phrase, LBS can reveal the topology structure of social networks of a user. Recently, Li *et al.* [29] enhanced Mobishare to propose new privacy-protected location-sharing scheme in mOSNs, namely MobiShare+, which introduced the concept of dummy queries and private set intersection to prevent LBS from knowing social information of a user. BMobiShare is a improved version over MobiShare+ in terms of transmission efficiency, where the existing private set intersection method is replaced by Bloom Filter [30]. However, the computation cost of BMobiShare is quite high.

In 2015, in order to improve privacy-protection against the insider attack, Li *et al.* introduced a multiple location server based location sharing system [31]. Although it provides higher security, it is resource-demanding and time-inefficient. As these schemes rely on the third-party location server, they associate the chance of LBS to collude with SNS in order to reveal the social information. Also, they incur a high transmission and storage cost [28], [29], [32], [33], [30]. To address this issue, very recently, Xiao *et al.* proposed CenLocShare [34], where SNS and LBS were amalgamated into one single server. This scheme reduces communication cost, storage cost and also increases user's privacy protection.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

Remark 1: The implementation of end-to-end encryption is an open research problem to many popular OSNs. The CEO of Facebook has recently published an article "A Privacy-Focused Vision for Social Networking", which claims that the OSN giant is planning to implement end-to-end encryption on all its messaging services to increase privacy levels, and it has started experimenting with end-to-end encryption already [35]. The lack of privacy in OSNs leads to various security hazards like the identity theft, information leakage, and government impinge on user privacy [36]. However, the proposed scheme does not aim at providing complete end-to-end encryption on all messages between mobile user and the social media service provider.

The idea proposed in this paper serves three basic purposes. First, it provides centralized storage of location-based information and social information into single entity [34]. Second, it ensures secure communication of location sharing and update based messages, thus protecting them from various malicious attackers. Finally, for location sharing, it facilitates a low computation and communication cost on mobile device, as it avoids encryption via public key infrastructure (PKI). These make the proposed scheme suitable for practical environments.

We find that existing centralized location sharing of the scheme suffer from the man-in-the-middle attack, replay attack, and DoS attack [37]. Our contribution is to secure location sharing and location query based messages and to protect them from adversary. We do not exploit direct key sharing between the user and the service provider. As shown in Figure 3, MU_i goes through a three-factor authentication process with CT, and establish the session key $SK_{MU_i,CT}$ $(= SK_{CT,MU_i})$, shared with CT. All location-based messages between MU_i and CT are encrypted with this key. Similarly, Figure 4 shows how CT and $LSSNS$ authenticate and establish their shared session key $SK_{S_j,CT}(= SK_{CT,S_j})$.

III. MATHEMATICAL FUNDAMENTALS

To describe our proposed scheme, we have applied the collision-resistant one-way hash function [38], Chebyshev polynomial [39], [40], biometrics and fuzzy extractor, bitwise XOR operator. In this section, we describe these fundamental concepts in brief.

A. THE COLLISION-RESISTANT ONE-WAY HASH FUNCTION

The input to a one-way cryptographic hash function H: $\{0, 1\}^* \rightarrow \{0, 1\}^k$ is any string of 0 and 1. That is, $s \in \{0, 1\}^*$. The output of the function is another binary string $H(s) \in \{0, 1\}^k$ whose length is fixed k bits. The property of collision-resistant of $H(\cdot)$ is described in the following [41].

Definition 1: The advantage probability of any adversary $\mathcal{A}$'s to find any collision with the execution time t_n is denoted and defined by $Adv_{\mathcal{A}}^{HASH}(t_n) = Pr[(p, q) \in_R \mathcal{A}: p \neq q$ and $H(p) = H(q)]$, where $Pr[M]$ is the probability of an event M and an adversary $\mathcal{A}$ selects a random pair (p, q). By an

(ϵ, t_n)-adversary $\mathcal{A}$ attacks the collision resistance of $H(\cdot)$, it specify that the computation time of $\mathcal{A}$ is at most t_n and that $Adv_{\mathcal{A}}^{HASH}(t_n) \leq \epsilon$.

B. THE CHEBYSHEV POLYNOMIAL: DEFINITION AND PROPERTIES

The Chebyshev polynomial $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ of degree n is defined as [39]:

$$T_n(x) = \begin{cases} cos(n \cdot arccos(x)) & \text{if } x \in [-1, 1] \\ cos(n\theta) & \text{if } x = cos\theta, \ \theta \in [0, \pi]. \end{cases}$$

The Chebyshev polynomial can be expressed in terms of the following recurrecnce relation.

$$T_n(x) = \begin{cases} 1 & \text{when } n \text{ is equal to } 0 \\ x & \text{when } n \text{ is equal to } 1 \\ & \text{when } n \text{ is greater} \\ 2xT_{n-1}(x) - T_{n-2}(x) & \text{than or equal to } 2. \end{cases}$$

Definition 2: The semi-group property of the enhanced Chebyshev polynomial holds on the interval $(-\infty, +\infty)$ and is defined as follows [42]. $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ (mod p), where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime number. Here, $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x))$ (mod p), where $Z_p^* = \{a | 0 < a < p, \gcd(a, p) = 1\} = \{1, 2, \ldots, p - 1\}$.

Definition 3: For any given x and y, it is computationally infeasible to find an integer s such that $T_s(x) = y$. It is referred to as the Chaotic map-based discrete logarithm problem (CMDLP) [43]. The advantage probability of $\mathcal{A}$ to solve CMDLP is $Adv_{\mathcal{A}}^{CMDLP}(t_2) = Pr[\mathcal{A}(x, y) = r : r \in Z_p^*, y = T_r(x) \pmod{p}]$.

C. THE BIOMETRICS AND FUZZY EXTRACTOR

For secure authentication, various authentication protocols use some biometrics features, such as iris and fingerprint as the key for their uniqueness property [44], [45]. Using the Fuzzy extractor technique, we can produce the identical output string, though the input biometric will differ from the stored biometric samples up to a given threshold limit for permissible error tolerance. The Fuzzy extractor is defined by two algorithms: *Generate*($\cdot$) and *Reproduce*($\cdot$), which are deterministic and probabilistic.

Definition 4: Let us suppose that a biometric key of length n bits is generated from the biometrics $\mathcal{B}$. We also consider that $\mathcal{R} = \{0, 1\}^k$ is a metric space of finite dimensional biometric data points. The following two functions are defined next.

- *Generate: This function generates a pair (η, μ), where $\eta \in \{0, 1\}^n$ represents the biometric key and μ is a public value which is used as a parameter by the Reproduce function for a given input $\mathcal{B} \in \mathcal{R}$.*
- *Reproduce: This function regenerates the original biometric key $\eta = Reproduce(\mathcal{B}_i, \mu)$, where $\eta \in \{0, 1\}^n$ from the entered biometrics $\mathcal{B}_i$ and original biometrics $\mathcal{B}$ and $\mathcal{B}_i$ are close in terms of some distance metric such*
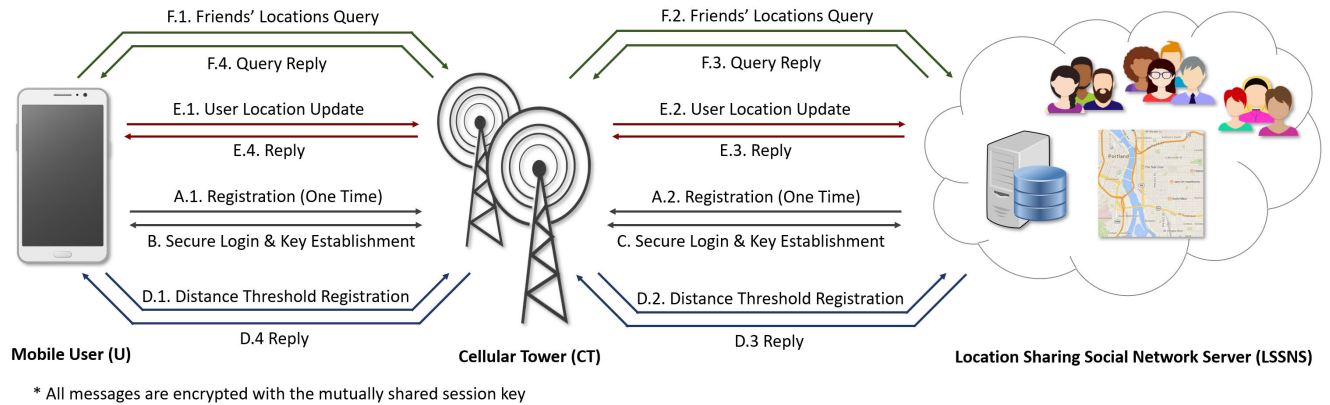
FIGURE 1. The architecture for location sharing in the mOSN through multiserver system.

as the Hamming distance. To be close, this distance must not to be more than $\mathcal{E}$*.* $\mathcal{E}$ *is a pre-defined threshold value.*

IV. THE ADVERSARY MODEL AND SYSTEM MODEL

This section briefly describes the basic attack model or adversary model applicable for our proposed scheme. Moreover, we depict the outline of the system model adopted for our proposed location sharing scheme for the online social network.

A. THE THREAT MODEL

We primarily assume that cellular tower (CT) is a trusted body and define the threat model concerning the location-sharing social network servers ($LSSNSs$) and the user (U). We define the model below:

- Registered entities like U, $LSSNS$ and CT communicate through a public insecure wireless channel. The proposed scheme adopts the widely-accepted Dolev-Yao threat model (DY model) [46]. An attacker or a malicious user has all the capabilities of executing all potential attacks defined in the classical DY model.
- A registered or authorized user or a privileged insider of the system may turn into a malicious user, who illegally intends to access various location or social information of other genuine users.
- $LSSNS$s exhibit an 'honest but curious' nature. They alone, or after colluding with other servers, try to retrieve the social network topology or location information of other registered users.
- Our proposed scheme assumes CT to be a trusted entity.

B. THE SYSTEM MODEL

Figure 1 shows the basic system model of the proposed scheme. Here, we define the basic entities, which are described as follows:

- **Mobile user** (U): Sends and responds to three types of request queries. These include sharing of location information to other social friends and strangers,

updating own location information and querying a friend's location information.
- **Location Sharing Social Network Servers** ($LSSNS$): Responsible for storing, updating and informing various location information of U.
- **Cellular Tower (CT)**:: It is a trusted entity, which receives, processes and forwards various messages of U and $LSSNS$. All messages communicated between U and $LSSNS$ are communicated via CT.

The overall flow of the model is shown in Figure 1. First, the mobile user and the Location Sharing Social Network Server $LSSNS_j$ register to a cellular tower CT (process A). This is a one-time operation and is executed through a secure channel. Next, the mobile user MU_i and $LSSNS_j$ make a secure login to the registered CT and establish a shared session key (processes B and C respectively). Thereafter, the mobile user registers a distance threshold to $LSSNS_j$ via CT, in which corresponding social friends can be searched (process D). When required, the mobile user updates his/her current location to $LSSNS_j$ through the cellular tower (process E). Finally, the mobile user obtains his/her social friends' identities and locations for those who are willing to share their information from $LSSNS_j$ through the cellular tower (process F).

In general, three major security challenges are primarily faced by location sharing schemes designed for mOSN applications. First, various location-based services must be privacy-preserving. An attacker or malicious user must not be able to access and/or modify personal information of U. Second, to ensure user location privacy, $LSSNS$ should store various fake or dummy identities of U. Finally, a physical distance threshold between U and friend or stranger of U must be registered. A location query about U's friends or strangers are processed only if their current physical distance is within that predefined distance threshold.

V. THE PROPOSED SCHEME

In order to design the proposed scheme, various symbols are used. The symbols and notations are tabulated in Table 1.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

TABLE 1. Symbols and notations used in the proposed scheme.

| Symbol	Description	Symbol	Description
MU_i	i^{th} mOSN user	ID_i	The identity of the ith user
CT	Cellular Tower	ID_{CT}	The identity of the CT
$LSSNS_j$	j^{th} Location Sharing Social Network Server	ID_{S_j}	The identity of the $LSSNS_j$
PW_i	The login password of MU_i	PW_{S_j}	The login password of $LSSNS_j$
$\mathcal{B}_i$	User biometric of MU_i	$MPWB_i$	Biometric embedded password of MU_i
TID_i	Temporary identity of MU_i	RID_i	The pseudo-identity of MU_i
RID_Γ	The pseudo-identity of CT	Ω_{u_i}	128-bits Random Variable
RN_{u_i}	128-bit random number chosen by MU_i	RN_{ct}	128-bit random number chosen by CT
X_Γ	1024-bit master secret key chosen by CT	$x_{CT_{U_i}}$	1024-bit secret key chosen by CT
$A_{U_{iCT}}$	Temporary Variable	$V_{CT_{U_i}}$	Temporary Variable
P_i^1	Temporary Variable used by MU_i	P_i^2	Temporary Variable used by MU_i
RPW_{S_j}	Masked password of $LSSNS_j$	C_j	Temporary masked password of $LSSNS_j$
E_1, E_2	Temporary Variables used by $LSSNS_j$	f_{S_j}	Temporary Variable used by $LSSNS_j$
PID_{S_j}	pseudo-identity of $LSSNS_j$	SN_j	Serial number of $LSSNS_j$
(x_{u_i}, y_{u_i})	The real location coordinate of MU_i	$Index_{u_i}$	The encrypted real index of user's location
SK	Symmetric key	$\mathcal{K}_{MU_i, \mathcal{F}}$	SK shared between MU_i and its friends $\mathcal{F}$
$\mathcal{SK}_{MU_i, CT}$	SK shared between MU_i and CT	$\mathcal{SK}_{CT, S_j}$	SK shared between CT and $LSSNS_j$
Θ	The set of social friends of MU_i	$\delta(\cdot)$	The euclidean distance function among users
$\mathcal{D}_{f_{u_i}}$	The registered distance threshold of MU_i	ds_{U_i}	Distance threshold for strangers' location query
$\mathcal{T}_{u_i}$	Timestamp generated by MU_i	$\mathcal{T}_{ct}$	Timestamp generated by CT
$H(\cdot)$	One way cryptographic hash function	$E_k(\cdot)/D_k(\cdot)$	Symmetric encryption/decryption using key k
$||, \oplus$	Concatenation, bitwise XOR operations	$A \to B : \langle M \rangle$	A sends message M to B via public channel
$\triangle T$	Maximum transmission delay	qf_{u_i}	Friend location query distance limit
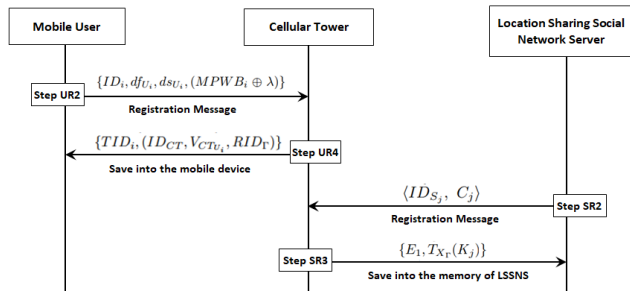
FIGURE 2. Message Communication in Registration Phase. Please refer to Sections V-A1 and V-A2.

A. THE REGISTRATION PHASE

This phase involves two distinct registration processes, namely, (a) the registration of a mOSN user (MU_i) to a cellular tower, and (b) the registration of a $LSSNS_j$ to a cellular tower. The registration process is a one-time operation that is executed through a secure channel; the message communication for this phase is shown in Figure 2.

1) MOBILE USER REGISTRATION

In this phase, a series of steps are executed for the registration of a mobile user MU_i to the CT. These steps are as follows.

Step UR1:

1) MU_i selects own identity, password, and biometrics as ID_i, PW_i, $\mathcal{B}_i$ respectively.
2) MU_i selects parameters n and λ, which are two 128-bit random numbers.

Step UR2:

1) MU_i uses the fuzzy extractor ($\cdot$) function to produce $(\eta_i, \mu_i) = Generation(\mathcal{B}_i)$ and computes the biometric embedded password $MPWB_i = H(ID_i|| H(PW_i|| \eta_i ||n))$.
2) Through a secure channel, MU_i delivers its registration message $\{ID_i, df_{U_i}, ds_{U_i}, (MPWB_i \oplus \lambda)\}$ to the CT.

