LPPRS: New Location Privacy Preserving Schemes Based on Ring Signature over Mobile Social Networks

Cailing Cai(✉), Tsz Hon Yuen, Handong Cui, Mingli Wu, and Siu-Ming Yiu

Department of Computer Science, The University of Hong Kong,
Pokfulam Road, Pokfulam, Hong Kong
{clcai,thyuen,hdcui,mlwu,smyiu}@cs.hku.hk

Abstract. There are two popular location-based service (LBS) applications: searching k-nearest neighbor Points of Interests (kNN POIs) and finding Nearby Friends (NF) via a social network server (SNS). Nevertheless, both applications are based on users' current locations, and no scheme has been devised yet to merge POIs, NF and SNS together. A series of works were proposed to preserve users' query privacy leaked from service attributes of POIs or location privacy over Mobile Social Networks (MSNs). However, their communication and computation costs are heavy.

In this paper, we design a novel LBS application named **NFPOI**, which allows users to search NF based on a given POI via an SNS. To preserve users' identity privacy, location privacy and query privacy, we firstly propose Location Privacy Preserving schemes based on Ring Signature (**LPPRS**). In our LPPRS, (1) Both user's real identity and real location are kept secret from others effectively. (2) Due to the anonymity of ring signature, the SNS was allowed to return query results while it cannot distinguish the real sender when processing a query message. Thus, the sender's query privacy is preserved even though the SNS knows the actual attributes and locations of POIs. (3) Neither a fully trusted third party (TTP) nor a pre-shared secret key with friends is required. A semi-TTP scheme and a TTP-free scheme were proposed respectively with different trade-offs in efficiency and security level. (4) Communication and computation costs for user side are less than existing works.

Keywords: Location privacy-preserving · Ring signature · Points of interests · Mobile social networks

1 Introduction

Location based services (LBS) are of great importance in our daily life. One LBS application is location-based searching, which allows users to query kNN POIs. For instance, Alice[1] can use the Google map to check how many bars, cinemas, or hospitals are within a radius of 3 km based on her current location.

[1] Alice represents a user or a user's device in this work.

Y. Wu and M. Yung (Eds.): Inscrypt 2020, LNCS 12612, pp. 288–303, 2021.
https://doi.org/10.1007/978-3-030-71852-7_19

Note that searching kNN POIs does not rely on MSNs, i.e., the information in Alice's social network, such as Alice's friend lists.

MSNs construct a sharing medium for individuals' daily communication. Via a SNS, provided by Twitter for instance, users can create profiles and share personal data like videos and pictures with friends in their social networks. Location sharing among social network friends is another popular function of SNS. After uploading a current location to SNS, users can query NF.

The searching goals of the above two functions are individual. One is for kNN POIs but another one is for NF. However, no LBS application achieves the two goals at the same time and thus cannot satisfy some specific cases. For example, Alice is currently in New York and she can search for friends near the hotel in London where she has booked or she can choose a hotel which is more likely near her friends living in London. Motivated by such demands, we design a novel application for searching NF based on a given POI via a SNS. We define it as **NFPOI**, which also allows users to search nearby POIs.

Since both of the LBS applications are based on a user's current and precise location, privacy concerns are raised by sensitive information leakage. The first one is **location privacy** that is revealed from the disclosure of users' exact locations. For instance, some fitness tracking APPs like Strava allows its users to record and share their jogging routes. However, in 2018, it was reported that the location of a secret US army base was leaked by the locations shared in the APP. Another one is **query privacy** leaked from the service attribute of POIs, e.g., amusement services, medical services, catering services, especially when a sender issues POIs query with the same service attribute continuously in a period. For example, if Alice frequently queries bars, an adversary can infer that Alice is an alcoholic and she may face some health issues caused by over drinking. As shown in [1], the adversary also can infer the sender's interests, health condition, eating habits, and so forth by analyzing the sender's POIs. In our LPPRS, a **continuous NFPOI query** refers that a sender continuously searches NF based on POIs with the same service attribute in a period. Otherwise, we denote that the user does a non-continuous NFPOI query.

To protect users' *location privacy*, a number of schemes are proposed, such as k-anonymity [2–5], dummy locations [6–9], obfuscation [10,11], mix zone [12,13], spatial transformation [14,15] and homomorphic encryption (HE) [16–19]. For *query privacy*, private information retrieval (PIR) proposed in [20–23] is a useful algorithm.

The k-anonymity and dummy location are applied to construct a cloak region for a user's location with $k-1$ locations, which can be obtained from the user or a TTP. Different from the dummy location algorithm, the $k-1$ locations of k-anonymity can be exact or dummy. The obfuscation algorithm is to select an appropriate location (not a cloak area) to substitute a user's exact location. For mix zone, a TTP will help a user change her identity when her location is in a specific zone, mixing the user's identity with others, but users cannot change locations. The space transformation is to map a user's location into another space with a one-way transformation. Paillier [16] and BV [17] are two popular HE algorithms applied in privacy-preserving, achieving additive homomorphic and full homomorphic respectively. The PIR allows a user to retrieve POIs from servers with indexes while the user does not reveal any content of POIs. More introductions about locations privacy protection can be found in recent surveys [24–26].

Limitations of Existing Works on Location Privacy and Query Privacy. (1) The k-anonymity algorithm in continuous queries is vulnerable to location-dependent attacks [27] and attackers can recognize users' identities with an anonymized graph [28]. (2) The accuracy is reduced when a query message is processed under a cloak region, e.g., k-anonymity and dummy location. (3) Users need to reveal their exact locations to a TTP, e.g., mix zone. (4) The communications and computations costs for users or servers sides are heavy, e.g., HE, PIR.

