Privacy-preserving Attribute-based Friend Search in Geosocial Networks with Untrusted Servers

Linke Guo*, Xiaoyan Zhu‡, Chi Zhang†, and Yuguang Fang*
*Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
‡National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
†School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China
Email: blackglk@ufl.edu, xyzhu@mail.xidian.edu.cn, chizhang@ustc.edu.cn, fang@ece.ufl.edu

Abstract—**Location-based Services (LBSs) enable mobile users to request and obtain certain services based on their current locations, such as finding nearby gas station, looking for coffee shops, and using online GPS navigation, etc. As a major branch of LBSs, geosocial networking services, such as Foursquare, become popular due to the explosive growth of smartphone users. Geosocial networking services allow people to use their location information to find potential friends who have similar interests within close proximity and initiate communications with each other. However, most existing geosocial networking services ask for mobile users' current location information and store it on an untrusted server with less privacy concerns. To some extent, mobile users need to reveal their interests and physical location information to a service provider in order to realize the functionality of geosocial networking, which apparently deteriorates users' privacy on the aspects of their profiles and locations. In this paper, we propose a privacy-preserving friend search scheme in geosocial networks without relying on a trusted centralized server. Our scheme lets localization infrastructures, such as base stations, create encrypted searchable tables on an untrusted server and allow mobile users to search for their possible friends using their profiles without exposing their location information. Extensive trace-driven simulation results and analysis show both the efficiency and privacy preservation of our proposed scheme.**

Index Terms—**Location Privacy, Geosocial Networking, Dynamic User Update.**

I. Introduction

Location-based services (LBSs), which utilize mobile users' current locations and tremendous web-based information, provide people with great benefits when they query their needs to service providers, e.g., finding the nearest fast food restaurants when a tourist travels to an unfamiliar city, or checking the newest film released in the nearby theater. According to a recent statistics [1] reported by Foursquare[1], there are more than 25 million registered users who create more than 2.5 billion check-ins on Foursquare by the year of 2012. Among these check-ins, mobile users are freely to choose services featured locations may provide, or, more importantly, they can initiate communication with other check-in users, which forms the geosocial networks (location-based social networking services). Other than traditional online social networks, such as Facebook, Twitter, and LinkedIn, geosocial networking services mainly focus on the social interactions among mobile users who stay in close proximity. In the current geosocial networks, mobile users obtain location information from localization infrastructures, like base stations (service carriers), wireless hotspots (Internet Service Providers), and GPS signals (satellites). Then, they upload it to the service provider, such as Foursquare, in order to find nearby check-in users and socialize with people having similar interests or profiles. However, the above process obviously reveals mobile users' privacy in terms of location, users' profiles, and their communication details, since both the friend search and communication processes are taken place on untrusted servers. For example, imagine Alice, who is a travel enthusiast, wants to find other travelers with the same interests, say, Bob, in a national park, and she also prefers to hide both her attributes or interests (travel) and location information from the untrusted server for privacy concerns. Current proposed schemes utilize anonymization techniques try to obfuscate users' identities and their location information, but these approaches apparently reveal users' sensitive information to the service provider, which not only violates the user-centric privacy requirements, but also lowers the possibility of finding the particular user, Bob. Thus, we propose a privacy-preserving friend search scheme in geosocial networks based on users' attributes, which hides user's identities and their attributes from the untrusted server using verified location information provided by localization infrastructures.

Related Works: Location privacy is defined in [2] by Beresford and Stajano, as "the ability to prevent other parties from learning one's current or past location". Existing approaches providing the location privacy of mobile users can be divided into two categories: anonymity based approaches and obfuscation based approaches. Several works [3]–[5] leverage trusted centralized servers or mix zone model to anonymize mobile users' identities using pseudonyms, which intends to provide k-anonymity [6]. The obfuscation based approaches [7]–[9] downgrade the quality of users' location information to protect location privacy. However, their schemes only provide users' location privacy without considering the privacy in terms of communications among mobile users. Moreover, the above approaches lower the quality of services and bring redundancy to the system, which is far away from the design of an ideal system. In [10], Vicente *et al.* discuss the possible privacy leakage in geosocial networks, such as location, absence, co-location, and identity privacy. Unfortunately, these works have not provided detailed constructions and proofs, which may lead to unexpected security breaches when we jointly consider users' identity and location privacy.