Note that ID_i and PW_i are randomized by concatenating 128-bit (16-byte) random numbers [43], [47], [48]. We mask the user id and password as $MPWB_i = H(ID_i|| H(PW_i|| \eta_i ||n))$. Thus, guessing of ID_i and PW_i from $MPWB_i$ is infeasible, as it is computationally hard to guess three secrets simultaneously. An 128-bit random number can generate 10^{38} possible values (as $2^{128} \approx 10^{38}$). So, the guessing possibility is only $\approx \frac{1}{10^{38}}$ [47], [49].

Step UR3:

1) CT randomly selects its own 1024-bit master secret key X_Γ.
2) For each $\langle CT \leftrightarrow MU_i \rangle$ pair, CT randomly selects a 1024-bit secret key $x_{CT_{U_i}}$.
3) CT computes $A_{U_{iCT}} = H(H(ID_i \oplus x_{CT_{U_i}}) ||X_\Gamma)$, $V_{CT_{U_i}} = A_{U_{iCT}} \oplus MPWB_i$.
4) CT chooses its pseudo-identity as $RID_\Gamma = H(ID_{CT}||X_\Gamma)$.

Step UR4:

1) CT provides an anonymous temporary identity for each mOSN user MU_i. This is done by selecting a random but temporary identity TID_i for each user MU_i.

IEEE Access·

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

2) CT saves $m\ \langle CT \leftrightarrow MU_i\rangle$ key-plus-id combinations $\{TID_i, (ID_{CT}, V_{CT_{U_i}}, RID_\Gamma) \mid 1 \le j \le m\}$ in mobile device of MU_i.

Note that for all MU_is, the CT saves the record $\{ID_i, TID_i, x_{CT_{U_i}}\}$ in own database.

Step UR5:

1) MU_i computes $P_i^1 = H(PW_i || \eta_i) \oplus n$ and $P_i^2 = H(ID_i || PW_i || \eta_i || n)$.

2) MU_i modifies $V_{CT_{U_i}}$ as $V'_{CT_{U_i}} = V_{CT_{U_i}} \oplus \lambda$, $RID_i = TID_i \oplus H(ID_i || V'_{CT_{U_i}})$ and $RID'_\Gamma = RID_\Gamma \oplus H(\eta_i || n)$ for all $1 \le j \le m$.

3) MU_i stores parameters $\langle \mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}$s, RID_is and $RID'_\Gamma \rangle$ and removes $V_{CT_{U_i}}$s, RID_Γ and TID_is from own mobile device.

2) THE LOCATION SHARING SOCIAL NETWORK SERVER REGISTRATION PHASE

Each location sharing social network server $LSSNS_j$ registers to the cellular tower CT through the following steps:

Step SR1:

1) $LSSNS_j$ chooses own id and password as ID_{S_j} and PW_{S_j}.

2) It selects one random number b of 128-bit long.

Step SR2:

1) $LSSNS_j$ computes masked password $RPW_{S_j} = H(ID_{S_j} || PW_{S_j})$ and $C_j = H(ID_{S_j} || PW_{S_j} || b)$.

2) $LSSNS_j$ submits $\langle ID_{S_j}, C_j \rangle$ to CT via a secure channel.

Step SR3:

1) CT uses its master secret key X_Γ and one random number r (128-bit) to compute $K_j = H(H(ID_{S_j} || X_\Gamma) \oplus r)$, and $E_1 = K_j \oplus C_j = K_j \oplus H(ID_{S_j} || PW_{S_j} || b)$.

2) CT embeds the parameters $\{E_1, T_{X_\Gamma}(K_j)\}$ in memory of each $LSSNS_j$.

3) CT saves pair $\langle ID_{S_j}, SN_j, r \rangle$ into its database, where SN_j is the identity or serial number of the server $LSSNS_j$.

Step SR4:

1) $LSSNS_j$ computes $E_2 = RPW_{S_j} \oplus b$ and $f_{S_j} = H(RPW_{S_j} || b)$.

2) $LSSNS_j$ stores $E_2, f_{S_j}, H(\cdot)$ into its own memory.

The Summary of registration process of MU_i and $LSSNS_j$ to CT is shown in Figure 3.

B. THE mOSN USER LOGIN, AUTHENTICATION AND KEY ESTABLISHMENT PHASE

The mOSN user MU_i makes a secure login to the registered CT and establishes a shared session key through the following steps:

Step ULA1:

1) MU_i inputs own identity, password, and biomerics (noisy) as ID_i, PW_i, and $\mathcal{B}'_i$ respectively.

2) Using stored μ_i and P_i^1, MU_i computes $\eta_i = Reproduction(\mathcal{B}'_i, \mu_i)$ and generates $n' = P_i^1 \oplus H(PW_i || \eta_i)$.
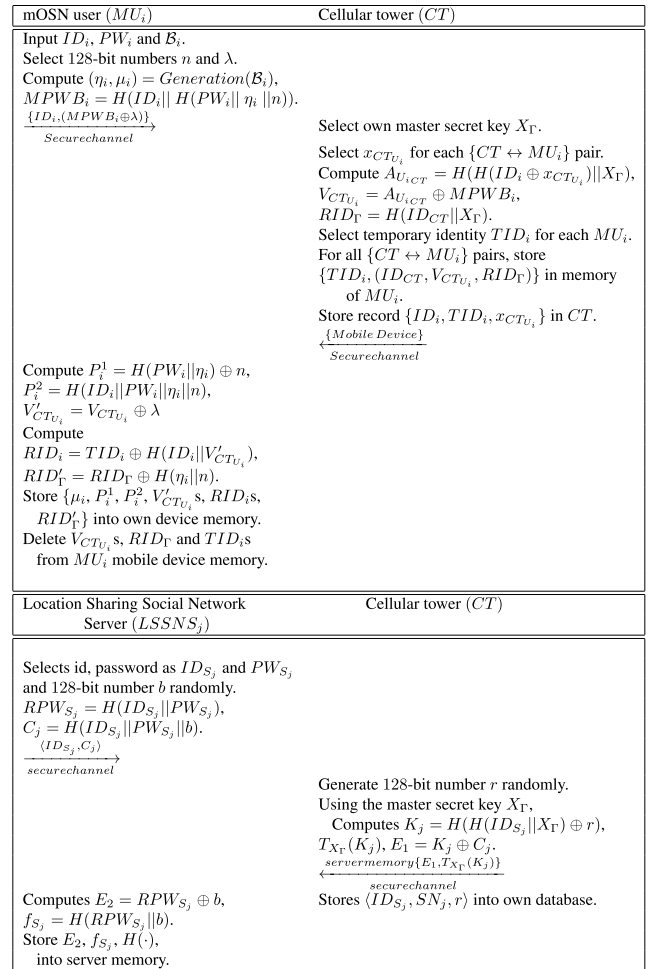

mOSN user (MU_i) — **Cellular tower (CT)**

Input ID_i, PW_i and $\mathcal{B}_i$.
Select 128-bit numbers n and λ.
Compute $(\eta_i, \mu_i) = Generation(\mathcal{B}_i)$,
$MPWB_i = H(ID_i || H(PW_i || \eta_i || n))$.

$\xrightarrow{\{ID_i, (MPWB_i \oplus \lambda)\}}$
$Secure\,channel$

Select own master secret key X_Γ.

Select $x_{CT_{U_i}}$ for each $\{CT \leftrightarrow MU_i\}$ pair.
Compute $A_{U_{iCT}} = H(H(ID_i \oplus x_{CT_{U_i}}) || X_\Gamma)$,
$V_{CT_{U_i}} = A_{U_{iCT}} \oplus MPWB_i$,
$RID_\Gamma = H(ID_{CT} || X_\Gamma)$.
Select temporary identity TID_i for each MU_i.
For all $\{CT \leftrightarrow MU_i\}$ pairs, store
$\{TID_i, (ID_{CT}, V_{CT_{U_i}}, RID_\Gamma)\}$ in memory of MU_i.
Store record $\{ID_i, TID_i, x_{CT_{U_i}}\}$ in CT.

$\xleftarrow{\{Mobile\ Device\}}$
$Secure\,channel$

Compute $P_i^1 = H(PW_i || \eta_i) \oplus n$,
$P_i^2 = H(ID_i || PW_i || \eta_i || n)$,
$V'_{CT_{U_i}} = V_{CT_{U_i}} \oplus \lambda$
Compute
$RID_i = TID_i \oplus H(ID_i || V'_{CT_{U_i}})$,
$RID'_\Gamma = RID_\Gamma \oplus H(\eta_i || n)$.
Store $\{\mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}$s, RID_is, $RID'_\Gamma\}$ into own device memory.
Delete $V_{CT_{U_i}}$s, RID_Γ and TID_is from MU_i mobile device memory.

Location Sharing Social Network Server ($LSSNS_j$) — **Cellular tower (CT)**

Selects id, password as ID_{S_j} and PW_{S_j} and 128-bit number b randomly.
$RPW_{S_j} = H(ID_{S_j} || PW_{S_j})$,
$C_j = H(ID_{S_j} || PW_{S_j} || b)$.

$\xrightarrow{\langle ID_{S_j}, C_j \rangle}$
$secure\,channel$

Generate 128-bit number r randomly.
Using the master secret key X_Γ,
Computes $K_j = H(H(ID_{S_j} || X_\Gamma) \oplus r)$,
$T_{X_\Gamma}(K_j)$, $E_1 = K_j \oplus C_j$.

$\xleftarrow{server\,memory\{E_1, T_{X_\Gamma}(K_j)\}}$
$secure\,channel$

Computes $E_2 = RPW_{S_j} \oplus b$,
$f_{S_j} = H(RPW_{S_j} || b)$.
Store $E_2, f_{S_j}, H(\cdot)$,
into server memory.

Stores $\langle ID_{S_j}, SN_j, r \rangle$ into own database.

FIGURE 3. The Registration phases of *MU_i* and *$LSSNS_j$* in the proposed scheme.

3) MU_i calculates $H(ID_i || PW_i || n' || \eta_i)$ and compares with stored P_i^2.

4) If the verification succeeds, go to *Step ULA2*, else, *exit*.

Step ULA2:

1) MU_i randomly generates Ω_{u_i} (128-bit number).

2) Using stored parameter $V'_{CT_{U_i}}$, MU_i computes:

   a) $MPWB_i = H(ID_i || H(PW_i || n' || \eta_i))$.

   b) $A_{U_{iCT}} = V'_{CT_{U_i}} \oplus MPWB_i$.

   c) $\mathcal{M}_1 = A_{U_{iCT}} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$.

   d) $TID_i = RID_i \oplus H(ID_i || V'_{CT_{U_i}})$.

   e) $TID_i^* = TID_i \oplus H(ID_{CT} || \mathcal{T}_{u_i})$

3) MU_i uses the current login timestamp $\mathcal{T}_{u_i}$ and computes a hash value $H_1 = H(ID_i || \mathcal{M}_1 || \Omega_{u_i} || \mathcal{T}_{u_i})$.

4) Through a public channel, MU_i sends $\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to CT.

Step ULA3:

1) CT verifies if $|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}| \overset{?}{\le} \Delta T$. If verification holds go to step 2, else *exit*.

2) CT calculates $TID_i = TID_i^* \oplus H(ID_{CT} || \mathcal{T}_{u_i})$.

3) Corresponding to calculated TID_i, CT finds the record $\langle \{ID_i, x_{CT_{U_i}}\} \rangle$ from own database.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

4) CT computes $B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})||X_\Gamma)$.

5) CT computes $\mathcal{P}_1 = \mathcal{M}_1 \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}) \oplus B_{CT_{U_i}} = \Omega_{u_i}$.

Note that CT obtains $\mathcal{P}_1$ of step (5), as $A_{U_iCT} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})|| X_\Gamma)$.

Step ULA4:

1) CT uses received parameters to prepares a hash value $H_2 = H(ID_i ||\mathcal{M}_1 ||\mathcal{P}_1 ||\mathcal{T}_{u_i})$.

2) CT verifies if $H_2 \overset{?}{=} H_1$. If verification holds go to step 3, else *exit*.

3) CT saves the record $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i} \rangle$ in its database.

4) CT generates a 128-bit random number Ω_{ct}.

5) CT computes $\mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$. Here $\mathcal{T}_{ct}$ is the current timestamp of CT.

6) CT computes shared session key $SK_{CT,MU_i} = H(ID_i ||ID_{CT} ||B_{CT_{U_i}} ||\mathcal{P}_1 ||\Omega_{ct} ||\mathcal{T}_{u_i} ||\mathcal{T}_{ct})$.

7) CT prepares hash value $H_3 = H(ID_i ||\mathcal{P}_1 ||\Omega_{ct} ||\mathcal{T}_{u_i} ||\mathcal{T}_{ct} ||SK_{CT,MU_i})$.

8) Through public channel, CT sends authentication response message $\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i.

Step ULA5:

1) MU_i receives an authentication response message from step 8 of $ULA4$.

2) MU_i verifies the transmission delay by comparing received and current timestamps. Go to step 3, if verification holds, else *exit*.

3) MU_i computes $\mathcal{P}_2 = \mathcal{M}_2 \oplus \mathcal{T}_{ct} \oplus ID_i \oplus A_{U_iCT} = \Omega_{ct}$.

Note that we obtain $\mathcal{P}_2$ of step (3), as $A_{U_iCT} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) ||X_\Gamma)$.

Step ULA6

1) MU_i generates a session key (mutually shared with CT) as $SK_{MU_i,CT} = H(ID_i ||ID_{CT} ||A_{U_iCT} ||\Omega_{u_i} ||\mathcal{P}_2 ||\mathcal{T}_{u_i} ||\mathcal{T}_{ct})$.

2) MU_i computes final hash value $H_4 = H(ID_i ||\Omega_{u_i} ||\mathcal{P}_2 ||\mathcal{T}_{u_i} ||\mathcal{T}_{ct} ||SK_{MU_i,CT})$.

3) If $H_4 \overset{?}{=} H_3$, then MU_i confirms that the session key $SK_{MU_i,CT} (= SK_{CT,MU_i})$ is mutually verified and established. Else, MU_i discards the session key and terminates the process.

For all message communicaions in the current session, MU_i and CT use this key for message encryption.

The Summary of mOSN user login, authentication and key establishment phase with CT is shown in Figure 5 and the message communications of Login, Authentication and Key Establishment Phase is shown in Figure 4.

C. THE $LSSNS_j$ LOGIN, AUTHENTICATION AND KEY ESTABLISHMENT PHASE

The $LSSNS_j$ makes a secure login to the registered cellular tower CT and establishes a shared session key through the following steps:

Step SLA1:

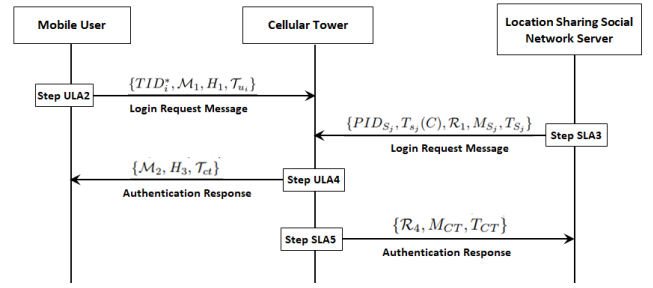1) $LSSNS_j$ inputs own id ID_{S_j} and password PW_{S_j}.

FIGURE 4. Message Communication in Login, Authentication and Key Establishment Phase. Please refer to Sections V-B and V-C.

2) $LSSNS_j$ generates $RPW_1 = H(ID_{S_j}||PW_{S_j})$ and $b_1 = E_2 \oplus PWB_1$.

3) $LSSNS_j$ uses generated b_1 and computes $f'_{S_j} = H(RPW_1 ||b_1)$.

4) $LSSNS_j$ verifies if $f_{S_j} \overset{?}{=} f'_{S_j}$. If verification holds, go to step $SLA2$, else *exit*.