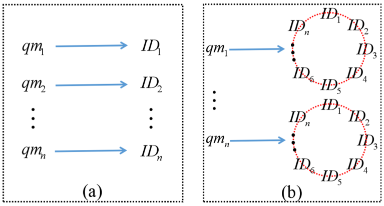


Fig. 1. Linkage between sender's identity and query message.

Furthermore, we find that the linkage between a sender and her query message is revealed to SNS directly in most existing schemes and thus allows SNS to infer more sensitive information, Fig. 1(a). For example, according to the location of POI, e.g., a bar, the sender's location, and her friends' IDs, SNS can infer that the sender's or her friends' future locations may be the specific bar with a high probability. Thus, besides location privacy and query privacy, hiding the linkage between the sender's query message and identity is essential. Since ring signature is a more powerful tool than k-anonymous to achieve anonymity, we apply a RingCT 3.0 algorithm proposed by Yuen et al. [29] to achieve anonymous query, Fig. 1(b). A comparison between k-anonymity and ring signature is presented in Table 1.

Table 1. Comparison between k-anonymity and ring signature

k-anonymity	• Anonymize a sender's exact location for LBS • Cannot resist location-dependent and anonymized graph attack for continuous queries • Cannot satisfy unconditional anonymity and cloak region easily causes vague query results
Ring signature	• Anonymize the linkage between a sender's identity and query message • SNS can only distinguish the real sender with the probability of $1/ring\_size$ even for continuous queries • Allow users to submit an exact location of POI to SNS without compromising the accuracy of query results

In this work, therefore, we design a lightweight and anonymous framework for NFPOI query, which preserves the sender's location privacy, query privacy and identity privacy simultaneously. To the best of our knowledge, no mechanism satisfying all requirements has been proposed yet.

Our Contributions. We propose location privacy-preserving schemes over MSNs based on ring signature (LPPRS) in two different security settings: with Semi-TTP and without TTP. In our scheme 1, there are three entities: User, Social Network Server (SNS) and Cloud Server (CS). CS is a semi-trusted third party. Our scheme 2 is TTP-free. The main contributions of our LPPRS are as follows.

1) **Proposed a novel LBS application over MSNs combining POI and NF together**. Our NFPOI successfully breaks the limitation of either POIs query or NF query. Via a SNS, NFPOI allows users to search NF based on a given POI. Thus, it can be used as a practical LBS searching fashion.
2) **Identity privacy and location privacy are preserved**. Instead of inputting email addresses or telephone numbers to SNS, only ring signature public keys are used to denote users' identities. Thus, users' registration IDs do not reveal any personal information to SNS. Users' location privacy is protected by submitting substitution locations to SNS. Different from algorithms discussed above, it is efficient because no heavy computation or communication cost is involved.
3) **Anonymous query and query privacy are preserved**. Due to the anonymity property of ring signature, although SNS can learn that the query message including exact location and attribute of POI is sent from the ring members, it cannot find out the real sender with an unnegligible probability. Thus, anonymous query is guaranteed in LPPRS. In addition, as it is impossible for SNS to distinguish whether two query messages are sent from the same user, query privacy is preserved after sending continuous queries.
4) **Achieved Semi-TTP and TTP-free**. In LPPRS, only a semi-trusted third party is involved in scheme 1 (Sect. 4.1). Computation cost of CS is trivial, as it only helps users select ring members and forwards messages to SNS. Apart from the sender's identity privacy, user's location privacy and query privacy will not be leaked to CS. Additionally, scheme 2 (Sect. 4.2) removes the need of utilizing a semi-trusted third party by using anonymity networks or anonymous algorithms and requiring the SNS to perform public key encryption.
5) **Achieve session key free**. Different from previous works, users do not share any session key with social network friends in advance, thus avoiding privacy leakage caused by dishonest friends when users share the session key with them.

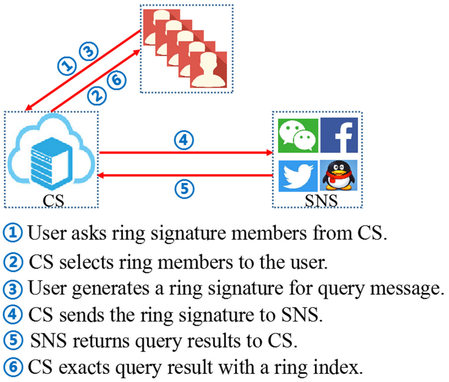
2 Preliminaries

Ring signature was proposed by Rivest et al. [30] in 2001. A ring is formed by n public keys $\boldsymbol{Y}$ among which one is the signer's public key and the remaining public keys are from $n-1$ other users. The signer generates a ring signature for a message using his own secret key. A verifier can validate the signature for the message with the ring $\boldsymbol{Y}$. The ring signature provides anonymity for the signer in the ring $\boldsymbol{Y}$ without using a trusted third party or a group manager. The unconditional anonymity of ring signature makes the attacker unable to distinguish the actual signer with probability greater than $1/n$, ever though the attacker has infinitely powerful computation and can access to an unbounded number of chosen-message signatures signed with the same ring members.

RingCT 3.0. Many ring signature schemes are proposed since the invention of ring signature. In 2019, Yuen et al. [29] proposed a new ring signature scheme named RingCT3.0 protocol to protect the privacy of a sender in Monero blockchain transaction. To the best of the authors' knowledge, it is the shortest ring signature scheme without trusted setup up to now. Thus, we use the RingCT 3.0 as a building block of our protocol[2].