Our Contributions: Our major contributions are as follows:

- We design a dynamic searchable encryption scheme using the verifiable location information to allow mobile users to search for friends, while maintaining location confidentiality to the untrusted server.
- We enable mobile users to search for friends based on attributes of their profiles without revealing location information to untrusted servers.
- To better adapt the frequent check-ins and check-outs in geosocial networks, we design several user update mecha-

This work was partially supported by US National Science Foundation under grant CNS-0916391. The work of Dr. Zhu was partially supported by the National Natural Science Foundation of China under Grant 61003300, Fundamental Research Funds for the Central Universities under Grant K5051201041, and the China 111 Project under Grant B08038. The work of Dr. Zhang was partially supported by National Natural Science Foundation of China under Grant 61202140.

[1] http://www.foursquare.com

nisms to help improve the scalability of the scheme.

- Extensive trace-driven simulations are deployed to verify the performance of our protocols on the aspects of security, efficiency, and feasibility.

The remainder of this paper is organized as follows. Section II introduces our motivation in designing the scheme. We describe the system model in Section III, along with the design objectives. The proposed scheme is presented in detail in Section IV, followed by the protocol evaluation in Section V. Finally, Section VI concludes the paper.

II. Motivation

To better elaborate our proposed scheme, we first highlight our intuitive idea of user-centric friend search in geosocial networking. For most online social networks, users post messages and find friends based on their own interests. Similarly, for geosocial networking services, mobile users also have the same intuition to make friends with each other according to current location and user-centric interests. It has been shown in [11] that individuals often befriend those who have similar interests, perform similar actions and frequently meet each other. Such an observation is called *homophily phenomenon*. According to [12], users who share similar interests in their profiles tend to form groups at certain geographic areas, where they can exchange messages and communicate with each other more efficiently. However, for real applications, although mobile users check in at certain places, most of them may not have any behavior related to the specific featured locations. For example, Alice checks in near a hospital, which does not directly reflect the identity of Alice on whether she is a doctor, a nurse, or even a patient. Hence, we conduct an experiment to explore the correlation between mobile users' check-ins and their real behaviors (we refer mobile users who leave *tips* as checked users).

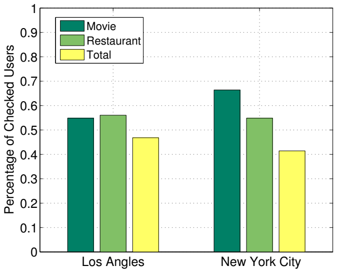


Fig. 1. Check-in Users Vs. Checked Users

We analyze a recent Foursquare dataset [13] collected at Los Angles and New York City, which includes 101,346 records generated by 49,062 users and 63,621 records generated by 31,544 users, respectively. As shown in Fig. 1, mobile users who leave *tips* in theaters take 53.5% and 66.4% of the check-in users in LA and NYC, respectively. Mobile users who have interests in restaurants and have dinners in their surroundings are about 56.1% and 54.8% of the check-in users in LA and NYC, respectively. As a total, more than 40% mobile users become checked users when they check in at certain places in both cities. The tips have their own advantages in reflecting user's real interests, which usually means a user has carried out some essential activities at featured locations.

Therefore, if we consider checked users to have same attributes at featured locations, it would be useful to design a friend search scheme among checked users in a close proximity, which renders them a way to easily communicate.

III. System Model

A. Network Model

We first give a brief introduction to the network model of our proposed scheme. As shown in Fig. 2, the system mainly consists of a central authority (CA), localization infrastructure, geosocial networking server, and mobile users. The central authority is a fully-trusted entity which is responsible for system setup and parameter distribution for mobile users. In our scheme, CA can go offline after the system starts, unless there is a new valid user comes in. Similar to current location-based services, our scheme relies on a fully-trusted localization infrastructure to pinpoint mobile users' physical location, and it will not leak the location information to third parties. Note that the localization infrastructure can be found in the existing communication systems, such as base stations in the cellular networks, wireless access point and the corresponding ISPs, GPS signal, and RFID tags used to track users' location. Since mobile users are inevitably involved in the communication with the infrastructures if they want to obtain the location-based services, it is infeasible to provide users' location privacy when they obtain services via the localization infrastructure. To provide attribute-based friend search, the system deploys an untrusted geosocial networking server to enable mobile users to be involved in the geosocial networking. For mobile users, applying common access technologies, we assume they can communicate with each other using the platform provided by the geosocial networking server. Note that we use geosocial networking servers and untrusted servers interchangeably in the elaboration of our scheme.