Step SLA2:

1) $LSSNS_j$ computes $C = E_1 \oplus H(ID_{S_j}|| PW_{S_j} ||b_1)$.

2) $LSSNS_j$ generates random number s_j.

3) $LSSNS_j$ computes $T_{s_j}(C)$.

4) $LSSNS_j$ computes $K_1 = T_{s_j}(T_{X_\Gamma}(C))$.

Step SLA3:

1) $LSSNS_j$ generates 128-bit random number Ω_{s_j}.

2) $LSSNS_j$ computes $\mathcal{R}_1 = C \oplus \Omega_{s_j} \oplus T_{s_j}(C) \oplus T_{S_j}$. Here, T_{S_j} is the current timestamp of $LSSNS_j$.

3) $LSSNS_j$ generates its pseudo identity $PID_{S_j} = ID_{S_j} \oplus H(K_1)$.

4) $LSSNS_j$ computes $M_{S_j} = H(ID_{S_j}|| C ||K_1 ||\Omega_{s_j}|| T_{S_j})$

5) Finally, through a public channel, $LSSNS_j$ sends its login request $\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}$ to the cellular tower CT.

Step SLA4:

1) CT receives login message and verifies if $|T^*_{S_j} - T_{S_j}| \overset{?}{\leq} \Delta T$. If verification holds go to step 2, else *exit*. Here, $T^*_{S_j}$ is the current timestamp.

2) CT calculates $K'_1 = T_{X_\Gamma}(T_{s_j}(C))$.

3) CT calculates $ID'_{S_j} = PID_{S_j} \oplus H(K'_1) = ID_{S_j} \oplus H(K_1) \oplus H(K'_1) = ID_{S_j}$.

4) CT verifies if $K'_1 \overset{?}{=} K_1$. If verification holds, CT ensures that $ID'_{S_j} = ID_{S_j}$ and go to step 5.

5) CT finds the record $\langle ID_{S_j}, SN_j, r \rangle$ in the database.

6) CT further computes $C' = H(H(ID'_{S_j}||X_\Gamma) \oplus r)$.

7) CT computes $\mathcal{R}_2 = \mathcal{R}_1 \oplus T_{S_j} \oplus C' \oplus T_{s_j}(C) = H(H(ID_{S_j} ||X_\Gamma) \oplus r) \oplus\Omega_{s_j} \oplus T_{S_j} \oplus T_{s_j}(C)) \oplus T_{S_j} \oplus H(H(ID_{S_j} ||X_\Gamma) \oplus r) \oplus T_{s_j}(C) = \Omega_{s_j}$.

8) CT uses the received parameters T_{S_j} and calculates $\mathcal{R}_3 = H(ID'_{S_j}||C'||K'_1||\mathcal{R}_2||T_{S_j})$.

9) CT verifies whether $\mathcal{R}_3 \overset{?}{=} \mathcal{R}_2$.

10) On successful verification, CT accepts the login request and considers the Location Server $LSSNS_j$ as

mOSN user (MU_i)	Cellular Tower (CT)

Login phase

Input ID_i, PW_i, and $\mathcal{B}'_i$.
Compute
$\eta_i = Reproduction(\mathcal{B}'_i, \mu_i)$
$n' = P_i^1 \oplus H(PW_i \| \eta_i)$.
Verifies if stored $P_i^2 = H(ID_i \| PW_i \| \text{n'} \| \eta_i)$?
If verification holds,
Generate 128-bit number Ω_{u_i}
Compute
$MPWB_i = H(ID_i \| H(PW_i \| n' \| \eta_i))$
$A_{U_{i\,CT}} = V'_{CT_{U_i}} \oplus MPWB_i$
$\mathcal{M}_1 = A_{U_{i\,CT}} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$
$TID_i = RID_i \oplus H(ID_i \| V'_{CT_{U_i}})$
$TID_i^* = TID_i \oplus H(ID_{CT} \| \mathcal{T}_{u_i})$
$H_1 = H(ID_i \| \mathcal{M}_1 \| \Omega_{u_i} \| \mathcal{T}_{u_i})$

$\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$
$\xrightarrow{\hspace{3cm}}$
(public channel)

Authentication phase

(Cellular Tower side)

Verify if $|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}| \leq \triangle T$?
Compute
$\quad TID_i = TID_i^* \oplus H(ID_{CT} \| \mathcal{T}_{u_i})$,
Corresponding to TID_i,
$\quad$ Find the record $\langle \{ID_i, x_{CT_{U_i}}\} \rangle$ from own database.
Compute
$\quad B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \| X_\Gamma)$.
$\quad \mathcal{P}_1 = \mathcal{M}_1 \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}) \oplus B_{CT_{U_i}}$.
$\quad = \Omega_{u_i}$, as $A_{U_{i\,CT}} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \| X_\Gamma)$,
$\quad H_2 = H(ID_i \| \mathcal{M}_1 \| \mathcal{P}_1 \| \mathcal{T}_{u_i})$.
Verify if $H_2 = H_1$?
If verification holds, accepts the user login request.
Save the record $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i} \rangle$ in its database.
Generate 128-bit random number Ω_{ct}
Compute
$\quad \mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$,
Compute session key as
$\quad SK_{CT,MU_i} = H(ID_i \| ID_{CT} \| B_{CT_{U_i}} \| \mathcal{P}_1 \| \Omega_{ct} \| \mathcal{T}_{u_i} \| \mathcal{T}_{ct})$,
Compute hash value as
$\quad H_3 = H(ID_i \| \mathcal{P}_1 \| \Omega_{ct} \| \mathcal{T}_{u_i} \| \mathcal{T}_{ct} \| SK_{CT,MU_i})$.

$\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$
$\xleftarrow{\hspace{3cm}}$
(public channel)

(mOSN user side)

Verify if $|\mathcal{T}_{ct}^* - \mathcal{T}_{ct}| \leq \triangle T$?
Compute
$\quad \mathcal{P}_2 = \mathcal{M}_2 \oplus \mathcal{T}_{ct} \oplus ID_i \oplus A_{U_{i\,CT}}$,
$\quad = \Omega_{ct}$, as $A_{U_{i\,CT}} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \| X_\Gamma)$.
Compute session key as
$\quad SK_{MU_i,CT} = H(ID_i \| ID_{CT} \| A_{U_{i\,CT}} \| \Omega_{u_i} \| \mathcal{P}_2 \| \mathcal{T}_{u_i} \| \mathcal{T}_{ct})$
Compute hash value
$H_4 = H(ID_i \| \Omega_{u_i} \| \mathcal{P}_2 \| \mathcal{T}_{u_i} \| \mathcal{T}_{ct} \| SK_{MU_i,CT})$.
Verify if $H_4 = H_3$?
If verification holds,
store session key $SK_{MU_i,CT}$ ($= SK_{CT,MU_i}$). | **Store session key SK_{CT,MU_i} ($= SK_{MU_i,CT}$).**

FIGURE 5. mOSN user login, authentication and key establishment phase in the proposed scheme.

authentic. Otherwise, *CT* terminates the session and *exit*.

Step SLA5:

1) *CT* selects a 128-bit random number Ω_{CT}.
2) *CT* computes $\mathcal{R}_4 = C' \oplus \Omega_{CT} \oplus T_{CT} = H(H(ID_{S_j} \| X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT}$. Here, T_{CT} is the current timestamp of *CT*.

3) *CT* computes the mutually shared session key $SK_{CT,S_j} = H(C' \| K'_1 \| T_{S_j} \| T_{CT} \| \mathcal{R}_2 \| \Omega_{CT})$.
4) *CT* computes $M_{CT} = H(ID_{S_j} \| SK_{CT,S_j} \| \mathcal{R}_2 \| \Omega_{CT} \| T_{S_j} \| T_{CT})$.
5) Through a public channel, *CT* sends the authentication response message $\{\mathcal{R}_4, M_{CT}, T_{CT}\}$ to $LSSNS_j$.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

Location Sharing Social Network Server ($LSSNS_j$)	Cellular tower (CT)
Login phase	
Input ID_{S_j} and PW_{S_j} Compute $\quad RPW_1 = H(ID_{S_j}||PW_{S_j})$ $\quad b_1 = E_2 \oplus PWB_1$ Verifies if stored $f_{S_j} = H(RPW_1||b_1)$? If verification holds, Generate 128-bit number s_j and Ω_{s_j} Compute $\quad C = E_1 \oplus H(ID_{S_j}|| PW_{S_j}||b_1)$ $\quad K_1 = T_{s_j}(T_{X_\Gamma}(C))$ $\quad \mathcal{R}_1 = C \oplus \Omega_{s_j} \oplus T_{s_j}(C) \oplus T_{S_j}$ Generate pseudo-identity $\quad PID_{S_j} = ID_{S_j} \oplus H(K_1)$ Compute $M_{S_j} = H(ID_{S_j}|| C ||K_1 ||\Omega_{s_j}|| T_{S_j})$ $\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}$ $\xrightarrow{\hspace{3cm}}$ **(public channel)**	
Authentication phase	
	Verify if $|T^\star_{S_j} - T_{S_j}| \leq \triangle T$? Compute $\quad K'_1 = T_{X_\Gamma}(T_{s_j}(C))$, $\quad ID'_{S_j} = PID_{S_j} \oplus H(K'_1) = ID_{S_j}$, Corresponding to ID_{S_j}, $\quad$ Find the record $\langle ID_{S_j}, SN_j, r\rangle$ from own database. Compute $\quad C' = H(H(ID'_{S_j}||X_\Gamma) \oplus r)$. $\quad \mathcal{R}_2 = \mathcal{R}_1 \oplus T_{S_j} \oplus C' \oplus T_{s_j}(C)$, $\quad = H(H(ID_{S_j}||X_\Gamma) \oplus r) \oplus \Omega_{s_j} \oplus T_{S_j} \oplus T_{s_j}(C))$ $\quad\quad \oplus T_{S_j} \oplus H(H(ID_{S_j}||X_\Gamma) \oplus r) \oplus T_{s_j}(C)$, $\quad = \Omega_{s_j}$. Using received T_{S_j}, Compute $\quad \mathcal{R}_3 = H(ID'_{S_j}||C'||K'_1||\mathcal{R}_2||T_{S_j})$ Verify if $\mathcal{R}_3 \stackrel{?}{=} \mathcal{R}_2$ If verification holds, $\quad CT$ accepts the login request. Generate 128-bit random number Ω_{CT} Compute $\quad \mathcal{R}_4 = C' \oplus \Omega_{CT} \oplus T_{CT}$, $\quad = H(H(ID_{S_j}||X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT}$ Compute session key as $\quad SK_{CT,S_j} = H(C'|| K'_1|| T_{S_j}|| T_{CT}|| \mathcal{R}_2|| \Omega_{CT})$, Compute hash value M_{CT} as $\quad M_{CT} = H(ID_{S_j}|| SK_{CT,S_j}|| \mathcal{R}_2|| \Omega_{CT}|| T_{S_j}|| T_{CT})$. $\xleftarrow{\{\mathcal{R}_4, M_{CT}, T_{CT}\}}$ **(public channel)**
Verify if $|T^\star_{CT} - T_{CT}| \leq \triangle T$? Compute $\mathcal{R}_5 = C' \oplus \mathcal{R}_4 \oplus T_{CT}$ $\quad = H(H(ID'_{S_j}||X_\Gamma) \oplus r) \oplus H(H(ID_{S_j}||X_\Gamma) \oplus r)$ $\quad\quad \oplus \Omega_{CT} \oplus T_{CT} \oplus T_{CT}$ $\quad = \Omega_{CT}$, Compute session key as $\quad SK_{S_j,CT} = H(C||K_1||T_{S_j}||T_{CT}||\Omega_{CT}||\mathcal{R}_5)$ Compute and verify hash value $\quad M_{CT} \stackrel{?}{=} H(ID_{S_j}||SK_{S_j,CT}||\Omega_{CT}||\mathcal{R}_5||T_{S_j}||TS_2)$. If verification holds, $\quad$ store session key $SK_{S_j,CT}(= SK_{CT,S_j})$	Store session key $SK_{CT,S_j} (= SK_{S_j,CT})$.

FIGURE 6. The $LSSNS_j$ login, authentication and key establishment phases of the proposed scheme.

Step SLA6:

1) $LSSNS_j$ receives the authentication response message from CT.

2) $LSSNS_j$ verifies the transmission delay $|T^*_{CT} - T_{CT}| \stackrel{?}{\leq} \triangle T$, where T^*_{CT} is the current timestamp. If verification holds, go to step 3, else *exit*.

3) $LSSNS_j$ computes $\mathcal{R}_5 = C' \oplus \mathcal{R}_4 \oplus T_{CT} = H(H(ID'_{S_j}||X_\Gamma) \oplus r) \oplus H(H(ID_{S_j}||X_\Gamma) \oplus r) \oplus \Omega_{CT} \oplus T_{CT} \oplus T_{CT} = \Omega_{CT}$.

Step SLA7:

1) $LSSNS_j$ generates the session key mutually shared with CT as $SK_{S_j,CT} = H(C ||K_1 ||T_{S_j} ||T_{CT} ||\Omega_{CT} ||\mathcal{R}_5)$.

2) $LSSNS_j$ verifies $M_{CT} \stackrel{?}{=} H(ID_{S_j} ||SK_{S_j,CT} ||\Omega_{CT} ||\mathcal{R}_5 ||T_{S_j} ||T_{CT})$.

3) If verification succeeds, $LSSNS_j$ confirms that the cellular tower CT is authentic and the current session key $SK_{S_j,CT} (= SK_{CT,S_j})$ is mutually verified and established. Otherwise, discard the session key and *exit*.

The summary of the $LSSNS_j$ login, authentication and key establishment phase is shown in Figure 6.
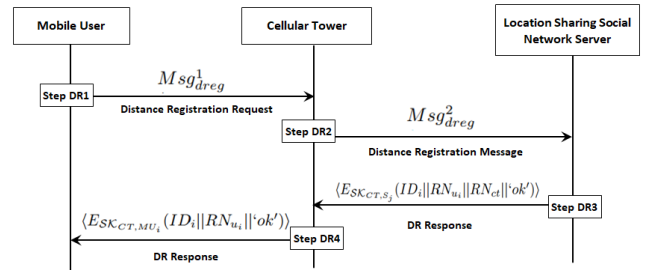

FIGURE 7. Message Communication in Distance Threshold Registration Phase refer to Section V-D.

D. THE DISTANCE THRESHOLD REGISTRATION PHASE

Every registered mOSN user MU_i needs to register a distance threshold to $LSSNS_j$ in which corresponding social friends can be searched and the message communications of Distance Threshold Registration Phase is shown in Figure 7.

Step DR1:

1) MU_i decides a distance threshold $\mathcal{D}_{f_{u_i}}$ beyond which MU_i does not allow his/her social friends to find himself in a friends' location query.

2) MU_i sends encrypted distance registration message $Msg^1_{dreg} = \langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i|| \mathcal{D}_{f_{u_i}}|| RN_{u_i}|| TS_{u_i}|| \mathcal{R}_{flag} = 1), H(ID_i|| RN_{u_i}|| TS_{u_i}), TS_{u_i}\rangle$.

Here, RN_{u_i}, TS_{u_i}, $H(\cdot)$ and $E(\cdot)$ convey their meaning as tabulated in Table 1. $\mathcal{R}_{flag} = 1$ indicates that this message in intended for the distance threshold registration.

Step DR2:

1) CT uses session key $\mathcal{SK}_{CT,MU_i}$ and decrypts $D_{\mathcal{SK}_{CT,MU_i}}(Msg^1_{dreg})$.

2) CT verifies if $|TS^*_{u_i} - TS_{u_i}| \stackrel{?}{\leq} \Delta T$, where $TS^*_{u_i}$ is the current timestamp. If verification holds, go to step 3, else terminate and *exit*.

3) CT computes the hash value $H(ID_i|| RN_{u_i}|| TS_{u_i})$. If the computed and received hash values are same, then go to step 4, else discards the message and *exit*.

4) CT logs in to $LSSNS_j$ and establishes the shared session key $\mathcal{SK}_{CT,S_j}$ as explained in subsection V-C.