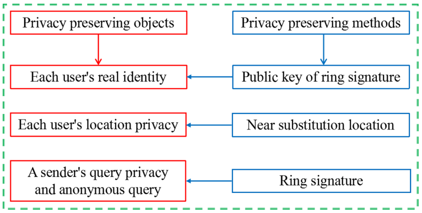
3 System Descriptions and Threat Model

3.1 System Descriptions



① ③
② ⑥
④
⑤
CS
SNS
① User asks ring signature members from CS.
② CS selects ring members to the user.
③ User generates a ring signature for query message.
④ CS sends the ring signature to SNS.
⑤ SNS returns query results to CS.
⑥ CS exacts query result with a ring index.

Fig. 2. Framework of Scheme 1.



Privacy preserving objects
Privacy preserving methods
Each user's real identity
Public key of ring signature
Each user's location privacy
Near substitution location
A sender's query privacy and anonymous query
Ring signature

Fig. 3. Privacy preserving objects and methods.

The framework of scheme 1 is shown in Fig. 2. There are three entities: Users, Social Network Server (SNS), Cloud Server (CS). Scheme 2 is TTP-free by moving the setting of CS.

Users. They can access CS and SNS via a smart device such as smartphone, smartwatch, iPad, and so on.

SNS. It carries out users' query messages based on their social network friends lists and locations.

[2] The details of RingCT 3.0 are in the full version.

CS. It assists users in three ways. (1) CS helps users select ring members for each location query. Since Alice's friend list is revealed to SNS, the ring members cannot be selected from Alice's friends simply. Otherwise, SNS can easily recognize that Alice is the sender by checking ring members' common friends. Besides, we require that the ring members must be registered users in SNS. Hence, it is not easy for Alice to construct a ring without the knowledge from users who are not her social friends. (2) CS conveys users' ring signatures to SNS. (3) CS extracts the final encrypted query results sent from SNS with a sender's ring index. It can prevent the sender from decrypting the query results of the $n-1$ decoy ring members. Therefore, this step provides protection against malicious users.

As shown in Fig. 3, the ring signature is applied to sign a sender's query message, and a ring signature public key is used to hide the sender's identity.

Substitution Location (*sl*): It is used to preserve users' exact locations. Similar to the works [10,11], we assume that there are some public buildings such as subway stations, bus stops, supermarkets, etc., around a user's current location. A nearby public location (not a cloak region) will be selected to replace the exact location in LPPRS. The choice is flexible, depending on the user's current location. For example, if Alice's current location is near a subway station exit, then that location is a better substitution.

District of Substitution Location and Ring Members: The district of substitution location represents a larger area, such as a town or a suburb. Since the location of POI is independent on the sender's substitution location, we propose that ring members are selected randomly from users who are in the same district as the sender's.

3.2 System Threat Model

The assumptions of system threat model in LPPRS are as follows.

1) The communications between three entities in LPPRS are via a secure channel. Thus, an eavesdropping attack is not considered in LPPRS.
2) Both CS and SNS are honest-but-curious, which means that they will execute schemes honestly while intend to infer more private information. In general, an entity is defined as a TTP when it knows each user's real identity, location, query message and query result, such as the setting of CT in [31]. Thus, similar to [32], we define that CS in our LPPRS is a semi-trusted entity (semi-TTP) since it does not have users' real query messages, exact locations and real query results.
3) Following to the works [31–33], we assume that CS and SNS cannot be controlled by the same adversary, because they are managed by two individual institutions. In other words, CS and SNS do not collude with each other.
4) CS and SNS can monitor users' information running in the system, respectively, including users' historical substitution locations, query messages, query results, and so on. Meanwhile, both entities receive all public parameters of algorithms applied in the mechanism.

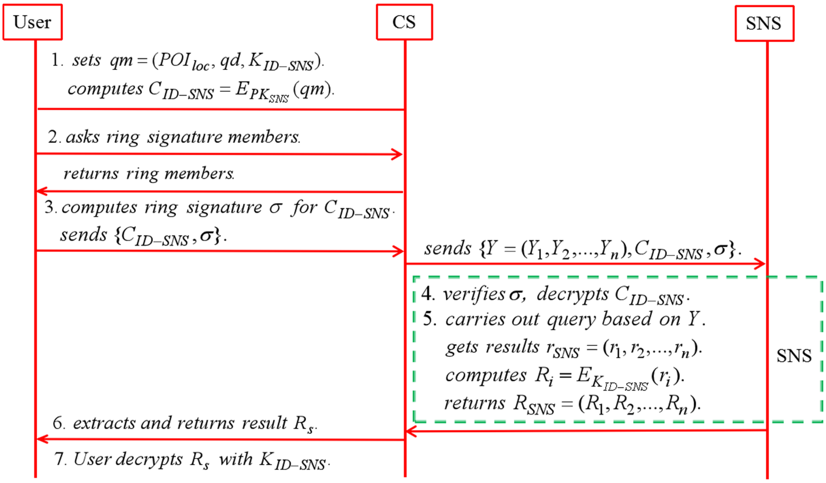

User
CS
SNS
1. sets $qm = (POI_{loc}, qd, K_{ID-SNS})$.
computes $C_{ID-SNS} = E_{PK_{SNS}}(qm)$.
2. asks ring signature members.
returns ring members.
3. computes ring signature σ for C_{ID-SNS}.
sends $\{C_{ID-SNS}, \sigma\}$.
sends $\{Y = (Y_1, Y_2, ..., Y_n), C_{ID-SNS}, \sigma\}$.
4. verifies σ, decrypts C_{ID-SNS}.
5. carries out query based on Y.
gets results $r_{SNS} = (r_1, r_2, ..., r_n)$.
computes $R_i = E_{K_{ID-SNS}}(r_i)$.
returns $R_{SNS} = (R_1, R_2, ..., R_n)$.
SNS
6. extracts and returns result R_s.
7. User decrypts R_s with K_{ID-SNS}.