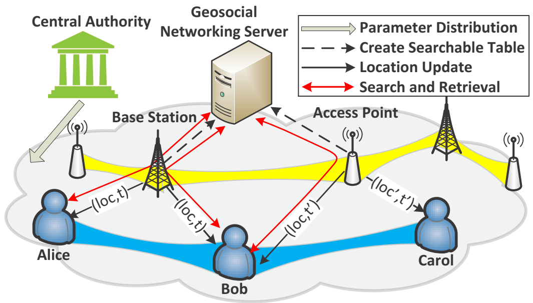


Fig. 2. System Model

B. Design Objectives

The major objective of the proposed scheme is to enable mobile users to search for friends with similar social attributes in a close geographic area while maintaining both their location privacy and profile privacy. We require that the untrusted server is not allowed to learn the content of users' identities and locations, nor can it be relied on storing users' social attributes or personal profiles. Also, our approach should enable mobile users to search over encrypted data stored on the untrusted server in order to maintain the original functionality of geosocial networking, say, making friends.

C. Adversarial Model

In this subsection, we describe the adversarial model that our system is able to thwart. Considering a set of malicious mobile users, they may actively fabricate their spatial and temporal information, which will largely deteriorate the accuracy and availability of geosocial networking. Also, we assume active adversaries intend to observe other mobile users' locations in order to stalk the corresponding users for malicious purposes. In terms of passive attacks, we assume the untrusted server is curious but honest, in the sense that it will follow the proposed protocol, but may analyze the statistic of encrypted data. The untrusted server is able to learn the encrypted indices and the corresponding ciphertexts.

On the other hand, we exclude several possible attacks, all of which are beyond the scope of this paper. Our scheme cannot identify adversaries if they stay in a close proximity with valid users and have verified social profiles. In most privacy-preserving schemes of location-based services, they have not considered this type of attack due to the fact that malicious behaviors cannot be determined via the location update and service request processes. Also, the proposed scheme is vulnerable to denial-of-service attacks in which a malicious user keeps sending requests to the server without finishing searching and authentication with the desired users. Collusion attacks and global observer attacks are not considered either in this work.

IV. Proposed Scheme

A. Overview

In our system, each mobile user i is generally characterized by a set of social attributes $|\mathcal{S}_i| := \varsigma$, such as *age*, *occupation*, and *interests*, where ς denotes the cardinality of the attribute set (a.k.a. personal social profile). Users choose their attributes that they intend to get verified, so that they can leverage them to search for friends with the same attributes. CA issues certificate σ after it verifies the authenticity of a user's claimed attributes, where σ corresponds to users' attributes. Then, CA may go off-line. We continue to use Alice's friend search as an example to illustrate the scheme. As shown in Fig. 2, Alice stays in the coverage of base station BS and wants to find a potential friend who appears in a close proximity and has the same attribute, say, *travel*. First, she checks in her location and herself by sending the request together with the proof of σ to BS. If BS successfully verifies the authenticity of σ, it sends to Alice a public/private key pair pk/sk corresponding to the location (loc) and time (τ). Meanwhile, BS applies KP-ABE [14] scheme to encrypt Alice's ID id_A using pk as $C := E_{pk(loc,\tau)}(id_A)$, then it creates a searchable table in the geosocial networking server with a token ω_s based on Alice's verified attribute *travel*. Suppose there are more than enough mobile users with the same attribute certificate σ checked in the same geographic area. Then, Alice can use the token ω_s to search the table and obtain others' ID using valid $sk(loc, \tau)$.

B. System Setup

1) Security Setup: This process helps initialize and define the security level of the system. We first give a brief definition of bilinear map. Let G_1, G_2 and G_T be three cyclic groups of prime order p. A bilinear map e is a map $e : G_1 \times G_2 \rightarrow G_T$ which has the properties of *bilinearity*, *computability*, and *non-degeneracy* [15], [16]. Given the security parameter ξ, CA first generates a parameter tuple $(p, G_1, G_2, G_T, e) \leftarrow 1^{\xi}$. Then, CA randomly selects two generators $g_1 \in G_1$, $g_2 \in G_2$.