5) CT encrypts and sends the distance registration message as $Msg^2_{dreg} = \langle E_{\mathcal{SK}_{CT,S_j}}(ID_i|| \mathcal{D}_{f_{u_i}}|| RN_{u_i}|| RN_{ct}|| TS_{ct}| \mathcal{R}_{flag} = 1), H(ID_i|| RN_{ct}|| TS_{ct}), TS_{ct}\rangle$ to $LSSNS_j$.

Step DR3:

1) $LSSNS_j$ decrypts Msg^2_{dreg} using the session key $\mathcal{SK}_{S_j,CT}$.

2) $LSSNS_j$ verifies communication delay using the received and current timestamp values.

3) $LSSNS_j$ verifies message integrity and authenticity by computing and comparing hash values with decrypted parameters.

4) If verifications of steps (2) and (3) are successful, go to Step 5, else terminate the session and *exit*.

5) $LSSNS_j$ saves record $\{ID_i, \mathcal{D}_{f_{u_i}}\}$ and sends response message $Msg^1_{resp} = E_{\mathcal{SK}_{CT,S_j}}(ID_i ||RN_{u_i} ||RN_{ct} ||`ok')$ to CT.
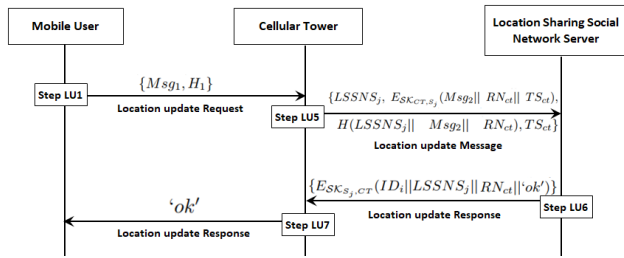
FIGURE 8. Message communication in user location update phase. Please refer to Section V-E.

Step DR4:

1) CT decrypts Msg_{resp}^1 using the shared session key.
2) CT verifies the received random number RN_{ct} and sends $Msg_{resp}^2 = E_{SK_{CT,MU_i}}(ID_i||RN_{u_i}||{}'ok')$ to MU_i.
3) MU_i decrypts the message Msg_{resp}^2 using the shared session key $SK_{MU_i,CT}$
4) MU_i verifies the random number RN_{u_i}. If these verification holds, go to step 5.Otherwise, termiate the session and *exit*.
5) MU_i reads 'ok' message and distance registration process successfully terminates.

E. THE USER LOCATION UPDATE PHASE

In this subsection, we describe how mOSN user MU_i updates his current location to the Location Sharing Social Network Server $LSSNS_j$ and the message communications of User Location Update Phase is shown in Figure 8. The location updation is done through the cellular tower CT, following the steps as mentioned in subsection V-B, MU_i makes a secure login to CT and mutually estabsishes a shared session key $SK_{MU_i,CT}(= SK_{CT,MU_i})$. Next, it executes the following steps:

Step LU1:

1) MU_i selects a one-time 128-bit random number RN_{u_i}.
2) MU_i uses the shared session key $SK_{MU_i,CT}$ and sends an encrypted message $Msg_1 = E_{SK_{MU_i,CT}}(ID_i|| x_{u_i}|| y_{u_i}|| E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})|| RN_{u_i}|| TS_{u_i})$ to CT.
3) MU_i sends a hash value $H_1 = H(ID_i|| x_{u_i}|| y_{u_i}|| E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})|| RN_{u_i}|| TS_{u_i})$ to CT.

Note that, (x_{u_i}, y_{u_i}) is the current location of MU_i, and $E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})$ is the current location of MU_i encrypted with the symmetric key $\{K_{MU_i,F}\}$, mutually shared between MU_i and all trusted friend F.

Step LU2:

1) CT receives location update message $\{Msg_1, H_1\}$ from MU_i.
2) CT uses SK_{CT,MU_i} and decrypts Msg_1 as $D_{SK_{CT,MU_i}}(Msg_1)$.
3) CT retrieves parameters ID_i, (x_{u_i}, y_{u_i}), $E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})$, RN_{u_i} and TS_{u_i} respectively (from step 2).
4) CT verifies if $|TS_{u_i}^* - TS_{u_i}| \overset{?}{\leq} \Delta T$, where $TS_{u_i}^*$ is the current timestamp. If the verification holds then go to step $LU3$, else discards the received message and *exit*.

Step LU3:

1) CT uses the decrypted parameter and computes a hash value $H_2 = H(ID_i|| x_{u_i}|| y_{u_i}|| E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})|| RN_{u_i}|| TS_{u_i})$.
2) CT verifies if $H_2 \overset{?}{=} H_1$. If the verification holds then go to step 3, else terminate the session and *exit*.
3) CT confirms the authenticity and integrity of the message and makes a login to $LSSNS_j$.
4) CT and $LSSNS_j$ establishes a mutually shared session key SK_{CT,S_j} as mentioned in subsection V-C.

Step LU4:

1) CT generates $\mathcal{L} - 1$ dummy locations and $\mathcal{L} - 1$ dummy encrypted string chosen randomly as $\{x_i^*, y_i^*, enc_i^*\}_{i=1...\mathcal{L}-1}$.
2) CT randomly put MU_i's real updated location information string at the nth place among the dummy information set, $(1 \leq n \leq \mathcal{L})$.
3) The sequence number of MU_i's real location update information is encrypted by CT with its own master secret key X_Γ, i.e., $Index_{u_i} = E_{X_\Gamma}[Sequence(x_{u_i}|| y_{u_i}|| E_{K_{MU_i,F}}(x_{u_i}, y_{u_i}))]$.

Step LU5:

1) CT prepares the message $Msg_2 = \{ID_i, (x_1^*, y_1^*, enc_1^*), \ldots, (x_{u_i}, y_{u_i}, E_{K_{MU_i,F}}(x_{u_i}, y_{u_i})), \ldots, (x_\mathcal{L}^*, y_\mathcal{L}^*, enc_\mathcal{L}^*), Index_{u_i}\}$.
2) CT generates a 128-bit random number RN_{ct}.
3) sends $\{LSSNS_j, E_{SK_{CT,S_j}}(Msg_2|| RN_{ct}|| TS_{ct}), H(LSSNS_j|| Msg_2|| RN_{ct}), TS_{ct}\}$ to server $LSSNS_j$.

Step LU6:

1) $LSSNS_j$ uses its session key $SK_{S_j,CT}$ (shared with CT) and decrypts the message Msg_2, random number RN_{ct}, and timestamp TS_{ct}.
2) $LSSNS_j$ checks the transmission delay using the received and current timestamps.
3) $LSSNS_j$ checks message integrity by checking computing a fresh hash value from the decrypted parameters.
4) $LSSNS_j$ updates the user location and sends $Msg_3 = E_{SK_{S_j,CT}}(ID_i||LSSNS_j||RN_{ct}||{}'ok')$ to CT.

Step LU7:

1) CT uses the session key SK_{CT,S_j} and decrypts Msg_3.
2) CT verifies the correctness of RN_{ct}. If it is correct, go to step 3, else terminate the session and *exit*.
3) CT forwards 'ok' to MU_i.

The user location update phase is summarized in Figure 9.

Remark 2: When the user reaches a new place, he/she updates his/her location in the LSSNS's database to ensure that LSSNS knows the user's real-time location. MU_i executes the user location update phase and sends the current location coordinate (x_{u_i}, y_{u_i}) (obtained by GPS) to LSSNS. As the user location update phase of our proposed scheme is based only on the private key encryption, cryptographic hash function and xor operation, it is both secure and lightweight. Table 4 reveals that considering all the entities, this location
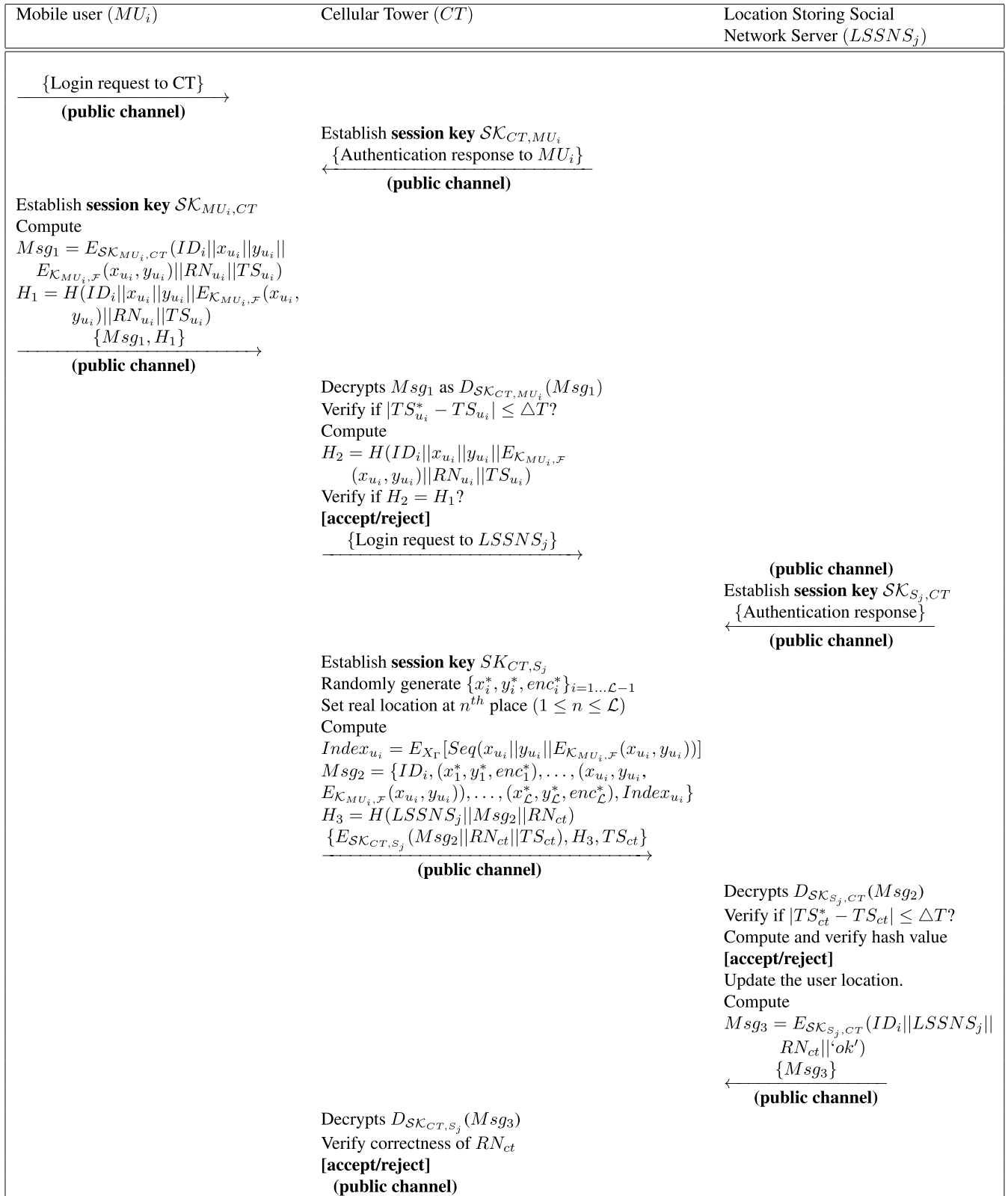
M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

Mobile user (MU_i)	Cellular Tower (CT)	Location Storing Social Network Server $(LSSNS_j)$

$\{$Login request to CT$\}$
$\xrightarrow{\hspace{3cm}}$
(public channel)

Establish **session key** $\mathcal{SK}_{CT,MU_i}$
$\{$Authentication response to $MU_i\}$
$\xleftarrow{\hspace{3cm}}$
(public channel)

Establish **session key** $\mathcal{SK}_{MU_i,CT}$
Compute
$Msg_1 = E_{\mathcal{SK}_{MU_i,CT}}(ID_i||x_{u_i}||y_{u_i}||$
$\quad E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i})||RN_{u_i}||TS_{u_i})$
$H_1 = H(ID_i||x_{u_i}||y_{u_i}||E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},$
$\quad y_{u_i})||RN_{u_i}||TS_{u_i})$
$\{Msg_1, H_1\}$
$\xrightarrow{\hspace{3cm}}$
(public channel)

Decrypts Msg_1 as $D_{\mathcal{SK}_{CT,MU_i}}(Msg_1)$
Verify if $|TS_{u_i}^* - TS_{u_i}| \leq \triangle T$?
Compute
$H_2 = H(ID_i||x_{u_i}||y_{u_i}||E_{\mathcal{K}_{MU_i,\mathcal{F}}}$
$\quad (x_{u_i}, y_{u_i})||RN_{u_i}||TS_{u_i})$
Verify if $H_2 = H_1$?
[accept/reject]
$\{$Login request to $LSSNS_j\}$
$\xrightarrow{\hspace{3cm}}$

(public channel)
Establish **session key** $\mathcal{SK}_{S_j,CT}$
$\{$Authentication response$\}$
$\xleftarrow{\hspace{2.5cm}}$
(public channel)

Establish **session key** SK_{CT,S_j}
Randomly generate $\{x_i^*, y_i^*, enc_i^*\}_{i=1...\mathcal{L}-1}$
Set real location at n^{th} place $(1 \leq n \leq \mathcal{L})$
Compute
$Index_{u_i} = E_{X_\Gamma}[Seq(x_{u_i}||y_{u_i}||E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}))]$
$Msg_2 = \{ID_i, (x_1^*, y_1^*, enc_1^*), \ldots, (x_{u_i}, y_{u_i},$
$E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i})), \ldots, (x_\mathcal{L}^*, y_\mathcal{L}^*, enc_\mathcal{L}^*), Index_{u_i}\}$
$H_3 = H(LSSNS_j||Msg_2||RN_{ct})$
$\{E_{\mathcal{SK}_{CT,S_j}}(Msg_2||RN_{ct}||TS_{ct}), H_3, TS_{ct}\}$
$\xrightarrow{\hspace{3cm}}$
(public channel)

Decrypts $D_{\mathcal{SK}_{S_j,CT}}(Msg_2)$
Verify if $|TS_{ct}^* - TS_{ct}| \leq \triangle T$?
Compute and verify hash value
[accept/reject]
Update the user location.
Compute
$Msg_3 = E_{\mathcal{SK}_{S_j,CT}}(ID_i||LSSNS_j||$
$\quad RN_{ct}||\text{'}ok\text{'})$
$\{Msg_3\}$
$\xleftarrow{\hspace{2.5cm}}$
(public channel)

Decrypts $D_{\mathcal{SK}_{CT,S_j}}(Msg_3)$
Verify correctness of RN_{ct}
[accept/reject]
(public channel)

FIGURE 9. User location update phase of the proposed scheme.

update process takes only 0.0721 second. Hence, LSSNS can update the current location of MU_i very quickly.

Depending on the population density, potential users, etc., the LTE technology nowadays requires cellular towers
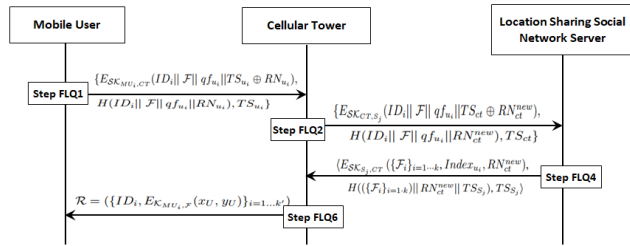
IEEE Access

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

FIGURE 10. The message communication in friends' locations query phase. Please refer to Section V-F.

(or BTSs) to be spaced in the range of 2km to 5km [50]. When the mOSN user MU_i moves to a new cellular tower zone, he/she needs to register to CT. Once the registration is complete, MU_i provides his/her location update to LSSNS and access other location-based services from LSSNS. This mobile user registration process is a one-time task and incurs very small computation cost on a mobile device. As presented in Figure 2 and Table 3, MU_i registration has the computation cost of $5*T_H + 5*T_X + T_{FE}$, which essentially takes only 0.0656 second. This evidences that the user registration process is very efficient.