Fig. 4. Steps of NFPOI query in scheme 1.

4 Our LPPRS

4.1 Construction of Scheme 1

Registration in Social Network Server. Alice's registration identity (ID) in SNS is a ring public key Y_A. We suppose that each user's ID is different from others. In LPPRS, based on the location of a POI, uploading personal location to SNS is not the prerequisite for NFPOI query. Thus, if Alice is willing to reveal it to social network friends, she can upload a substitution location sl_A to SNS.

Registration in Cloud Server. Alice's ID in CS is also Y_A. Once Alice updates her ID in SNS, she will send the new ID to CS simultaneously. Different from SNS, CS records Alice's ring and corresponding index of her query message. Besides, instead of sending a sl_A, Alice only sends the district of sl_A to CS once she has updated her location in SNS.

Query Steps. There are seven steps in scheme 1, seeing Fig. 4.

- **Step 1: Alice sets qm.** Firstly, to prevent CS from knowing query results sending from SNS, Alice randomly generates a one-time-key K_{ID-SNS} of AES[3]. Instead of sending K_{ID-SNS} to SNS directly, Alice adds the key into a query message denoted by $qm = (POI_{loc}, qd, K_{ID-SNS})$, where POI_{loc} is the exact location of POI, and qd is a radius of query distance. Secondly, Alice encrypts qm to get $C_{ID-SNS} = E_{PK_{SNS}}(qm)$, where PK_{SNS} is a RSA[4] public key of SNS.

[3] AES represents a symmetric encryption algorithm in this work.

[4] RSA represents an asymmetric encryption algorithm in this work. Note that RSA can be replaced by Elliptic Curve Cryptography or other asymmetric encryption algorithms in trade-offs in efficiency and security.

Suppose that Alice desires to search kNN friends within $qd = 2\,\text{km}$, denoted the query region by $\odot_{qd}$. Due to the distance between a user's real location and substitution location, a border case is that the user's substitution location is outside $\odot_{qd}$, while the user's real location is inside $\odot_{qd}$. Since SNS performs qm based on users' substitution locations, SNS will not add the user to its query result. For this case, we propose that qd sent to SNS is larger than 2 km, e.g., 4 km (double times), flexibly avoiding omitting all kNN friends within $\odot_{qd=2km}$. Besides, we set that results returned by SNS are recorded increasingly based on the distance among POI_{loc} and her friends' locations (seeing Step 5). Thus, Alice can quickly learn about whose location is around and can obtain those friends' exact locations by privately communicating with them.

- **Step 2: Alice asks ring Y from CS.** Based on Alice's district, CS selects $n-1$ ring members from its M_{CS} randomly and keeps the ring index. After that, CS sends ring $Y = (Y_1, Y_2, ..., Y_n)$ to Alice. On the other hand, if Alice does continuous queries, then ring Y and its indexes are the same as the first time during the whole period. Otherwise, ring Y is deleted by CS and Alice after a NFPOI query.
- **Step 3: Alice computes σ for C_{ID-SNS}.** Once Alice obtains the ring Y from CS, Alice keeps her index secretly. Based on RingCT 3.0, Alice computes a ring signature σ for C_{ID-SNS}. After that, Alice sends (C_{ID-SNS}, σ) to CS. For CS, once it receives Alice's query message, it firstly records Alice's ring Y and index. Later, CS sends $(C_{ID-SNS}, \sigma, (Y_1, Y_2, ..., Y_n))$ to SNS.
- **Step 4: SNS verifies σ.** For σ, if it is valid, SNS decrypts C_{ID-SNS} with its RSA private key to get the query message, and keeps the session key K_{ID-SNS} secretly. Otherwise, SNS rejects the query.
- **Step 5: SNS performs qm based on ring Y.** Firstly, due to the anonymity of ring signature, SNS cannot find out that $qm = (POI_{loc}, qd, K_{ID-SNS})$ is sent from Alice. Thus, SNS carries out qm based on $(Y_1, Y_2, ..., Y_n)$ and records results from Y_1 to Y_n sequentially. Denoted query results for ring Y by $r_{SNS} = (r_1, r_2, ..., r_n)$ and r_s is the query result for ring member in Y with index $s, (s = 1, ..., n)$. In general, r_s is a set and its each element is in the form of (d_t, ID_t, sl_t), where t is a number of Y_s' friends whose substitution locations are in $\odot_{qd=4km}$, d_t is a distance satisfying $d_t = dist(POI_{loc}, sl_t) < qd$, and ID_t represents a Y_s' friend. We require that SNS records results according to d_t increasingly. The smaller value of d_t implies the nearer friend. Secondly, SNS encrypts each r_i with the session key K_{ID-SNS} and gets $R_i = E_{K_{ID-SNS}}(r_i)$. Denoted the ciphertext results by $R_{SNS} = (R_1, R_2, ..., R_n)$. Finally, SNS sends R_{SNS} to CS.
- **Step 6: CS extracts result R_s.** After receiving $R_{SNS} = (R_1, R_2, ..., R_n)$ from SNS, CS exacts the result R_s with index s, sends it to Alice, and discards the rest results.
- **Step 7: Alice decrypts R_s.** Finally, Alice can learn about how many friends are nearby the POI by decrypting R_s with K_{ID-SNS}.