2) Certificate Issuance: In order to guarantee the verifiability of users' attributes, we let CA issue the corresponding certificates after it checks the authenticity of the attributes. Here, we use Boneh-Boyen signature scheme [17] to sign each attribute value, which is secure under weak chosen message attacks based on the $q-\mathsf{SDH}$ assumption. CA uses $\chi_\imath, z_\imath \in Z_p^*$ to sign different values of one specific attribute in an attribute classification. We can rewrite each user's attribute as $m_{\imath.\jmath} \in Z_p$, where $\imath \in |\mathcal{I}|$ is the general classification of attributes and $\jmath$ denotes the specific value of the attribute $\imath$. Note that both $\imath$ and $\jmath$ are the indices in its own set. Given a verified attribute $m_{\imath.\jmath}$, TA outputs a certificate as $\sigma^i_{\imath.\jmath} = g_1^{1/(\chi_\imath + m_{\imath.\jmath} + z_\imath \delta^i_\imath)}$, public parameters $\mathcal{K}_\imath = g_2^{\chi_\imath}$, $\kappa_\imath = g_2^{z_\imath}$, and gives them to valid users, where $\delta^i_\imath \in Z_p \backslash \{-\frac{\chi_\imath + m_{\imath.\jmath}}{z_\imath}\}$ is a random number chosen by CA for each attribute of mobile user i. In what follows, we use $m^i_{\imath.\jmath}$ to denote user i's attribute value $\jmath$ on attribute classification $\imath$.

3) Key Distribution: In our scheme, each localization infrastructure (we use base station BS in what follows) generates and distributes a set of public/private key pairs corresponding to users' current spatial and temporal information. Note that for simplicity concerns, we do not consider multi-authority scenario in distributing key pairs, in the sense that mobile users only obtain key pairs from the base station in which they currently stay. We apply *key-policy attribute-based encryption* (KP-ABE) as our basic cryptographic primitive to help encrypt users' ids and store on the untrusted server, which allows mobile users who have private keys corresponding to the same spatial and temporal information to decrypt. The key distribution is run as follows,

Setup: BS first chooses spatial and temporal information as a set $\mathcal{P} := (x_1, x_2, ..., x_n)$, where each element of $x_i \in Z_p^*$ represents geographic coordinate and temporal information. Here, each base station chooses the granularity of their update and report policy by setting different cardinality of their set $\mathcal{P}$. Then, it chooses $t_1, t_2, ..., t_{n+1} \in Z_p$ and generates $\mathcal{G}_1 = g_1^{t_1}, \mathcal{G}_2 = g_1^{t_2}, ..., \mathcal{G}_{n+1} = g_1^{t_{n+1}}$ and $\mathcal{H}_1 = g_2^{t_1}, \mathcal{H}_2 = g_2^{t_2}, ..., \mathcal{H}_{n+1} = g_2^{t_{n+1}}$. Redefine a function $T_0(x_i) = g_2^{t_i}$, where $1 \leqslant i \leqslant n$ corresponds to each element in $\mathcal{P}$, and also define the Lagrange coefficient $L_{i,\mathcal{P}} = \prod_{j \in \mathcal{P}, j \neq i} \frac{x-j}{i-j}$ is for i and $\mathcal{P}$. Then, it randomly chooses $y \in Z_p$ and publishes $\{g_0 = g_1^y, n, \{\mathcal{G}_i, \mathcal{H}_i\}_{i=1}^{n+1}\}$ to all mobile users in its coverage and keeps $\{y, t_1, t_2, ..., t_{n+1}\}$ as the master secret key.

Key Generation: For each mobile user within the coverage of BS, it defines a $d-1$ degree polynomial q such that $q(0) = y$. Then, we define a $sk := \{g_1^{q_i(0)/t_i}\}_{i=1}^{n+1}$ used to encrypt mobile users' ids and store on the untrusted server, where $q_i(x)$ is a random polynomial for each node on the access structure.

Key Distribution and Maintenance: For each time period τ, BS renders new check-in mobile users the sk via secure channel, such as SSL. When the pre-set refresh time expires, BS re-encrypts the ciphertext on the untrusted server and updates the private key as sk' to non-check-out mobile users.

C. Privacy-preserving Friend Search

In this subsection, we elaborate our privacy-preserving friend search scheme in detail, which consists of three major building block: check-in verification, dynamic attribute-based searchable encryption, and searchable encryption maintenance.