F. THE FRIENDS' LOCATIONS QUERY PHASE

In this subsection, we describe how MU_i achieves his/her social friends' identity and location, who are willing to share their information. The message communications of Friends' Locations Query Phase is shown in Figure 10.

Step FLQ1:

1) MU_i makes a secure login to CT and mutually establishes a shared session key $\mathcal{SK}_{MU_i,CT}(=\mathcal{SK}_{CT,MU_i})$ (As explained in subsection V-B).
2) MU_i sends $\{E_{\mathcal{SK}_{MU_i,CT}}(ID_i||\mathcal{F}||qf_{u_i}||TS_{u_i}\oplus RN_{u_i}), H(ID_i||\mathcal{F}||qf_{u_i}||RN_{u_i}), TS_{u_i}\}$ to CT.

Note that the message $\mathcal{F}$ is a request to find 'friends'. TS_{u_i}, RN_{u_i}, $E(\cdot)$ and $H(\cdot)$ convey their usual meanings as explained in Table 1.

Step FLQ2:

1) CT receives request from MU_i, and checks if $|TS_{u_i}^* - TS_{u_i}| \overset{?}{\leq} \Delta T$. Here, $TS_{u_i}^*$ is the current timestamp. If the verification holds, go to step 2, else *exit*.
2) CT uses its session key $\mathcal{SK}_{CT,MU_i}$ (shared with MU_i) to decrypt the encrypted user message.
3) CT uses received timestamp TS_{u_i} and parameter $TS_{u_i}\oplus RN_{u_i}$ to retrieve the random number as $RN'_{u_i}=TS_{u_i}\oplus RN_{u_i}\oplus TS_{u_i}$.
4) CT computes hash value $H_3 = H(ID_i||\mathcal{F}||qf_{u_i}||RN'_{u_i})$. If the computed H_3 and the received hash value does not match, then CT rejects the request immediately. Otherwise, go to step 5.
5) CT logs in to the server $LSSNS_j$ and creates a shared session key $\mathcal{SK}_{CT,S_j}$, as explained in subsection V-C.

6) Through a public channel, CT forwards $\{E_{\mathcal{SK}_{CT,S_j}}(ID_i||\mathcal{F}||qf_{u_i}||TS_{ct}\oplus RN_{ct}^{new}), H(ID_i||\mathcal{F}||qf_{u_i}||RN_{ct}^{new}), TS_{ct}\}$ to $LSSNS_j$.

Step FLQ3:

1) $LSSNS_j$ receives the message from CT and decrypts the message using its session key $\mathcal{SK}_{S_j,CT}$ (shared with CT).
2) $LSSNS_j$ checks the communication delay using current timestamp TS_{ct}^* and the received timestamp TS_{ct}. if verification holds, go to step 3, else terminate the session and *exit*.
3) $LSSNS_j$ retrieves RN_{ct}^{new} and computes fresh hash value with the decrypted parameter and compares with the received hash value.
4) If verification holds, go to step *FLQ4*, else terminate the session and *exit*.

Step FLQ4:

1) $LSSNS_j$ finds the set Θ containing a database entry for all friends of MU_i.
2) $LSSNS_j$ finds whether $\delta((x_p, y_p), (x_{u_{i_t}}, y_{u_{i_t}})) \leq min(qf_{u_i}, df_s)_{s\in\Theta}, p = 1, \cdots, k$, and $t = 1, \cdots, k$, where $\delta(\cdot)$ is the distance function and $(x_{u_{i_t}}, y_{u_{i_t}})$ are one real and $k-1$ fake locations of MU_i. Here, database entry of ID_i is excluded.
3) For all friends $\alpha \in \Theta$, $LSSNS_j$ includes record $(\alpha, (p, enc_p^*), Index_\alpha)$ in the result set if the coordinate $(x_{\alpha_{i_t}}, y_{\alpha_{i_t}})$ meets the distance requirement.
4) Corresponding to k coordinate entries of MU_i $(x_{u_i}, y_{u_i})_{i=1\cdots k}$, $LSSNS_j$ prepares k subsets $\{\mathcal{F}_i\}_{i=1\cdots k}$ and adds them to result set.
5) $LSSNS_j$ uses RN_{ct}^{new} (the random number sent by CT), TS_{S_j} (the current timestamp) and encrypts the result set using the shared session key $\mathcal{SK}_{S_j,CT}$.
6) Through public channel, $LSSNS_j$ forwards message $\langle E_{\mathcal{SK}_{S_j,CT}}(\{\mathcal{F}_i\}_{i=1\cdots k}, Index_{u_i}, RN_{ct}^{new}), H((\{\mathcal{F}_i\}_{i=1\cdots k})||RN_{ct}^{new}||TS_{S_j}), TS_{S_j}\rangle$ to CT.

Step FLQ5:

1) CT receives the encrypted result set from $LSSNS_j$.
2) CT decrypts it using own shared session key $\mathcal{SK}_{CT,S_j}$ to obtain $\{\mathcal{F}_i\}_{i=1\cdots k}$, $Index_{u_i}$, and RN_{ct}^{new}.
3) CT verifies transmission delay by comparing the current timestamp $TS_{S_j}^*$ and the received timestamp TS_{S_j}.
4) CT verifies the value of received RN_{ct}^{new} with stored RN_{ct}^{new}.
5) If both verification of step 3 and 4 is successful, then go to step *FLQ6*, else *exit*.

Step FLQ6:

1) CT decrypts $Index_{u_i}$ as $D_{X_\Gamma}(Index_{u_i}) = D_{X_\Gamma}(E_{X_\Gamma}(Seq(x_{u_i}||y_{u_i}||E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i}, y_{u_i}))))$ and retrieves the real sequence number γ. CT uses its master secret key for the decryption.
2) CT discards all records $\{\mathcal{F}_i\}_{i\neq\gamma}$ and accepts only $\mathcal{F}_\gamma$.
3) CT finds every present user U in the dataset $\mathcal{F}_\gamma$.
4) CT decrypts $Index_U$ and finds its real center point location γ_U.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE*Access*

TABLE 2. Notations and their descriptions used in BAN logic.

Symbol	Description
$Q \mid\equiv S$	Q believes that the statement S is true
$Q \lhd S$	Q can see the statement S
$\#(S)$	Formula S is considered as fresh
$Q \mid\sim S$	Q said the statement S once
$Q \Rightarrow S$	Q keeps jurisdiction over the statement S
$\langle S \rangle_T$	Formula S is combined with the formula T
$Q \xleftrightarrow{K} R$	Only Q and R know the value of the key K and it is used for communication between them
$Q \stackrel{S}{\rightleftharpoons} R$	Only Q and R know the secret statement S. Principals trusted by Q & R may know S
SK	Current session key

5) CT retrieves $enc_{\gamma_U} = E_{\mathcal{K}_{MU_i, \mathcal{F}}}(x_U, y_U)$ and prepares answer set $\mathcal{R}$.

6) CT sends the friend set $\mathcal{R} = (\{ID_{U_i}, E_{\mathcal{K}_{MU_i, \mathcal{F}}}(x_U, y_U)\}_{i=1...k'})$ to MU_i.

Remark 3: The existing location sharing schemes for OSN suffer from multiple security drawbacks. The purpose of our research is to design a secure and efficient location sharing scheme for OSN. User location updates and friend's location queries are two essential operations for location sharing services. As mentioned in existing location sharing schemes [28], [29], [34], group key-establishment among a user and its trusted social friends is an intrinsic requirements. In the literature, several group key distribution and key-establishment schemes among social friends have been proposed in distributed online social networks [51]–[54].

Unlike those schemes, our proposed one is not designed for purpose group key distribution and key-establishment among social friends. That said, the key-establishment process among social friends advocated by Y. Jung et al. [51] and L. Guo et al. [54] can be adapted to work with our proposed scheme.

VI. SECURITY ANALYSIS
In this section, we provide the detail security analysis of the proposed scheme. This is done in two ways. First, we present the authentication proof Using Burrows-Abadi-Needham (BAN) logic. Second, we present an informal security analysis to logically explain how and why the proposed scheme resists various security attacks.

A. AUTHENTICATION PROOF USING BAN LOGIC
BAN logic is used to analyze the security of any authentication scheme to verify the secure transmission between two communicating parties of that network [55]. In this section, we use BAN logic to show that the proposed scheme actually achieves the authentication goals. The basic syntax and semantics of BAN logic are explained in Table 2.
The main logical postulates of the BAN logic are defined by a set of laws or rules as listed below [55], [56].

- Law 1 (Message Meaning Law (MML)).
  $\frac{Q\mid\equiv R\xleftrightarrow{K} Q, Q \lhd \langle S \rangle_K}{Q\mid\equiv R\mid\sim S}$.
- Law 2 (Nonce Verification Law (NVL)).
  $\frac{Q\mid\equiv\#(S), Q\mid\equiv R\mid\sim S}{Q\mid\equiv R\mid\equiv S}$.
- Law 3 (Freshness Conjuncatenation Law (FCL)).
  $\frac{Q\mid\equiv\#(S)}{Q\mid\equiv\#(S,T)}$.
- Law 4 (Jurisdiction Law (JL)).
  $\frac{Q\mid\equiv R\Rightarrow S, Q\mid\equiv R\mid\equiv S}{Q\mid\equiv S}$.
- Law 5 (Additional Laws (AL)).
  $\frac{Q\mid\equiv(S,T)}{Q\mid\equiv S}$, $\frac{Q\lhd(S,T)}{Q\lhd S}$, $\frac{Q\mid\equiv R\mid\sim(S,T)}{Q\mid\equiv R\mid\sim S}$.

In order to show that the proposed scheme ensures authentication, two goals, as mentioned in the following, must be achieved.

Goal 1: $MU_i \mid\equiv (MU_i \xleftrightarrow{SK} CT)$.

Goal 2: $CT \mid\equiv (MU_i \xleftrightarrow{SK} CT)$.

In the proposed scheme, there will be two basic types of messages as follows:

Message 1: $MU_i \rightarrow CT$: { $TID_i^*, H(H(ID_i \oplus x_{CT_{u_i}}||x_\Gamma)) \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT}), \mathcal{T}_{u_i}, H_1$}.

Message 2: $CT \rightarrow MU_i$: $\{B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i, \mathcal{T}_{ct}, H_3\}$.

The above generic messages have to be converted to idealized messages. These idealized messages are as follows.

Message 1: $MU_i \rightarrow CT$: $\{TID_i, \mathcal{T}_{u_i}, \langle ID_i, x_{CT_{U_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma, H_1\}$.

Message 2: $CT \rightarrow A$: $\{\mathcal{T}_{ct}, \langle \Omega_{ct} \mathcal{T}_{ct}, ID_i, \rangle x_\Gamma, H_3\}$.

With the following assumptions, the authentication proof of our proposed scheme is presented as follows:
A.1: $MU_i \mid\equiv \#(\mathcal{T}_{ct})$;
A.2: $CT \mid\equiv \#(\mathcal{T}_{u_i})$;
A.3: $MU_i \mid\equiv (MU_i \stackrel{A_{u_i CT}}{\rightleftharpoons} CT)$;
A.4: $CT \mid\equiv (MU_i \stackrel{A_{u_i CT}}{\rightleftharpoons} CT)$;
A.5: $MU_i \mid\equiv CT \Rightarrow (ID_{ct}, \Omega_{ct}, \mathcal{T}_{ct})$;
A.6: $CT \mid\equiv MU_i \Rightarrow (ID_i, \Omega_{u_i}, \mathcal{T}_{u_i})$;
A.7: $MU_i \mid\equiv \mathcal{T}_{u_i}$;
A.8: $MU_i \mid\equiv \Omega_{u_i}$;
A.9: $MU_i \mid\equiv ID_i$;
A.10: $MU_i \mid\equiv ID_{CT}$;
A.11: $CT \mid\equiv \mathcal{T}_{ct}$;
A.12: $CT \mid\equiv \Omega_{ct}$;
A.13: $CT \mid\equiv ID_{CT}$.

Next, we shall show that two goals mentioned earlier can be achieved using the assumptions, idealized messages and Basic BAN logic laws.

From the first message, we may obtain the following.

- S_1: $CT \lhd \{ID_i, \mathcal{T}_{u_i}, \langle ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma, H_1\}$.
- S_2: Using AL, we derive: $CT \lhd \langle ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT})\rangle x_\Gamma$.
- S_3: According to A.4 and MML, we obtain, $CT \mid\equiv MU_i \mid\sim (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.

IEEE Access

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

- S_4: According to A.2 and FCL, we get, $CT \mid\equiv \#(ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_5: According to NVL, we have, $CT \mid\equiv MU_i \mid\equiv (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_6: Using A.6 and JL, we get, $CT \mid\equiv (ID_i, x_{CT_{u_i}}, \Omega_{u_i}, \mathcal{T}_{u_i}, H(ID_{CT}))$.
- S_7: From S_6 and AL, we obtain, $CT \mid\equiv \Omega_{u_i}, CT \mid\equiv \mathcal{T}_{u_i}, CT \mid\equiv ID_i$.
- S_8: According to A.11, A.12, A.13, we get, $CT \mid\equiv ID_{CT}, CT \mid\equiv \mathcal{T}_{ct}$ and $CT \mid\equiv \Omega_{ct}$.
- S_9: Since $SK_{CT,MU_i} = H(ID_i \mid\mid ID_{CT} \mid\mid B_{CT_{u_i}} \mid\mid \mathcal{P}1 \mid\mid \Omega_{ct} \mid\mid \mathcal{T}_{u_i} \mid\mid \mathcal{T}_{ct})$ and the results in Steps S_7 and S_8 give $CT \mid\equiv (MU_i \overset{SK_{CT,MU_i}}{\longleftrightarrow} CT)$. **(Goal 2)**
- S_{10}: Using the message 2 and AL, we obtain, $MU_i \lhd \langle \Omega_{ct}, \mathcal{T}_{ct} \rangle_{x_\Gamma}$.
- S_{11}: According to A.3 and MML, we get, $MU_i \mid\equiv CT \mid\curvearrowright (\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{12}: Using A.1 and FCL, we obtain, $MU_i \mid\equiv \#(\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{13}: Using NVL, we obtain, $MU_i \mid\equiv CT \mid\equiv (\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{14}: A.5 and JL give $MU_i \mid\equiv (\Omega_{ct}, \mathcal{T}_{ct})$.
- S_{15}: According to S_{14} and AL, we have, $MU_i \mid\equiv \Omega_{ct}, MU_i \mid\equiv \mathcal{T}_{ct}$.
- S_{16}: According to A.7-A.10, we obtain, $MU_i \mid\equiv ID_i, MU_i \mid\equiv ID_{CT}, MU_i \mid\equiv \mathcal{T}_{u_i}, MU_i \mid\equiv \Omega_{ct}$.
- S_{17}: The results of Steps S_{15} and S_{16} give $MU_i \mid\equiv (MU_i \overset{SK_{CT,MU_i}}{\longleftrightarrow} CT)$. **(Goal 1)**

Consequently, both the goals are achieved to ensure that mutual authentication between MU_i and CT is established.

B. INFORMAL SECURITY ANALYSIS

In this section, we present an informal analysis of the security of the proposed scheme. This analysis aims to logically show that our scheme can successfully defend against the following known attacks.

1) THE REPLAY ATTACK

In the proposed scheme, two message communications are needed by the login phase and the authentication phase. In the process of login, MU_i sends $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to CT, whereas in authentication phase, CT sends $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i. CT does not accept Msg_1 if $|\mathcal{T}_{u_i}^* - \mathcal{T}_{u_i}| \geq \Delta T$. Additionally, CT computes $H_2 = H(ID_i \mid\mid \mathcal{M}_1 \mid\mid \mathcal{P}_1 \mid\mid \mathcal{T}_{u_i})$ and checks whether $H_2 = H_1$ or not. This computation is crucial to prevent a replay attack. The cellular tower CT rejects any request for log-in if this checking does not succeed. We have explained in *Step LA5* in Section V-B, of the mOSN user login, authentication and key establishment phase how an attacker cannot succeed in replaying the authentication message Msg_2. Moreover, CT also stores parameters $\langle ID_i, \Omega_{u_i}, \mathcal{T}_{u_i} \rangle$ in its repository. In case CT receives another login request message, say $Msg_1^n = \{TID_n^*, \mathcal{M}_1^n, H_1^n, \mathcal{T}_{u_n}\}$, it first checks whether T_{u_n} is valid or not. If it is found to be valid, CT gies on to check whether the extracted $TID_n^* = TID_n \oplus H(ID_{CT} \mid\mid \mathcal{T}_{u_n})$ is the same as the TID_n stored in

its repository for the same ID_n. If they are the same, Msg_1^n is considered being a replay message. Thus, our proposed scheme is capable of resisting a strong replay attack with the help of current timestamp and a random nonce.