4.2 Construction of Scheme 2

Registration in SNS. Firstly, each user generates a RSA public key, denoted as RSA_{ID}. Secondly, since users' IDs are ring public keys, we assume that all IDs in SNS are public. Besides, users' location districts are also published by SNS. Thus, Alice's public information is in the form of (Y_A, RSA_{Y_A}, country/city, district), if she has updated a location to SNS. Otherwise, her public information is (Y_A, $RSA_{Y_A}, \perp, \perp$). Note that all users' social relationships are not published.

Query Steps. Without the setting of CS, there are six steps in scheme 2.

- **Step 1: Alice sets** qm**.** In scheme 2, a symmetric random private key is removed from qm, $qm = (POI_{loc}, qd)$. Next, Alice encrypts qm to get $C_{ID-SNS} = E_{PK_{SNS}}(qm)$.
- **Step 2: Alice selects ring** Y **personally.** Based on the public information offered by SNS, the same as scheme 1, Alice randomly selected ring Y from the same district with her location. If Alice needs continuous queries, then she will keep ring Y and use it to sign new qm during the period of continuous queries. Otherwise, ring Y is deleted after obtaining query results.
- **Step 3: Alice computes** σ **for** C_{ID-SNS}**.** Firstly, Alice computes a ring signature σ for C_{ID-SNS}, and sends message $\{C_{ID-SNS}, \sigma, (Y_1, Y_2, ..., Y_n))\}$ to SNS. Secondly, similar to [18], we assume that the communication between Alice and SNS is via anonymized algorithms [34] or an anonymized network (e.g., Tor[5]).
- **Step 4: SNS verifies** σ**.** (This step is the same as scheme 1.)
- **Step 5: SNS carries out** qm **based on ring** Y**.** Different from scheme 1, SNS encrypts query results $r_{SNS} = (r_1, r_2, ..., r_n)$ with each ring member's RSA_{ID}, getting $R_s = E_{RSA_{Y_s}}(r_s)$, $s = 1, ..., n$. Next, SNS returns $R_{SNS} = (R_1, R_2, ..., R_n)$ to the sender.
- **Step 6: Alice decrypts the query result.** After obtaining R_{SNS}, Alice selects the result with her ring index, and decrypts it with RSA secret key. Note that even though Alice can obtain all ring members' results, she only can obtain her own friends' information by decrypting the result with personal RSA private key.

Note that following to scheme 2, SNS also can apply RSA to encrypt $r_{SNS} = (r_1, r_2, ..., r_n)$ in scheme 1, while considering the setting of CS and the efficiency of AES, we adopt AES to encrypt query results instead of RSA for scheme 1.

5 Schemes Comparison and Security Analysis

5.1 Scheme Comparison

Comparisons among LPPRS and other schemes are shown in Table 2.

[5] https://www.torproject.org/.

Table 2. Comparison between LPPRS and other schemes. The following symbols are used: DS: Digital Signature, FL: friends' list, qt: query type, qd: query distance, qm: query message, σ: ring signature, f: the number of friends, [$\#x$]: runs for x-times.

Scheme	User	User	Server(s)	Server(s)
	Comp. cost	Comm. cost	Comp. cost	knows
[31] U-CT-SNS-LS (TTP: CT)	DS.Sign AES[#1+f]	ID||qt||qd	CT: pseudonyms & dummy loc RSA.Enc & RSA.Dec SNS: DS.Verify LS: RSA.Enc	CT & SNS: ID & fake IDs CT: loc & dummy locs SNS: FL LS: fake IDs & dummy locs
[33] U-SNS-LS	RSA.Enc AES[#4]	ID||qt||qd||loc.cipher	SNS: pseudonyms & k-anonymity LS: RSA.Dec & AES[#2]	SNS: ID & fake IDs & FL LS: fake ID & real loc
[35] U-SNS-LS	Broadcast Enc, DS.Sign AES[# > 2+f]	ID||qt||qd	SNS: pseudonyms & DS.Verify LS: AES[#2] & DS.Sign	SNS: ID & fake IDs & FL LS: fake ID & real loc
[36] U-SNS	ORE.Enc[#2] ORE.QGen AES[# > f]	multi-qm.cipher	SNS: ORE.Cmp Index construction Index maintenance	SNS: ID & loc.cipher
[18] U-SNS	CP-ABE Paillier HE Functional Enc	(To a friend) multi-times comm.	negligible (mainly computed by users)	SNS: ID & FL
[1] U-LS	DUMMY-Q technique	multi-(loc||POIs)	LS: multi-query processing	LS: ID & real loc
[32] U-SA-LS (Semi-TTP : SA)	Hilbert Curve RSA.Enc	loc||POI	SA: anonymity area compute redundant results LS: RSA.Dec & loc transform	SA: ID
[37] U-LS	RSA.Enc[#2], RSA.Dec[#2] Bilinear Pairing[#n] Deniable Authentication	multi-times comm.	LS: RSA.Dec[#2] RSA.Enc[#2] Bilinear Pairing	LS: ID & real qm
Ours. 1 U-CS-SNS (Semi-TTP : CS)	RSA.Enc Ring.Sign AES	qm.cipher||σ	SNS: RSA.Dec Ring.Verify AES[#n]	CS: sender's ring index SNS: FL & real qm
Ours. 2 U-SNS	RSA.Enc Ring.Sign RSA.Dec	qm.cipher||σ	SNS: RSA.Dec Ring.Verify RSA.Enc[#n]	SNS: FL & real qm