1) Check-in Verification: Assuming there are $\mathbb{N}$ mobile users reach a particular location covered by BS, they first need to check in at this location. However, to avoid the attacks on invalid attributes, we ask BS to verify the authenticity of both identity and attribute values of the corresponding user. By sending the commitment and proof of $\{m||\sigma||\mathcal{K}||\kappa\}$ to BS, it can check whether the following equation is satisfied without directly revealing the above private parameters:

$$e(\sigma_{\imath.\jmath}, \mathcal{K}_\imath \cdot \kappa_\imath \cdot g_2^{m_{\imath.\jmath}}) = e(g_1, g_2),^2 \qquad (1)$$

If the verification process succeeds, BS issues the corresponding pk/sk to check-in users. Otherwise, BS rejects the check-in requests. Since the attribute values are pre-verified by CA, we assume the identity is also verified along with the successful verification on mobile users' attributes. For more privacy preservation protocols, we refer [18] for more detail on using non-interactive zero-knowledge proof for verification.

[2] The operation $(\cdot)$ is multiplication operation on the corresponding groups, e.g., G_1, G_2, and G_T.

2) Dynamic Attributed-based Searchable Encryption: In what follows, we apply part of the dynamic searchable symmetric scheme proposed in [19] and extend it to develop our searchable public-key encryption scheme. First, we define a set of pseudo-random functions as: $\mathfrak{F} : \{0,1\}^{\xi} \times \{0,1\}^* \rightarrow \{0,1\}^{\xi}$, $\mathfrak{G} : \{0,1\}^{\xi} \times \{0,1\}^* \rightarrow \{0,1\}^*$, and $\mathfrak{P} : \{0,1\}^{\xi} \times \{0,1\}^* \rightarrow \{0,1\}^{\xi}$. Let $H_1 : \{0,1\}^* \rightarrow \{0,1\}^*$ and $H_2 : \{0,1\}^* \rightarrow \{0,1\}^*$ be two cryptographic hash functions. Our dynamic attribute-based searchable encryption consists of the following main subprotocols: $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Search}, \mathsf{Dec})$ and complementary subprotocols to maintain the searchable tables and data: $(\mathsf{AddToken}, \mathsf{DelToken}, \mathsf{Add}, \mathsf{Del}, \mathsf{ReEnc})$, which will be discussed later. The main construction is designed as follows,

$\mathsf{Gen}(1^{\xi})$: Given the input of security parameters ξ in the previous system setup phase, each base station BS outputs three $|\xi|$-bit strings as keys: $K = (K_1, K_2, K_3)$.

$\mathsf{Enc}(K, id)$: Suppose there are $\mathbb{N}$ users checked in at BS. To encrypt their identities $(id_1, id_2, ..., id_{\mathbb{N}} \in G_T)$,

1. Let BS create and stroe a searchable array A_s, a deletion array A_d, a searchable table T_s, and a deletion table T_d on the untrusted server, where $|\mathsf{T}_s| = |\mathcal{I}|$ and $|\mathsf{T}_d| = \mathbb{N}$.
2. For each attribute value $m_{\imath.\jmath} \in \mathcal{I}$,

a) BS creates a list $\mathsf{L}_{m_{\imath.\jmath}}$ of mobile users who have been successfully verified with the certification $\sigma_{\imath.\jmath}$. The list consists of nodes $(\mathsf{N}_1, .., \mathsf{N}_{\eta})$ and is stored at random location in A_s, where η is the number of mobile users who have the certificate $\sigma_{\imath.\jmath}$. Define N_i as:

$$\mathsf{N}_i := (\langle i, \mathsf{addr}_s(\mathsf{N}_{i+1})\rangle \oplus H_1(\mathfrak{P}_{K_3}(m_{\imath.\jmath}), r_i), r_i)$$

where $r_i \in Z_p$ is a random number and addr is the address of the corresponding arrays and tables.

b) Set a pointer to the first node of $\mathsf{L}_{m_{\imath.\jmath}}$ in T_s as

$$\mathsf{T}_s[\mathfrak{F}_{K_1}(m_{\imath.\jmath})] := \langle \mathsf{addr}_s(\mathsf{N}_1), \mathsf{addr}_d(\mathsf{N}_1^*)\rangle \oplus \mathfrak{G}_{K_2}(m_{\imath.\jmath}),$$

where N_1^* denotes the dual node of N_1, and it is stored in A_d (refer Fig. 3).