2) THE MAN-IN-THE-MIDDLE ATTACK

An adversary $\mathcal{A}$ may attempt to modify login or authentication message through a man-in-the-middle attack. In order to execute this attack, $\mathcal{A}$ set up an independent parallel connection with both MU_i and CT for a specific session. Additionally, to invalidate the login request of an authorized user, the attacker may modify some parameters from the request message. In the proposed scheme, the credentials of both login and authentication message, such as ID_{CT}, RID_Γ, $A_{U_{iCT}}$, etc. are generated with fuzzy extractor, hash function, bitwise XOR and random nonce. This makes adversary $\mathcal{A}$ very difficult to regenerate and modify. As a consequence, the proposed scheme can resist the man-in-the-middle attack.

3) THE STOLEN/LOST MOBILE DEVICE ATTACK

Suppose the mobile device of the user MU_i has been stolen or lost, an adversary can easily find P_i^1 and P_i^2, which are stored in the memory of the device. However, ID_i, PW_i, and biometric η_i are not stored directly in the device. From stored $P_i^1 = H(PW_i \mid\mid \eta_i) \oplus n$ and $P_i^2 = H(ID_i \mid\mid PW_i \mid\mid \eta_i \mid\mid n)$, it is computationally infeasible to identify or predict all these parameters. Furthermore, P_i^1 and P_i^2 are masked with a random number n and the collision-resistant hash function $H(\cdot)$. This makes it a computationally infeasible problem to predict all the credentials in polynomial time. Therefore, the proposed scheme resists this type of attacks.

4) THE OFFLINE PASSWORD GUESSING ATTACK

As describe in Section V-B, a mobile user MU_i needs the identity ID_i and password PW_i for its login. An adversary can obtain the P_i^1 and P_i^2 from the lost or stolen mobile device, but it cannot guess and compute identity ID_i, password PW_i, and biometric η_i at the same time as it is computationally infeasible. Hence, this scheme can prevent the offline password guessing attack.

5) KNOWN KEY SECRECY/FORWARD SECRECY

An adversary may obtain the current session key, but with that compromised session key, it cannot compute previous session keys. As per the proposed scheme, the session key is computed as $SK_{CT,MU_i} = SK_{MU_i,CT} = H(ID_i \mid\mid ID_{CT} \mid\mid B_{CT_{U_i}} \mid\mid \mathcal{P}_1 \mid\mid \Omega_{ct} \mid\mid \mathcal{T}_{u_i} \mid\mid \mathcal{T}_{ct})$ where $A_{U_{iCT}} = B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}}) \mid\mid X_\Gamma)$. With the use of $\Omega_i, T_{u_i}, \Omega_j$, and T_{u_j}, a new login key for each session, $SK_{CT,MU_i} = SK_{MU_i,CT}$ is generated freshly and uniquely. So, the key cannot be used further in future. Moreover, before establishing a session key, both MU_i and CT mutually validated each other. Hence, the proposed scheme confirms that the leakage of temporal information does not break the secrecy of the session key and it provides the session key security.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE Access

TABLE 3. Computation cost of the proposed scheme.

Phase/Entity	Mobile User (MU_i)	Cellular Tower (CT)	Location Server $(LSSNS_j)$	Total execution time (in ms)
Mobile User Login and Authentication	$10*T_H + 10*T_X + T_{FE}$	$-$	$-$	71.58
	$-$	$7*T_H + 8*T_X$	$-$	
Location Server Login and Authentication	$-$	$-$	$7*T_H + 8*T_X + T_{CH}$	45.54
	$-$	$6*T_X + T_{CH}$	$-$	
Distance Threshold Registration	$T_H + 2*T_{sym}$	$2*T_H + 4*T_{sym}$	$2*T_{sym}$	71.10
User Location Update	$T_H + 2*T_{sym}$	$2*T_H + 4*T_{sym}$	$2*T_{sym}$	72.10
Friends' Location Query	$T_H + 2*T_{sym}$	$3*T_H + 4*T_{sym}$	$2*T_H + 2*T_{sym}$	72.6

6) USER ANONYMITY

In this proposed scheme, the anonymity property of any mobile user is maintained. An adversary may eavesdrop a login or authentication message communicated between MU_i and CT, but adversary cannot get the original ID_i from those messages. At the time of login MU_i send $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to cellular tower CT. Instead of its original identity MU_i send its temporary identity TID_i embedded in $TID_i^* = TID_i \oplus H(ID_{CT} || \mathcal{T}_{u_i})$, which is valid for only one session. Furthermore, it is not possible to compute ID_i from $\mathcal{M}_1 = A_{U_{iCT}} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$ and $H_1 = H(ID_i || \mathcal{M}_1 || \Omega_{u_i} || \mathcal{T}_{u_i})$. At the time of authentication, CT transmits back a authentication response message $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i where $\mathcal{M}_2 = B_{CT_{U_i}} \oplus \Omega_{ct} \oplus \mathcal{T}_{ct} \oplus ID_i$ and $H_3 = H(ID_i || \mathcal{P}_1 || \Omega_{ct} || \mathcal{T}_{u_i} || \mathcal{T}_{ct} || SK_{CT,MU_i})$. So, from any intrude message, it is not feasible to figure out the original ID_i by an adversary. Thus, the proposed scheme can preserve the anonymity property of any user.

7) THE PARALLEL SESSION AND REFLECTION ATTACK

In the proposed scheme, an adversary cannot start a new session with CT using any fake identity, obtaining from any eavesdropped messages $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$. As described in section V-B, an adversary cannot obtain the correct identity ID_i, password PW_i or the biometric key η_i of any legal user MU_i with an offline password guessing attack. Hence, from any eavesdropped message, an attacker cannot create a valid login request message Msg_1, so a new session with CT as a legal user not possible. Thus, our proposed scheme can protect the parallel session and reflection attacks.

8) SESSION KEY SECURITY

For establishing a new session, a mutually computed session key $SK_{MU_i,CT} (= SK_{CT,MU_i})$ is shared between MU_i and CT. The session key is computed as follows:

$$SK_{CT,MU_i} = H(ID_i || ID_{CT} || B_{CT_{U_i}} || \mathcal{P}_1 || \Omega_{ct} || \mathcal{T}_{u_i} || \mathcal{T}_{ct})$$
$$= H(ID_i || ID_{CT} || A_{U_{iCT}} || \mathcal{P}_1 || \Omega_{ct} || \mathcal{T}_{u_i} || \mathcal{T}_{ct})$$
$$= H(ID_i || ID_{CT} || A_{U_{iCT}} || \Omega_{u_i} || \Omega_{ct} || \mathcal{T}_{u_i} || \mathcal{T}_{ct})$$
$$= H(ID_i || ID_{CT} || A_{U_{iCT}} || \Omega_{u_i} || \mathcal{P}_2 || \mathcal{T}_{u_i} || \mathcal{T}_{ct})$$
$$= SK_{MU_i,CT}$$

Both MU_i and CT authenticate each other to compute the mutually shared session key. Moreover, an adversary needs

the credentials ID_i, ID_{CT}, $B_{CT_{U_i}} = (A_{U_{iCT}})$ for computing session keys. Therefore, the session keys are fully secured in our proposed scheme.

9) THE EPHEMERAL SECRET LEAKAGE ATTACK

An adversary may obtain the temporary (ephemeral) secrets (e.g., random variable) of any session from a compromised mobile device if those are not deleted properly. In this kind of attacks, with the mentioned information, an attacker can initiate an ephemeral secret leakage attack. As per our proposed scheme, our session key is generated as follows:

$$SK_{MU_i,CT} = H(ID_i || ID_{CT} || A_{U_{iCT}} || \Omega_{u_i} || \mathcal{P}_2 || \mathcal{T}_{u_i} || \mathcal{T}_{ct})$$

where $A_{U_{iCT}} = V'_{CT_{U_i}} \oplus (H(ID_i || H(PW_i || n' || \eta_i)))$. Ω_{u_i} is a 128-bit random number there. With this single random number, an attacker cannot regenerate the session key $SK_{MU_i,CT}$, as it requires some other credentials, such as ID_i, ID_{CT}, PW_i etc. Thus, our scheme can defend the ephemeral secret leakage attack.

10) THE USER IMPERSONATION ATTACK

In the user impersonation attack, an adversary pretends itself as an authorized user to the cellular tower. So, for login, an adversary needs the credentials value of ID_i, PW_i, $\mathcal{B}'_i$. As it is already discussed that in our proposed scheme, these credentials are not sent directly through the public channel or saved in the device memory, or it is computationally infeasible to obtain them from the easily available information. If an adversary wants to send a login message $Msg_1 = \{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ to CT, it needs to compute $TID_i^* (= TID_i \oplus H(ID_{CT} || \mathcal{T}_{u_i})$ and $\mathcal{M}_1 = A_{U_{iCT}} \oplus \Omega_{u_i} \oplus \mathcal{T}_{u_i} \oplus H(ID_{CT})$. After receiving the login request, with the help of timestamp value $\mathcal{T}_{u_i}$ and the random variable $A_{U_{iCT}}$, CT can determine if the received message Msg_1 is original or replayed. Therefore, with invalid ID_i, PW_i, and $\mathcal{B}'_i$, it is not possible to generate or modify Msg_1. Thus, our proposed scheme can defend the user impersonation attack.

11) THE SERVER IMPERSONATION ATTACK

In the server impersonation attack, an attacker may pretend itself as a genuine server. In our proposed scheme, after receiving valid login request message, a cellular tower CT replies back with an authorization message $Msg_2 = \{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ to MU_i. For calculating $\mathcal{M}_2 (= B_{CT_{U_i}} \oplus \Omega_{ct} \oplus$

IEEE Access

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

```
(*   channels *)
free pch: channel. (* public channel *)
free sch: channel [private]. (* private channel *)
(*   shared keys  *)
free SKmuct:bitstring [private].(* the session key of user *)
free SKctmu:bitstring [private]. (* the session key of cellular tower *)
(*  Cellular tower secret key   *)
free Xtau:bitstring [private].
free XctU:bitstring [private].
(*   constants  *)
free IDCT:bitstring [private].
free IDmu:bitstring [private].
free PW:bitstring [private].
const Bi:bitstring [private].
const dfu:bitstring [private].
const dsu:bitstring [private].
(*  functions and equations  *)
fun H(bitstring):bitstring. (* hash function *)
fun Generation(bitstring):bitstring. (* Fuzzy extractor function *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation *)
fun con(bitstring,bitstring):bitstring. (* string concatenation *)
equation forall x:bitstring,y:bitstring; xor(xor(x,y),y) = x.
(*  aims for verification  *)
query attacker(SKmuct).
query attacker(SKctmu).
query id:bitstring; inj-event(UserAuth(id)) ==> inj-event(UserStart(id)).
(*  event  *)
event UserStart(bitstring). (* User starts authentication *)
event UserAuth(bitstring). (* User is authenticated *)
(*—event CTReg(bitstring). (* Cellular Tower starts Registration *)
```

FIGURE 11. Code for channel declarations, keys, constants, functions, equations, queries and events.

$\mathcal{T}_{ct} \oplus ID_i$) and hash value $H_3(= H(ID_i \,||\mathcal{P}_1\,||\Omega_{ct}\,||\mathcal{T}_{u_i}\,||\mathcal{T}_{ct}\,||SK_{CT,MU_i}))$, an attacker needs the secret key, X_Γ of the cellular tower and the random number $x_{CT_{U_i}}$ as $B_{CT_{U_i}} = H(H(ID_i \oplus x_{CT_{U_i}})\,||X_\Gamma)$. Hence, our proposed scheme can resist the server impersonation attack.

12) THE PRIVILEGED-INSIDER ATTACK

This kind of attacks is launched by an internal user who may be authorized to use the system that is attacked. Suppose that an adversary, who is an internal user also, obtains the registration credentials ID_i, $(MPWB_i \oplus \lambda)$ from the mobile registration request Msg_1. However, as discussed in section V-B, it is not feasible to compute the PW_i and the biometric key $\eta_i =$ even if the adversary has that lost or stolen mobile device. Without the knowledge of λ, it also not possible to calculate $MPWB_i$ from $(MPWB_i \oplus \lambda)$. So, our scheme can resist this type of attacks.

VII. FORMAL SECURITY VERIFICATION USING PROVERIF

In this section, we present the formal security verification of the proposed scheme using based ProVerif simulation tool [57]. This tool is based on applied pi calculus and can be used to verify whether an attacker can attack the session key [48]. We have modelled the proposed scheme in ProVerif and corresponding the source codes have been presented in Figure 11, Figure 12, Figure 13, and Figure 14.

In Figure 11, the code for channel declarations is presented along with the definition of constants, free variables,functions, equations, queries and events, which are

```
(*———user starts———*)
let MUser=
new n:bitstring;
new lamda:bitstring;
let meu = Generation(Bi) in
let MPWB = H(con(IDmu,H(con(PW,con(meu,n))))) in
out(sch,(IDmu,xor(MPWB,lamda)));
in(sch,(TIDMU:bitstring,VctU:bitstring,cRID:bitstring));
let P1 = xor(H(con(PW,meu)),n) in
let P2 = H(con(IDmu,con(PW,con(meu,n)))) in
let VctU1 = xor(VctU,lamda) in
let RIDi = xor(TIDMU,H(con(IDmu,VctU1))) in
let RID1 = xor(cRID,H(con(meu,n))) in
!
(
event UserStart(IDmu);
let n1 = xor(P1,H(con(PW,meu))) in
let P21 = H(con(IDmu,con(PW,con(meu,n1)))) in
if P2 = P21 then
new Omega:bitstring;
new TUi:bitstring; (*——— Current Timestamp ———*)
let MPWB1 = H(con(IDmu,H(con(PW,con(meu,n1))))) in
let mAuct = xor(VctU1,MPWB1) in
let M1 = xor(mAuct,xor(Omega,xor(TUi,H(IDCT)))) in
let TID1 = xor(RIDi,H(con(IDmu,VctU1))) in
let TIDi = xor(TID1,H(con(IDCT,TUi))) in
let H1 = H(con(IDmu,con(M1,con(Omega,TUi)))) in
out(pch,(TIDi,M1,H1,TUi));
in(pch,(M2:bitstring,xH3:bitstring,xTct:bitstring)); (*——- received after authentication ——-*)
let P22 = xor(M2,xor(xTct,xor(IDmu,mAuct))) in
let SKmuct = H(con(IDmu,con(IDCT,con(mAuct,con(Omega,con(P22,con(TUi,xTct))))))) in
let H4 = H(con(IDmu,con(Omega,con(P22,con(TUi,con(xTct,SKmuct)))))) in
if H4 = xH3 then
0
).
```

FIGURE 12. Code in ProVerif for the process of MU_i, the ith mobile user.

```
let CTReg =
in(sch,(xIDmu:bitstring,uMPWB:bitstring));
new TIDMU:bitstring;
let RID = H(con(IDCT,Xtau)) in
let Auct = H(con(H(xor(xIDmu,XctU)),Xtau)) in
let VctU = xor(Auct,uMPWB) in
out(sch,(TIDMU,VctU,RID)).
(*————————————*) heightlet CTAuth =
in(pch,(mTIDi:bitstring,mM1:bitstring,mH1:bitstring,mTUi:bitstring));
new TSi:bitstring; (*—Received Timestamp—*)
let TID2 = xor(mTIDi,H(con(IDCT,mTUi))) in
let Bctu = H(con(H(xor(IDmu,XctU)),Xtau)) in
let P1 = xor(mM1,xor(mTUi,xor(H(IDCT),Bctu))) in
let H2 = H(con(IDmu,con(mM1,con(P1,mTUi)))) in
if H2 = mH1 then
event UserAuth(IDmu);
new OmegaCT:bitstring;
new Tct:bitstring;
let M2 = xor(Bctu,xor(OmegaCT,xor(Tct,IDmu))) in
let SKctmu =H(con(IDmu,con(IDCT,con(Bctu,con(P1,con(OmegaCT,con(mTUi,Tct))))))) in
let H3 = H(con(IDmu,con(P1,con(OmegaCT,con(mTUi,con(Tct,SKctmu)))))) in
out(pch,(M2,H3,Tct)).
let CT = CTReg — CTAuth.
process !MUser — !CT
```

FIGURE 13. Code in ProVerif for the process of CT.

needed to model the proposed scheme. Figure 12 depicts the ProVerif code for mobile user MU registration, login, authentication and key-establishment process with CT. Cellular tower registration process (CTReg) and authentication process (CTAuth) have been presented as a parallel composition in Figure 13.