- Column 1 (Scheme): For each scheme, we summarize the involved entities such as User (U), SNS, location server (LS), cloud server (CS). CT represents Cell Tower in [31] and SA represents Semi-Anonymizer in [32]. We also describe the type of TTP used if there is one.
- Column 2 (A user's comp.cost[6]): The cryptographic operations computed by a user are listed. We use [#] to represent the number of times when an algorithm runs by the user multiple times. For example, in [31], the sender runs the AES once in registration period. Besides, a query result includes several locations, encrypted by friends' private keys respectively. Thus, the sender totally needs to perform the AES for (1+f) times, denoted as AES[#1+f], where f is the number of friends of a query result.
- Column 3 (A user's comm.cost[7]): To simplify, we only compare a user's comm.cost of sending query messages, excluding registration and location updating periods. Note that the loc.cipher and qm.cipher represent the ciphertext of location and query message respectively. The multi-times comm. means there are multiple communications between two entities. Unless otherwise specified, the user sends the query to the party connected to U in column 1.
- Column 4 (Comm.cost of server(s)): Its description is similar to column 2.
- Column 5 (Sever knows): We summarize a user's privacy that is revealed to server(s).

Detailed comparisons in different perspectives are given as follows.

TTP. Our scheme 1 and [32] have a semi-TTP, which both cannot obtain users' real IDs and locations, but the semi-TTP in [32] needs to help users perform extra computations for query results. [1,37] are TTP-free schemes, but [1] only focus on preserving user's query privacy, and user's comp.cost and comm.cost are all heavy in [37]. Our scheme 2 is TTP-free, offering privacy-preserving for a user's identity, location and query message simultaneously.

Comp.cost (User and server(s)). In our LPPRS, for each query, the sender only needs to compute the RSA, AES and ring signature one time, respectively. However, in [31,33,35,36], the sender needs to run the RSA or AES several times. Besides, due to the running costs of CP-ABE/Hilbert curve/Bilinear Pairing, user's comp.cost from [18,32,37] are significant. Different from [31,33,35,36], comp.cost for server side in [18] is negligible, since the computation is mainly done by two parties for each query. Comparing to [37], our LPPRS is lightweight as the server does not need to perform the bilinear pairing operations.

Comm.cost (User). In our LPPRS, a query message sent to the server only includes a RSA ciphertext and a ring siganture. However, in [1,36], the sender's query message either contains multiple dummy POIs or multiple locations encrypted with AES. For [18,37], the user has to interact with her friend or the server multi-times. Thus, user's comm.costs in [1,18,36,37] are all heavy.

Server(s) Knows. In our LPPRS, a user's real ID and location are not revealed to any party as they are preserved by a ring public key and a substitution location respectively. However, at least one server knows a user's real identity or location in [1,18,31–33,35–37]. For query privacy, our schemes and [37] allow SNS to obtain an anonymous query message in the form of plaintext. Our LPPRS is based on the ring signature that preserves query privacy perfectly, while [37] enables the sender to deny her behavior when the server tells her data to others, with a deniable ring authentication algorithm.

Searching Method and Session Key. Based on users' current locations, schemes [18,31,33,35,36] and [1,32,37] are designed to offer privacy-preserving for searching kNN NF and POIs respectively. Our NFPOI focuses on NF searching based on a given POI via SNS. In addition, different from [31,33,35,36], we do not require users to share session keys with friends, successfully avoiding privacy leakage from malicious users. Due to the length limitation, we do not show both items in Table 2.

5.2 Security Analysis

In this section, we analyze that the sensitive information that CS and SNS intend to infer is preserved when they perform inference attacks.

For SNS, it knows all users' friends lists, query messages, ring members and some users' substitution locations, while it desires to infer the real sender and users' exact locations. For CS, it stores users' historical and current districts of substitution locations, encrypted query messages, encrypted query results, ring members and ring indexes, while it hopes to acquire the plaintext of users' query message, query results and exact locations.

Inference Attack Resistant: A mechanism is inference attack resistant if an adversary in probabilistic polynomial time cannot infer a user's real value over a possibility ϵ, where ϵ depends on the secure parameter of a specific privacy preserving algorithm.

Property 1. Our LPPRS is inference attack resistant to SNS.

(1) Given a query message signed by Alice Y_A with a ring $Y = (Y_1, Y_2, ..., Y_n)$, the possibility that SNS infers the real sender is $\epsilon = \frac{1}{n}$.

Analysis 1: Firstly, if Alice does not need continuous query, different query message is signed with different ring $Y = (Y_1, Y_2, ..., Y_n)$. Each ring member in Y is selected from whole registration IDs of SNS, as long as they are in the same district as Alice. Besides, the location of POI is independent on Alice's substitution location. Thus, SNS cannot find out Alice by matching each ring member's substitution location with the location of POI. In addition, each ring Y is generated randomly and the ring members are not chosen from Alice's social network friends. Thus, even though SNS has all users' social friends lists, it cannot recognize Alice by checking ring members' common friends, or via performing joint analysis based on a large number of ring signatures.

Secondly, if Alice needs a continuous query, all of her query messages are signed with the same ring and index. Hence, due to the perfect anonymity of ring signature, it is impossible for SNS to find out whether two query messages are sent from the same user. Therefore, without the knowledge of the ring index, even though SNS obtains POI and its exact location, it only has the possibility of $\frac{1}{n}$ to identify Alice as the real sender.