3. Since each id may contain multiple attribute values, BS creates A_d and T_d to better maintain the searchable table. For each identity id_i,

a) BS creates a list L_{id} of dual nodes $(\mathsf{D}_1, ..., \mathsf{D}_{\zeta})$ and stores it at random locations in A_d, where D_i is associated with a node in $\mathsf{L}_{m_{\imath.\jmath}}$ and ζ is the total number of verified attributes of the mobile user id. Define D_i as,

$$\begin{aligned} \mathsf{D}_i \;\; := \;\; & (\langle \mathsf{addr}_d(\mathsf{D}_{i+1}), \mathsf{addr}_d(\mathsf{N}_{-1}^*), \mathsf{addr}_d(\mathsf{N}_{+1}^*), \\ & \mathsf{addr}_s(\mathsf{N}), \mathsf{addr}_s(\mathsf{N}_{-1}), \mathsf{addr}_s(\mathsf{N}_{+1}), \mathfrak{F}_{K_1}(m_{\imath.\jmath})\rangle \\ & \oplus H_2(\mathfrak{P}_{K_3}(id), r_i'), r_i'). \end{aligned}$$

where N_{-1} and N_{+1} denote the preceding and succeeding nodes of N, respectively.

b) Similarly, BS stores a pointer to the first node in L_{id} in T_d as

$$\mathsf{T}_d[\mathfrak{F}_{K_1}(id)] := \mathsf{addr}_d(\mathsf{D}_1) \oplus \mathfrak{G}_{K_2}(id),$$

4. For each id checked in with BS, BS applies the key-private KP-ABE scheme [14], [20] in an asymmetric setting to encrypt their identities as follows. For the current temporal and geographic information in $\mathcal{P}$, BS chooses a random number $\lambda \in Z_p$, and encrypts all valid mobile users with $id \in G_T$ as $C := (\gamma_{\mathcal{P}}, E' = id \cdot e(g_0, g_2)^{\lambda}, \{E_i = T_0(x_i)^{\lambda}\}_{x_i \in \gamma_{\mathcal{P}}})$, where $\gamma_{\mathcal{P}}$ represents the access structure. Then, BS stores $\{C_k\}_{k=1}^{\mathbb{N}}$ in the untrusted server together with searchable arrays and tables $\mathcal{J} := \{\mathsf{A}_s, \mathsf{A}_d, \mathsf{T}_s, \mathsf{T}_d\}$ as shown in Fig.3.

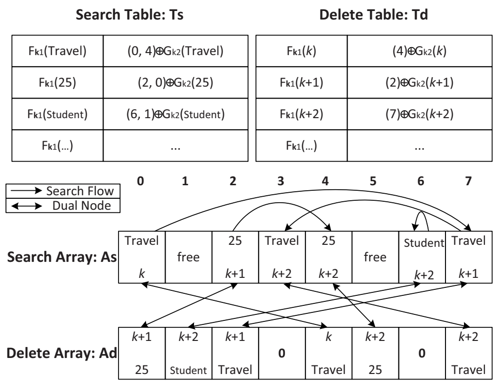

Search Table: Ts
Fk1(Travel) | (0, 4)⊕Gk2(Travel)
Fk1(25) | (2, 0)⊕Gk2(25)
Fk1(Student) | (6, 1)⊕Gk2(Student)
Fk1(...) | ...
Delete Table: Td
Fk1(k) | (4)⊕Gk2(k)
Fk1(k+1) | (2)⊕Gk2(k+1)
Fk1(k+2) | (7)⊕Gk2(k+2)
Fk1(...) | ...
Search Flow
Dual Node
0 1 2 3 4 5 6 7
Search Array: As
Travel k | free | 25 k+1 | Travel k+2 | 25 k+2 | free | Student k+2 | Travel k+1
Delete Array: Ad
k+1 25 | k+2 Student | k+1 Travel | 0 | k Travel | k+2 25 | 0 | k+2 Travel

Fig. 3. Dynamic Searchable Encryption

We continue to use Alice's friend search as an example to elaborate the subprotocols. Considering Alice who is searching for the attribute value $m_{\imath.\jmath} = \textit{Travel}$, BS encrypts and stores the mobile users with *Travel* on the untrusted server as $(0, 4) \oplus \mathfrak{G}_{K_2}(\textit{Travel})$ in T_s and the corresponding entries in T_d.

$\mathsf{Search}(\mathcal{J}, K, m_{\imath.\jmath}, \{C_k\}_{k=1}^{\mathbb{N}})$: Let mobile users use their attributes to search for other check-in users who have the similar social profile.