Finally, we execute the codes given in the previous three Figures in the latest version (1.93) of ProVerif simulation tool. The results of session key secrecy (from the user as well as cellular tower) and authentication are presented in Figure 14. The following observations can be drawn from the results.

- RESULT inj-event (UserAuth(id)) ==> inj-event (UserStart(id)) is true.
- RESULT not attacker(SKmuct[]) is true.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

TABLE 4. Communication cost of the proposed scheme.

Phase	Entity	Communicated message	Size (bits)
ULA	MU_i side	$\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$	512
	CT side	$\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$	352
SLA	$LSSNS_j$ side	$\{PID_{S_j}, T_{s_j}(C), \mathcal{R}_1, M_{S_j}, T_{S_j}\}$	640
	CT side	$\{\mathcal{R}_4, M_{CT}, T_{CT}\}$	352
DTR	MU_i side	$\langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|\mathcal{D}_{f_{u_i}}\|RN_{u_i}\|TS_{u_i}\|\mathcal{R}_{flag}=1), H(ID_i\|RN_{u_i}\|TS_{u_i}), TS_{u_i}\rangle$	320
	CT to $LSSNS_j$	$\langle E_{\mathcal{SK}_{CT,S_j}}(ID_i\|\mathcal{D}_{f_{u_i}}\|RN_{u_i}\|RN_{ct}\|TS_{ct}\|\mathcal{R}_{flag}=1), H(ID_i\|RN_{ct}\|TS_{ct}), TS_{ct}\rangle$	320
	CT to MU_i	$E_{\mathcal{SK}_{CT,MU_i}}(ID_i\|RN_{u_i}\|`ok')$	128
	$LSSNS_j$ side	$\langle E_{\mathcal{SK}_{CT,S_j}}(ID_i\|RN_{u_i}\|RN_{ct}\|`ok')\rangle$	128
ULU	MU_i side	$\langle E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|x_{u_i}\|y_{u_i}\|E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_{u_i},y_{u_i})\|RN_{u_i}\|TS_{u_i}), H_1\rangle$	288
	CT to $LSSNS_j$	$\{ID_{S_j}, E_{\mathcal{SK}_{CT,S_j}}(Msg_2\|RN_{ct}\|TS_{ct}), H(LSSNS_j\|Msg_2\|RN_{ct}), TS_{ct}\}$	480
	CT to MU_i	$E_{\mathcal{SK}_{CT,MU_i}}(`ok')$	128
	$LSSNS_j$ side	$E_{\mathcal{SK}_{S_j,CT}}(ID_i\|LSSNS\|RN_{ct}\|`ok')$	128
FLQ	MU_i side	$\{E_{\mathcal{SK}_{MU_i,CT}}(ID_i\|\mathcal{F}\|qf_{u_i}\|TS_{u_i}\oplus RN_{u_i}), H(ID_i\|\mathcal{F}\|qf_{u_i}\|RN_{u_i}), TS_{u_i}\}$	320
	CT to $LSSNS_j$	$\{E_{\mathcal{SK}_{CT,S_j}}(ID_i\|\mathcal{F}\|qf_{u_i}\|TS_{ct}\oplus RN_{ct}^{new}), H(ID_i\|\mathcal{F}\|qf_{u_i}\|RN_{ct}^{new}), TS_{ct}\}$	320
	CT to MU_i	$(\{ID_{U_i}, E_{\mathcal{K}_{MU_i,\mathcal{F}}}(x_U, y_U)\}_{i=1...k'})$	288
	$LSSNS_j$ side	$\langle E_{\mathcal{SK}_{S_j,CT}}(\{\mathcal{F}_i\}_{i=1...k}, Index_{u_i}, RN_{ct}^{new}), H((\{\mathcal{F}_i\}_{i=1\cdot k})\|RN_{ct}^{new}\|TS_{S_j}), TS_{S_j}\rangle$	320

ULA: Mobile user login and authentication phase; **SLA:** Location server login and authentication phase; **DTR:** Distance threshold registration phase; **ULU:** User location update phase; **FLQ:** Friends' Location Query;

File "./tmpfiles/19349353/inpProt.pv", line 79, characters 5-10:
Warning: identifier SKmuct rebound.
File "./tmpfiles/19349353/inpProt.pv", line 112, characters 5-10:
Warning: identifier SKctmu rebound.
Completing equations...
Completing equations...
– Query not attacker(SKmuct[])
Completing...
Starting query not attacker(SKmuct[])
RESULT not attacker(SKmuct[]) is true.
– Query not attacker(SKctmu[])
Completing...
Starting query not attacker(SKctmu[])
RESULT not attacker(SKctmu[]) is true.
– Query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
Completing...
200 rules inserted. The rule base contains 200 rules. 10 rules in the queue.
Starting query inj-event(UserAuth(id)) ==> inj-event(UserStart(id))
RESULT inj-event(UserAuth(id)) ==> inj-event(UserStart(id)) is true.

FIGURE 14. Results of the ProVerif simulation and their analysis.

- RESULT not attacker(SKctmu[]) is true.

From the result set mentioned above, we conclude that the proposed scheme passes the required security verification.

VIII. PERFORMANCE ANALYSIS

In this section, we present the computation and communication cost of our proposed scheme. It is to be noted that the proposed scheme avoids cryptographic operations such as bilinear pairing, elliptic curve point multiplication operation etc., as they incur high computation overhead.

A. COMPUTATION COST ANALYSIS

Table 5 shows various cryptographic operations, corresponding notations and their execution time on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, as performed in [39], [63]. Due to the fuzzy extractor $Rep(\cdot)$ function for

TABLE 5. Various notations used and their time complexity.

Symbol	Description	Execution time (in milliseconds)
T_H	One-way hash function	0.50
T_{sym}	symmetric key encryption/decryption	8.70
T_M	Elliptic curve point multiplication	63.08
T_{CH}	Chebyshev map operation	21.02
T_{FE}	Fuzzy extractor operation	$\approx T_M$

extracting the biometric key α_i, we require $T_{FE} \approx T_M$ [64]. Symmetric encryption/decryption has been given for a AES-128 symmetric cryptosystem. The mobile user registration and *LSSNS* Registration mechanism is a one-time process. As a result, we have not considered the computation cost of the registration phases. In Table 3 we have tabulated the computational overhead for the main three entities of our scheme MU_i, CT and $LSSNS_j$. For MU_i, during login phase overhead is $10*T_H + 10*T_X + T_{FE}$. Since bitwise XOR operation, T_X time is negligible, the overhead will be $10*T_H + T_{FE}$. For the authentication, required overhead of CT will be $7*T_H + 8*T_X \approx 7*T_H$. Hence, overall computation cost of mobile user login and authentication phase is $17*T_H + T_{FE} = 17*0.5 + 1*63.08 = 71.58$ms. Following the same procedure, we calculate the computation cost and the exact execution time of all other remaining phases of the proposed scheme and tabulate them in Table 3.

B. COMMUNICATION COST ANALYSIS

In order to calculate the overall communication overhead of our proposed scheme, we have assumed standard bit sizes

IEEE Access

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

TABLE 6. Security and functionality comparison.

Security attribute /Scheme	C. C. Lee *et al.* [58]	X. Li *et al.* [59]	Tsai-Lo [60]	Irshad *et al.* [61]	H. Wang *et al.* [62]	Our
Stolen smart card attack	✓	✓	X	X	✓	✓
Supports three-factor authentication	X	X	X	X	✓	✓
Off-line password guessing attack	✓	✓	X	✓	✓	✓
On-line password guessing attack	✓	✓	✓	✓	✓	✓
Strong replay attack	✓	✓	✓	✓	✓	✓
Privileged insider attack	✓	✓	X	✓	✓	✓
User impersonation attack	✓	✓	✓	✓	✓	✓
Server impersonation attack	X	✓	✓	✓	✓	✓
Denial of service attack	✓	✓	✓	✓	✓	✓
Known session key secrecy	✓	✓	✓	✓	✓	✓
User anonymity provision	✓	✓	✓	✓	✓	✓
Forward secrecy	✓	✓	✓	✓	✓	✓
Session key security	✓	✓	X	✓	✓	✓
Session key recovery attack	✓	✓	✓	X	✓	✓
Login phase efficiency	X	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓
Supports Location-Sharing	X	X	X	X	X	✓
Supports Friends' Locations Query	X	X	X	X	X	✓
Formal security analysis	X	✓	X	X	✓	✓
Simulation using AVISPA/ProVerif	X	X	X	X	X	✓

TABLE 7. Storage analysis of the proposed scheme.

Entity	Parameters	Total size
MU_i	$\mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}, RID_i,$ $RID'_\Gamma, SK_{MU_i,CT}$	1088 bit
CT	$X_\Gamma, ID_{CT}, ID_i, TID_i, x_{CT_{U_i}},$ $ID_{S_j}, SN_j, r, SK_{CT,MU_i}, SK_{CT,S_j})$	3296 bit
$LSSNS_j$	$E_1, T_{X_\Gamma}(K_j), E_2, f_{S_j}, SK_{S_j,CT}$	800 bit

TABLE 8. Comparison of communication costs.

Scheme	Communication rounds	No. of bits
C. C. Lee *et al.* [58]	3	1088
X. Li *et al.* [59]	5	2592
Tsai-Lo [60]	5	2560
Irshad *et al.* [61]	5	3072
H. Wang *et al.* [62]	5	3200
Our	4	1856

of various parameters and cryptographic function outputs. As an example, the bit size of used identity, random numbers and timestamp are 160, 128 and 32 bits respectively. The size of output of hash function $H(\cdot)$ is 160 bits, (if we use SHA-1 hash function [65]) and output of symmetric encryption/decryption (for example, Advanced Encryption Standard or AES-128 [66]) is 128 bits and the prime number is 160 bits. For mobile user login and authentication in our proposed scheme, two message communications are required. In step ULA2 of Section V-B, CT receives the login request message from mobile user MU_i. In step ULA4, CT sends one authentication response message to the MU_i. The communication cost for transmission of the MU_i login message $\{TID_i^*, \mathcal{M}_1, H_1, \mathcal{T}_{u_i}\}$ requires $(160 + 160 + 160 + 32) = 512$ bits and authentication response message $\{\mathcal{M}_2, H_3, \mathcal{T}_{ct}\}$ requires $(160 + 160 + 32) = 352$ bits. In the same fashion, we calculate the communication cost of messages communicated in various other phases of the proposed scheme. Table 4 shows the detailed communication cost for different phases.

C. STORAGE OVERHEAD ANALYSIS

We have three different entities in our scheme - mobile device (MU_i), cellular tower(CT) and location sharing social network server($LSSNS_j$). We have calculated the storage requirement for each of them separately. The lengths of some important parameters that are needed to calculate the storage space are as follows:

Device identity or serial number:: 160 bit
Output of a secured one way hash function $H(\cdot)$:: 160 bit
Session key:: 160 bit
One random number, r:: 128 bit
Master secret key, X_Γ:: 1024-bit
Secret key, $x_{CT_{U_i}}$:: 1024-bit
Fuzzy Extractor, μ_i:: 128 bit

According to our proposed scheme, a mobile device MU_i mandatorily needs to store $\mu_i, P_i^1, P_i^2, V'_{CT_{U_i}}, RID_i, RID'_\Gamma, SK_{MU_i,CT}$. Hence, the required storage space of MU_i is $= 128 + 160 + 160 + 160 + 160 + 160 + 160 = 1088$ bit.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE*Access*

TABLE 9. Comparison of computational costs among related schemes.

Phase	Entity	C. C. Lee *et al.* [58]	X. Li *et al.* [59]	Tsai-Lo [60]	Irshad *et al.* [61]	H. Wang *et al.* [62]	Our
Mobile user and server login & authentication phase	Mobile user	$5T_H + 3T_{CH}$ ≈ 65.56 ms	$7T_H + 3T_{CH}$ ≈ 66.56 ms	$8T_H + 2T_{CH}$ ≈ 46.04 ms	$11T_H + 3T_{CH}$ ≈ 68.56 ms	$11T_H + 2T_{CH}$ ≈ 47.54 ms	$10T_H + T_{FE}$ ≈ 68.08 ms
	Server	$6T_H + 3T_{CH}$ ≈ 66.06 ms	$4T_H + 2T_{CH}$ ≈ 44.04 ms	$5T_H + 2T_{CH}$ ≈ 44.54 ms	$7T_H + 2T_{CH}$ ≈ 45.54 ms	$8T_H + 2T_{CH}$ ≈ 46.04 ms	$7T_H + T_{CH}$ ≈ 24.52 ms
	RC or CT	–	$8T_H + T_{CH}$ ≈ 25.02 ms	$10T_H$ ≈ 5 ms	$11T_H + T_{CH}$ ≈ 26.52 ms	$8T_H + 3T_{CH}$ $+2T_{Sym}$ ≈ 84.46 ms	$14T_H + T_{CH}$ ≈ 24.52 ms
Total computation cost		≈ 131.62 ms	≈ 135.62 ms	≈ 95.58 ms	≈ 140.62 ms	≈ 178.04 ms	≈ 117.122 ms

A cellular tower, CT, needs minimum $\{X_\Gamma + ID_{CT} + ID_i + TID_i + x_{CT_{U_i}} + ID_{S_j} + SN_j + r + SK_{CT,MU_i} + SK_{CT,S_j}\} = 1024 + 160 + 160 + 160 + 1024 + 160 + 160 + 128 + 160 + 160\} = 3296$ bit storage space to complete its processing. $LSSNS_j$ requires $\{E_1 + T_{X_\Gamma}(K_j) + E_2 + f_{S_j} + SK_{S_j,CT}\} = 160 + 160 + 160 + 160 + 160 = 800$ bit. Table 7 shows the storage analysis of our proposed scheme.

IX. PERFORMANCE AND COMPARATIVE STUDY

In this section, we present a comparative study of our proposed scheme with some recent chaotic-map based user authentication schemes under multi-server environment, such as schemes proposed by C. C. Lee *et al.* [58], X. Li *et al.* [59], Tsai-Lo [60], Irshad *et al.* [61] and H. Wang *et al.* [62]. The comparative study includes detail analysis and comparison in terms of security and functionality features, computation overheads and communication overheads.

In Table 6, we have tabulated an overall security and functionality features comparison among our proposed scheme and other related authentication and key-establishment schemes. It is seen that a large number of the recent schemes do not support three-factor authentication, as they do not include user biometrics [43]. The tabulation result reveals that the existing schemes suffer from various security attacks like stolen smart card attack [60], [61], server impersonation attack [58], session key recovery attack [61] and login phase inefficiency [58]. Moreover, it is observed that these chaotic-map based authentication schemes can not support proper location-sharing and friends' locations query feature. It is clear from Table 6 that the proposed scheme overcomes such security and functionality weaknesses of the existing schemes.