(2) Given a substitution location sending from Alice, the possibility that SNS deduces Alice's exact location is $\epsilon = \frac{1}{w}$.

Analysis 2: Suppose Alice's substitution location is a subway station, and there are 'w' buildings around it. Since the substitution location is selected by Alice secretly, SNS can infer Alice's exact location with the possibility of $\frac{1}{w}$ at most, even though SNS knows that what buildings are near the subway station.

Property 2. Our LPPRS is inference attack resistant to CS.

Analysis 3: As a semi-TTP, CS receives query messages from users and query results from SNS. For Alice's query message $qm = (POI_{loc}, qd, K_{ID-SNS})$, it is encrypted with PK_{SNS}. The corresponding private key of PK_{SNS} is kept by SNS secretly, so CS only owns the ciphertext of Alice's query message.

For query results $R_{SNS} = (R_1, R_2, ..., R_n)$ sending from SNS, they are encrypted by SNS with a systematic key K_{ID-SNS}, generated by Alice secretly and randomly. Hence, given R_{SNS}, without the knowledge of K_{ID-SNS}, CS cannot obtain the plaintexts of them.

For users' locations, CS only obtains districts of users' substitution location, so the possibility that CS can infer Alice's real location is far less than $\frac{1}{w}$.

From analysis 1 and analysis 3, we can conclude that Alice's query privacy and the linkage between her ID and query messages are preserved anonymously. From analysis 2, we can deduce that users' location privacy is also preserved.

6 Evaluation

This section shows that our LPPRS are practical, via evaluating communication and computation costs for the user side and server side, respectively.

- **Comm.cost**: RSA, AES and ring signature (RS) are three main algorithms applied in LPPRS. For RSA and AES, the key length is represented by 2048 bytes and 256 bytes respectively. For 2048-byte RSA with PKCS#1 padding, the ciphertext size is 256 bytes for every 245 bytes message. For a ring size of n, the ring signature size of RingCT 3.0 is $2\lceil\log_2(n)\rceil + 7$ elements in $\mathbb{G}$ and 7 elements in $\mathbb{Z}_p$. Based on Curve 25519, each element in $\mathbb{G}$ and $\mathbb{Z}_p$ has the length of 33 bytes and 32 bytes respectively. Thus, we have $|\sigma| = (2\log n + 7) * 33 + 7 * 32 = 66\log n + 455$ bytes. For a ring size of 1024, the signature is 1115 bytes.
- **Comp.cost (User):**
  - RSA.Enc. It is used to encrypt query message, $C_{ID-SNS} = E_{PK_{SNS}}(qm)$.
  - RS.Sign. To sign C_{ID-SNS}, a ring signature of RingCT 3.0 is dominated by 3 multi-exponentiations in $\mathbb{G}$ of size $2n + 1$, $2n$ and $n + 1$ respectively, where n is the size of ring members.
  - AES.Dec. It is performed to get final result r_s. (Scheme 1)
  - RSA.Dec. It is performed to get final result r_s. (Scheme 2)

    Note that the above computations can be done offline by users.
- **Comp.cost (SNS)**
  - RS.Verify. It is dominated by 2 multi-exponentiations in $\mathbb{G}$ of size $2n + 2log_2 n + 1$ and $n + 4$ respectively.
  - RSA.Dec. SNS applies it to decrypt C_{ID-SNS} and obtain qm.
  - Perform qm. SNS calculates results r_{SNS} based on qm and ring Y.
  - AES.Enc. To obtain ciphertexts R_{SNS} of r_{SNS}. (Scheme 1)
  - RSA.Enc. To obtain ciphertexts R_{SNS} of r_{SNS}. (Scheme 2)
- **Comp.cost (CS):** CS does not need to perform any cryptographic algorithm. It just needs to select ring members and forward information between users and SNS.
- **The total running time of LPPRS:** The running time of RS.Sign and RS.Verify of RingCT 3.0 for different ring members n are given in [29]. Referring to the test data of AES and RSA algorithms providing by Crypto++ library[8], the running time in LPPRS for AES or RSA algorithm is negligible. Thus, the total running time of SNS (T_{SNS}) is mainly dominated by the time of RS.Verify ($T_{RS.Verify}$) and the computing time of query message (T_{qm}),

[8] https://www.cryptopp.com/benchmarks.html.

$T_{SNS} \approx T_{RS.Verify} + T_{qm}$. Based on RingCT 3.0, even if the size n of a ring is 1000, its' verification time is less than 3 s. Thus, $T_{RS.Verify}$ does not increase T_{SNS} remarkably. For T_{qm}, it is reasonable to set that SNS calculates results for ring members simultaneously, instead of one by one. Therefore, we can conclude that our LPPRS is practical to protect users' privacy with the ring signature.

7 Conclusion

In this paper, we present a new LBS application named NFPOI, which firstly combines SNS with POI and NF. Additionally, two privacy preserving frameworks (semi-TTP and TTP-free) based on ring signature are proposed in our LPPRS, aiming to offer anonymity for a sender's query message, and preserve the sender's location privacy and query privacy efficiently.

Firstly, ring signature is applied to sign the ciphertext of a query message. Based on the anonymity of ring signature, LPPRS supports SNS to return query results for a query message while it cannot find out who is the real sender. Thus, query privacy is preserved even when the sender does continuous queries. Secondly, a lightweight location privacy preserving algorithm called substitution location is applied to hide users' real locations. Thirdly, no entity in LPPRS is assumed fully trusted and the pre-sharing session key for friends is not required. Furthermore, our LPPRS is secure under inference attacks. Finally, users' communication costs and computation costs are lower than previous works according to comparisons shown in Table 2.