1. Alice obtains the search token $\omega_s = (\omega_1, \omega_2, \omega_3) := (\mathfrak{F}_{K_1}(m_{\imath.\jmath}), \mathfrak{G}_{K_2}(m_{\imath.\jmath}), \mathfrak{P}_{K_3}(m_{\imath.\jmath}))$ from BS based on her own attribute value $m_{\imath.\jmath}$. Then, she sends ω_s to the untrusted server.
2. If $\mathsf{T}_s[\omega_1] = \perp$, abort the search protocol; otherwise, recover the pointer by computing $\mathsf{addr}_s[\mathsf{N}_1] := \mathsf{T}_s[\omega_1] \oplus \omega_2$.
3. Look up $(\psi, r_1) := \mathsf{A}_s[\mathsf{addr}_s[\mathsf{N}_1]])$ and output the index of $\{C_k\}_{k=1}^{\mathbb{N}}$ by decrypting N_1 as $(id_i, \mathsf{addr}_s(\mathsf{N}_2)) := \psi \oplus H_1(\omega_3, r_1)$.
4. Continue to look up N_2 until all entries and indices are found. The protocol outputs $\{C_k\}_{k=1}^{\mathbb{N}_{\imath.\jmath}}$ which are the ciphertexts of mobile users' ids with the attribute value $m_{\imath.\jmath}$.

As shown in Fig. 3, the search process first outputs the first address as $\mathsf{addr}_s[\mathsf{N}_1] := \mathsf{A}_s[0]$. Then, it continues to search $\mathsf{A}_s[7]$ and $\mathsf{A}_s[3]$, until Alice obtains the indices of the ciphertext as $\{k, k+1, k+2\}$.

$\mathsf{Dec}(sk, \{C_k\}_{k=1}^{\mathbb{N}_{\imath.\jmath}})$: Alice derives the plaintext id using the private key $sk := \{\tilde{d}_i : g_1^{q_i(0)/t_i}\}_{i=1}^{n+1}$,

$$\begin{aligned} id \;\; &= \;\; \frac{E'}{\prod_{i \in \gamma_{\mathcal{P}}} e(\tilde{d}_i, E_i)^{L_{i,\gamma_{\mathcal{P}}}(0)}} \\ &= \;\; id \cdot e(g_0, g_2)^{\lambda} \prod_{i \in \gamma_{\mathcal{P}}} \frac{1}{e\left(g_1^{\frac{q_i(0)}{t_i}}, g_2^{\lambda t_i}\right)^{L_{i,\gamma_{\mathcal{P}}}(0)}} \\ &= \;\; id \cdot e(g_1, g_2)^{\lambda y} \prod_{i \in \gamma_{\mathcal{P}}} \frac{1}{e(g_1, g_2)^{\lambda q_i(0) L_{i,\gamma_{\mathcal{P}}}(0)}} \\ &= \;\; id \cdot e(g_1, g_2)^{\lambda y} \cdot \frac{1}{e(g_1, g_2)^{\lambda y}} = id, \end{aligned}$$

the above process aggregates each pairing results on each leaf node of the access structure in the bottom-up manner using polynomial interpolation. Finally, if Alice's spatial and temporal set $\gamma_{\mathcal{P}}$ corresponds to the access structure, she is able to recover other mobile users checked in at this location.

3) Searchable Encryption Maintenance: Based on the analysis of Foursquare dataset, we notice that the service provider will record users' trace files in their storage, which obviously compromises users' location privacy. In what follows, we discuss the maintenance of the searchable arrays and tables, which includes *dynamic user update*, *key update*, and *token update*. *Dynamic user update* consists of user addition, deletion and attribute addition, deletion, while *key update* and *token update* mainly focus on updating users' current private keys and the revocation of expired mobile users. Due to page limits, we will discuss the *key update* and *token update* in our full version paper.

Dynamic User Update: Based on the characteristic of current geosocial networking applications, mobile users frequently check in and out at their interested locations, in which the update on the untrusted server should be dynamic and efficient. Hence, we come up with the following subprotocols to fulfill our design objectives. First, we consider the most complicated scenario where we need to add Bob to the current tables with attribute $m_{\imath.\jmath}$,

$\mathsf{AddToken}(K, id')$: Given a new user id', BS creates the corresponding ciphertext and searchable token used to update $\mathcal{J}$.