In Table 9, we tabulate and compare the computation overheads of the proposed scheme with the relevant schemes [58]–[61], [62]. The mobile user registration phase and the location sharing social network server registration phase are an one-time process only. Hence, for calculation as well as comparison of communication cost, we consider only user and server login, authentication and key-establishment phases for the proposed and related schemes. Table 5 shows

various cryptographic operations, corresponding notations and their execution time on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, as performed in [39], [63]. For all the given schemes, we separately tabulated computation for the user, server and the registration center or the cellular tower. Also, in Table 9, we mention and compare total computation cost for each relevant scheme.

It is observed that total computation cost of our proposed scheme is ≈ 117.122 ms only, whereas computation cost of C. C. Lee *et al.*'s scheme [58] is ≈ 131.62 ms, X. Li *et al.*'s scheme [59] is ≈ 135.62 ms, Tsai-Lo's scheme [60] is ≈ 95.58 ms, Irshad *et al.*'s scheme [61] is ≈ 140.62 ms and H. Wang *et al.*'s scheme [62] is ≈ 178.04 ms. It is to be noted that, except Tsai-Lo's scheme, we have the lowest computation cost. The reason behind such low computation cost of our proposed schemes is that, we use only two chaotic map operations for authentication and key-establishment purpose, which is the minimum among other related existing schemes.

Table 8 shows and compares message communication rounds and communication cost (in bits) of the proposed scheme with related schemes [58]–[61], [62]. Since the user and server registration phasesare executed only once, we consider only user and server login, authentication & key-establishment phases for calculation of communication cost for the proposed scheme and other schemes. In our proposed scheme, mOSN user and location server login and authentication phase needs 864 bits and 992 bits of message communication respectively, with a total communication cost of 1856 bits. From Table 8, it is clear that, compared to all related scheme, except C. C. Lee *et al.*'s scheme [58], the proposed scheme has the minimum communication cost. Unfortunately, as shown in Table 6, C. C. Lee *et al.*'s scheme is vulnerable to some serious security attacks. Overall, the proposed scheme is both efficient and provides much greater security and functionality features for the smart devices as compared to all existing compared schemes.

X. CONCLUSION

This paper presents an efficient location sharing scheme for mOSNs and shows the ability to resist various active and passive security attacks that are present in the existing schemes.

IEEE Access

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

The proposed scheme integrates LBS and SNS into a set of single entity servers, thereby reducing their internal communication overhead. Our location sharing scheme for mOSNs shows both efficiency and flexibility in location update, sharing, and query of social friends and social strangers. Formal security verification, authentication proof and simulation results prove the security strength of the proposed scheme.

REFERENCES

[1] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan, "A survey on mobile social networks: Applications, platforms, system architectures, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1557–1581, 3rd Quart., 2015.

[2] G. De Francisci Morales, A. Gionis, and C. Lucchese, "From chatter to headlines: Harnessing the real-time Web for personalized news recommendation," in *Proc. 5th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2012, pp. 153–162.

[3] N. Jabeur, S. Zeadally, and B. Sayed, "Mobile social networking applications," *Commun. ACM*, vol. 56, no. 3, pp. 71–79, 2013.

[4] B. W. Laura Garton and C. Haythornthwaite, "Studying online social networks," *J. Comput.-Med. Commun.*, vol. 3, no. 1, 1997, Art. no. JCMC313.

[5] S. Yazji, P. Scheuermann, R. P. Dick, G. Trajcevski, and R. Jin, "Efficient location aware intrusion detection to protect mobile devices," *Pers. Ubiquitous Comput.*, vol. 18, no. 1, pp. 143–162, Jan. 2014.

[6] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for Internet of Things," *Future Gener. Comput. Syst.*, vol. 89, pp. 110–125, Dec. 2018.

[7] S. Kumari, M. K. Khan, and M. Atiquzzaman, "User authentication schemes for wireless sensor networks: A review," *Ad Hoc Netw.*, vol. 27, pp. 159–194, Apr. 2015.

[8] Y. Sun, M. Chen, L. Hu, Y. Qian, and M. M. Hassan, "ASA: Against statistical attacks for privacy-aware users in location based service," *Future Gener. Comput. Syst.*, vol. 70, pp. 48–58, May 2017.

[9] J. Lin, J. I. Hong, D. P. Siewiorek, and N. Sadeh, "Rethinking location sharing: Exploring the implications of social-driven vs. purpose-driven location sharing," in *Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, Copenhagen, Denmark, 2010, pp. 85–94.

[10] 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 *Proc. 15th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, Philadelphia, PA, USA, 2014, pp. 43–52.

[11] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 729–737, doi: 10.1109/INFCOM.2012.6195818.

[12] H. Li, H. Zhu, S. Du, X. Liang, and X. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 646–660, Jul. 2018.

[13] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Philadelphia, PA, USA, 2012, pp. 628–637.

[14] P. S. Modi. (Aug. 31, 2019). *How to Fake Your Location on Facebook, WhatsApp and Snapchat [Guide]*. Accessed: Sep. 2020. [Online]. Available: https://www.mobigyaan.com/how-to-fake-your-location-on-facebook-whatsapp-and-snapchat-guide

[15] Z. Qian, C. Chen, I. You, and S. Lu, "ACSP: A novel security protocol against counting attack for UHF RFID systems," *Comput. Math. With Appl.*, vol. 63, no. 2, pp. 492–500, Jan. 2012.

[16] S. Rass, R. Wigoutschnigg, and P. Schartner, "Doubly-anonymous crowds: Using secret-sharing to achieve sender-and receiver-anonymity," *JoWUA*, vol. 2, no. 4, pp. 27–41, 2011.

[17] Y. Lei, A. Quintero, and S. Pierre, "Mobile services access and payment through reusable tickets," *Comput. Commun.*, vol. 32, no. 4, pp. 602–610, Mar. 2009.

[18] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 05, pp. 557–570, Oct. 2002.

[19] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, 2003, pp. 31–42.

[20] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proc. 21st Int. Conf. Data Eng. Workshops (ICDEW)*, 2005, p. 1248.

[21] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Proc. Int. Symp. Spatial Temporal Databases*. Berlin, Germany: Springer, 2007, pp. 239–257.

[22] Y. Ouyang, Z. Le, Y. Xu, N. Triandopoulos, S. Zhang, J. Ford, and F. Makedon, "Providing anonymity in wireless sensor networks," in *Proc. IEEE Int. Conf. Pervas. Services*, Jul. 2007, pp. 145–148.

[23] M. Rahman, M. Mambo, A. Inomata, and E. Okamoto, "An anonymous on-demand position-based routing in mobile ad hoc networks," in *Proc. Int. Symp. Appl. Internet (SAINT)*, 2006, pp. 300–306.

[24] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervas. Comput.*, vol. 2, no. 1, pp. 46–55, Jan. 2003.

[25] Z. Chen, X. Hu, X. Ju, and K. G. Shin, "LISA: Location information ScrAmbler for privacy protection on smartphones," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2013, pp. 296–304.

[26] F. Rahman, M. E. Hoque, F. A. Kawsar, and S. I. Ahamed, "Preserve your privacy with PCO: A privacy sensitive architecture for context obfuscation for pervasive E-Community based applications," in *Proc. IEEE 2nd Int. Conf. Social Comput.*, Aug. 2010, pp. 41–48.

[27] L. P. Cox, A. Dalton, and V. Marupadi, "SmokeScreen: Flexible privacy controls for presence-sharing," in *Proc. 5th Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, 2007, pp. 233–245.

[28] W. Wei, F. Xu, and Q. Li, "MobiShare: Flexible privacy-preserving location sharing in mobile online social networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 2616–2620.

[29] J. Li, J. Li, X. Chen, Z. Liu, and C. Jia, "$MobiShare$+: Security improved system for location sharing in mobile online social networks," *J. Internet Serv. Inf. Secur.*, vol. 4, no. 1, pp. 25–36, 2014.

[30] N. Shen, J. Yang, K. Yuan, C. Fu, and C. Jia, "An efficient and privacy-preserving location sharing mechanism," *Comput. Standards Interface*, vol. 44, pp. 102–109, Feb. 2016.

[31] 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 Syst. J.*, vol. 11, no. 2, pp. 439–448, Jun. 2017.

[32] Z. Liu, D. Luo, J. Li, X. Chen, and C. Jia, "N-mobishare: New privacy-preserving location-sharing system for mobile online social networks," *Int. J. Comput. Math.*, vol. 93, no. 2, pp. 384–400, Feb. 2016.

[33] Z. Liu, J. Li, X. Chen, J. Li, and C. Jia, "New privacy-preserving location sharing system for mobile online social networks," in *Proc. 8th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, Oct. 2013, pp. 214–218.

[34] X. Xiao, C. Chen, A. K. Sangaiah, G. Hu, R. Ye, and Y. Jiang, "Cen-LocShare: A centralized privacy-preserving location-sharing system for mobile online social networks," *Future Gener. Comput. Syst.*, vol. 86, pp. 863–872, Sep. 2018.

[35] M. Zuckerberg. (Mar. 7, 2019). *A Privacy-Focused Vision for Social Networking*. [Online]. Available: https://www.facebook.com/notes/mark-zuckerberg/a-privacy-focused-vision-for-social-networking/10156700570096634

[36] S. Guha, K. Tang, and P. Francis, "NOYB: Privacy in online social networks," in *Proc. 1st workshop Online Social Netw. (WOSP)*, Seattle, WA, USA, 2008, pp. 49–54.

[37] M. Bhattacharya, S. Roy, S. Banerjee, and S. Chattopadhyay, "Cryptanalysis of a Centralized Location-Sharing Scheme for Mobile Online Social Networks," in *Proc. Springer Adv. Comput. Syst. Secur.*, Kolkata, India, 2020, pp. 1–14.

[38] W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed. Englewood Cliffs:, NJ, USA: Prentice-Hall, 2004.

[39] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*. Cham, Switzerland: Springer, 2011.

[40] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on chebyshev polynomials," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005.

[41] P. Sarkar, "A simple and generic construction of authenticated encryption with associated data," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 4, pp. 1–16, Dec. 2010.

[42] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons Fractals*, vol. 37, no. 3, pp. 669–674, Aug. 2008.

[43] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 824–839, Sep. 2018.

M. Bhattacharya *et al.*: Privacy-Preserving Efficient Location-Sharing Scheme for Mobile Online Social Network Applications

IEEE *Access*

[44] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Adv. Cryptol.* Cham, Switzerland: Springer, 2004, pp. 523–540, 2004.

[45] K. Simoens, J. Bringer, H. Chabanne, and S. Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 833–841, Apr. 2012.

[46] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[47] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, Jan. 2018.

[48] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.

[49] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[50] (Sep. 2020). [Online]. Available: https://en.wikipedia.org/wiki/Cell_site

[51] Y. Jung, Y. Nam, J. Kim, W. Jeon, H. Lee, and D. Won, "Key management scheme using dynamic identity-based broadcast encryption for social network services," in *Advances in Computer Science and its Applications*, vol. 279. Berlin, Germany: Springer, 2014, pp. 435–443.

[52] F. Gunther, M. Manulis, and T. Strufe, "Key management in distributed online social networks," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Lucca, Italy, Jun. 2011, pp. 1–7, doi: 10.1109/WoW-MoM.2011.5986171.

[53] S. Venkatesan, V. A. Oleshchuk, C. Chellappan, and S. Prakash, "Analysis of key management protocols for social networks," *Social Netw. Anal. Mining*, vol. 6, no. 1, pp. 1–16, Dec. 2016, doi: 10.1007/s13278-015-0310-0.

[54] L. Guo, C. Zhang, and Y. Fang, "A trust-based privacy-preserving friend recommendation scheme for online social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 4, pp. 413–427, Jul. 2015, doi: 10.1109/TDSC.2014.2355824.

[55] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[56] P. F. Syverson and I. Cervesato, "The Logic of Authentication Protocols," in *Proc. Int. School Found. Secur. Anal. Des.*, London, U.K., 2001, pp. 63–136.

[57] M. Abadi, B. Blanchet, and H. Comon-Lundh, "Models and proofs of protocol security: A progress report," in *Proc. 21st Int. Conf. Comput. Aided Verification*, Grenoble, France, 2009, pp. 35–49.

[58] C.-C. Lee, D.-C. Lou, C.-T. Li, and C.-W. Hsu, "An extended chaotic-maps-based protocol with key agreement for multiserver environments," *Nonlinear Dyn.*, vol. 76, no. 1, pp. 853–866, Apr. 2014.

[59] X. Li, J. Niu, S. Kumari, S. H. Islam, F. Wu, M. K. Khan, and A. K. Das, "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security," *Wireless Pers. Commun.*, vol. 89, pp. 1–29, Aug. 2016.

[60] J.-L. Tsai and N.-W. Lo, "A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card," *Int. J. Commun. Syst.*, vol. 28, no. 13, pp. 1955–1963, Sep. 2015.

[61] A. Irshad, S. A. Chaudhry, Q. Xie, X. Li, M. S. Farash, S. Kumari, and F. Wu, "An enhanced and provably secure chaotic map-based authenticated key agreement in multi-server architecture," *Arabian J. Sci. Eng.*, vol. 43, no. 2, pp. 811–828, Feb. 2018.

[62] H. Wang, D. Guo, Q. Wen, and H. Zhang, "Chaotic map-based authentication protocol for multiple servers architecture," *IEEE Access*, vol. 7, pp. 161340–161349, 2019.

[63] B. Schneier, *Appl. Cryptography Protocols Algorithms Source Code C*, 2nd ed. Hoboken, NJ, USA: Wiley, 1996.

[64] D. He, N. Kumar, J.-H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 30–37, Feb. 2014.

[65] Secure Hash Standard. (Apr. 1998). *FIPS PUB 180-1, National Institute of Standards and Technology (NIST)*. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[66] Advanced Encryption Standard. (Nov. 2001). *FIPS PUB 197, National Institute of Standards and Technology (NIST)*. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

MUNMUN BHATTACHARYA received the M.E. degree in software engineering from Jadavpur University, Kolkata, in 2009. She is currently an Assistant Professor with the Department of Information Technology, Jadavpur University. She has ten years of teaching experience on different theoretical and sessional subjects, such as database management systems, operating systems, distributed systems, microprocessors, and so on. Her research interest includes designing different algorithms to provide security to online social networks data as well as users.

SANDIP ROY received the M.Tech. degree in computer science and engineering from the West Bengal University of Technology, Kolkata, India, in 2007, and the Ph.D. degree in information technology from Jadavpur University, Kolkata, in 2019. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Asansol Engineering College, India. He has authored more than 12 technical research articles published by IEEE, Elsevier, Springer, John Wiley, and so on. His research interests include security in remote user authentication, the Internet of Things (IoT), online social network security, wireless sensor network security, and the design of fine-grained access control protocols.

KAMLESH MISTRY received the Ph.D. degree from the Department of Computer and Information Sciences, Northumbria University, U.K. He is currently a Lecturer of computer science with Northumbria University. Before this, he worked as a Research Associate with Teesside University, U.K. His research interests include image feature extraction, image feature selection/optimisation, and machine learning algorithms.

HUBERT P. H. SHUM (Senior Member, IEEE) received the bachelor's and master's degrees from the City University of Hong Kong, and the Ph.D. degree from The University of Edinburgh. He is currently an Associate Professor of computer science with Durham University. Before this, he was the Director of Research/Associate Professor/Senior Lecturer with Northumbria University, a Postdoctoral Researcher with RIKEN, Japan, and a Research Assistant with the City University of Hong Kong. He led funded research projects as the Principal Investigator awarded by EPSRC, the Ministry of Defence (DASA), and the Royal Society. He has published over 100 research articles in the fields of computer graphics, computer vision, motion analysis, and machine learning.

SAMIRAN CHATTOPADHYAY (Senior Member, IEEE) received the bachelor's and master's degrees in computer science and engineering from IIT Kharagpur, India, and the Ph.D. degree from Jadavpur University, Kolkata, India. He is currently working as a Professor with the Department of Information Technology, Jadavpur University. He is having over 25 years of teaching experience at Jadavpur University, four years of industry experience, and 12 years of technical consultancy in the reputed industry houses. He has authored over 110 articles in international journals and conferences.

• • •