In LPPRS, the anonymity of a query message is related to the size of ring members n, which also influences the computations costs of SNS. Thus, the balance between the anonymity and the ring size n is a trade-off.

References

1. Pingley, A., Zhang, N., Fu, X., Choi, H.-A., Subramaniam, S., Zhao, W.: Protection of query privacy for continuous location based services. In: 2011 Proceedings IEEE INFOCOM, pp. 1710–1718. IEEE (2011)
2. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002)
3. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42 (2003)
4. Yang, D., Fang, X., Xue, G.: Truthful incentive mechanisms for k-anonymity location privacy. In: 2013 Proceedings IEEE INFOCOM, pp. 2994–3002. IEEE (2013)
5. Niu, B., Li, Q., Zhu, X., Cao, G., Li, H.: Achieving k-anonymity in privacy-aware location-based services. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 754–762. IEEE (2014)
6. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: ICPS 2005. Proceedings. International Conference on Pervasive Services, 2005, pp. 88–97. IEEE (2005)

7. Lu, H., Jensen, C.S., Yiu, M.L.: PAD: privacy-area aware, dummy-based location privacy in mobile services. In: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access, pp. 16–23 (2008)
8. Liu, H., Li, X., Li, H., Ma, J., Ma, X.: Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
9. Sun, G., Song, L., Liao, D., Hongfang, Yu., Chang, V.: Towards privacy preservation for "check-in" services in location-based social networks. Inf. Sci. **481**, 616–634 (2019)
10. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 177–189 (2004)
11. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) Pervasive 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005). https://doi.org/10.1007/11428572_10
12. Beresford, A.R., Stajano, F.: Mix zones: user privacy in location-aware services. In: IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, pp. 127–131. IEEE (2004)
13. Freudiger, J., Shokri, R., Hubaux, J.-P.: On the optimal placement of mix zones. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 216–234. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03168-7_13
14. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 239–257. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73540-3_14
15. Hu, H., Xu, J., Ren, C., Choi, B.: Processing private queries over untrusted data cloud through privacy homomorphism. In: 2011 IEEE 27th International Conference on Data Engineering, pp. 601–612. IEEE (2011)
16. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
17. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29
18. Li, X.-Y., Jung, T.: Search me if you can: privacy-preserving location query service. In: 2013 Proceedings IEEE INFOCOM, pp. 2760–2768. IEEE (2013)
19. Novak, E., Li, Q.: Near-pri: private, proximity based location sharing. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, pp. 37–45. IEEE (2014)
20. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 121–132 (2008)
21. Khoshgozaran, A., Shirani-Mehr, H., Shahabi, C.: SPIRAL: a scalable private information retrieval approach to location privacy. In: 2008 Ninth International Conference on Mobile Data Management Workshops, MDMW, pp. 55–62. IEEE (2008)
22. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest neighbor search with strong location privacy. Proc. VLDB Endow. **3**(1–2), 619–629 (2010)

23. Paulet, R., Kaosar, M.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. IEEE Trans. Knowl. Data Eng. **26**(5), 1200–1210 (2013)
24. Gupta, R., Rao, U.P.: An exploration to location based service and its privacy preserving techniques: a survey. Wirel. Pers. Commun. **96**(2), 1973–2007 (2017)
25. Liu, B., Zhou, W., Zhu, T., Gao, L., Xiang, Y.: Location privacy and its applications: a systematic study. IEEE Access **6**, 17606–17624 (2018)
26. Almusaylim, Z.A., Jhanjhi, N.Z.: Comprehensive review: privacy protection of user in location-aware services of mobile cloud computing. Wirel. Pers. Commun. **111**(1), 541–564 (2020)
27. Liao, D., Li, H., Sun, G., Anand, V.: Protecting user trajectory in location-based services. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2015)
28. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 173–187. IEEE (2009)
29. Yuen, T.H., et al.: RingCT 3.0 for blockchain confidential transaction: shorter size and stronger security. Technical report, Cryptology ePrint Archive, Report 2019/508. To appear in FC 2020 (2019)
30. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
31. Wei, W., Xu, F., Li, Q.: Mobishare: flexible privacy-preserving location sharing in mobile online social networks. In: 2012 Proceedings IEEE INFOCOM, pp. 2616–2620. IEEE (2012)
32. Peng, T., Liu, Q., Wang, G., Xiang, Y., Chen, S.: Multidimensional privacy preservation in location-based services. Futur. Gener. Comput. Syst. **93**, 312–326 (2019)
33. Liu, Z., Li, J., Chen, X., Li, J., Jia, C.: New privacy-preserving location sharing system for mobile online social networks. In: 2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 214–218. IEEE (2013)
34. Liu, Y., Han, J., Wang, J.: Rumor riding: anonymizing unstructured peer-to-peer systems. IEEE Trans. Parallel Distrib. Syst. **22**(3), 464–475 (2010)
35. Li, J., Yan, H., Liu, Z., Chen, X., Huang, X., Wong, D.S.: Location-sharing systems with enhanced privacy in mobile online social networks. IEEE Syst. J. **11**(2), 439–448 (2015)
36. Schlegel, R., Chow, C.-Y., Huang, Q., Wong, D.S.: Privacy-preserving location sharing services for social networks. IEEE Trans. Serv. Comput. **10**(5), 811–825 (2016)
37. Zeng, S., Yi, M., He, M., Chen, Y.: New approach for privacy-aware location-based service communications. Wireless Pers. Commun. **101**(2), 1057–1073 (2018)