1. BS computes P based on id' as:

$$\begin{aligned}\mathsf{P} \quad := \quad & (\mathfrak{F}_{K_1}(m_{\imath.\jmath}), \mathfrak{G}_{K_2}(m_{\imath.\jmath}), \langle k', \mathbf{0}\rangle \oplus H_1(\mathfrak{P}_{K_3}(m_{\imath.\jmath}), r), \\ & r, \langle \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathfrak{F}_{K_1}(m_{\imath.\jmath})\rangle \oplus H_2(\mathfrak{P}_{K_3}(id'), r'), r')\end{aligned}$$

   where r and r' are $|\xi|$-bit random strings and $\mathbf{0}$ is a $\log|\mathsf{A}_s|$-bit length string of 0. We use $\mathsf{P}[j], 1 \leqslant j \leqslant 6$ to denote the elements in P.
2. Then, BS recalls KP-ABE scheme in Enc to encrypt the new id' with current $\mathcal{P}$ and obtains $C_{k'}$.
3. BS creates token $\omega_a := (\mathfrak{F}_{K_1}(id'), \mathfrak{G}_{K_2}(id'), \mathsf{P})$ and stores $C_{k'}$ to the untrusted server.

$\mathsf{Add}(\mathcal{J}, \{C_k\}_{k=1}^{\mathbb{N}}, \omega_a)$: BS updates $\mathcal{J}$ to enable further search.

1. Search empty location in T_s and outputs ϕ, then compute $(\phi_{-1}, \phi^*) := \mathsf{A}_s[\phi]$. Note the list of free entries is also constructed as $\mathsf{L}_{m_{\imath.\jmath}}$ and L_{id}.
2. Update the search table to point ϕ_{-1} as the start pointer of empty entry. Then, recover the first node N_1 by computing $\langle \mathsf{addr}_s(\mathsf{N}_1), \mathsf{addr}_d(\mathsf{N}_1^*)\rangle := \mathsf{T}_s[\mathsf{P}[1]] \oplus \mathsf{P}[2]$.
3. Store a new node at location ϕ and modify the forward pointer as $\mathsf{A}_s[\phi] := (\mathsf{P}[3] \oplus \langle \mathbf{0}, \mathsf{addr}_s(\mathsf{N}_1)\rangle, \mathsf{P}[4])$. Update the attribute $m_{\imath.\jmath}$ in T_s to allow users to search as $\mathsf{T}_s[\mathsf{P}[1]] := (\phi, \phi^*) \oplus \mathsf{P}[2]$.
4. Compute $(\mathsf{D}_1, r) := \mathsf{A}_d[\mathsf{addr}_d(\mathsf{N}_1^*)]$, then update N_1^* as $\mathsf{A}_d[\mathsf{addr}_d(\mathsf{N}_1^*)] := (\mathsf{D}_1 \oplus \langle \mathbf{0}, \phi^*, \mathbf{0}, \mathbf{0}, \phi, \mathbf{0}, \mathbf{0}\rangle, r)$. Also update the new first node in $\mathsf{A}_d[\phi^*] := \mathsf{P}[5] \oplus \langle \phi_{-1}^*, \mathbf{0}, \mathsf{addr}_d(\mathsf{N}_1^*), \phi, \mathbf{0}, \mathsf{addr}_s(\mathsf{N}_1), \mathsf{P}[1]\rangle, \mathsf{P}[6])$, where ϕ_{-1}^* is the address that points to the next available free entry which may be used to store Bob's another possible attribute value m'.
5. Update deletion table T_d by setting $\mathsf{T}_d[\omega_1] := \langle \phi^*, \mathbf{0}\rangle \oplus \omega_2$.

In Fig. 4, BS updates $\mathsf{A}_s[1]$ as the first entry of mobile users with attribute value *Travel* and stores new node N_1 in A_s and the corresponding dual node N_1^* in $\mathsf{A}_d[3]$. By updating $\mathcal{J}$, Alice is able to search for mobile users who have the attribute $m_{\imath.\jmath}$ in a sequential manner with the beginning user $id' = $ Bob. Therefore, we can update new users and their new attribute values using the above approach.

Second, another regular update is user check-out, where mobile users leave the current location or unsubscribe the geosocial networking functionality. Here, we consider to remove $id'' =$ Carol who has checked in using attribute value $m_{\imath.\jmath}$.

$\mathsf{DelToken}(K, id'')$: Given the index k'' (e.g., $k+2$ in A_d in the Fig. 4) corresponding to id'' that BS wants to delete, it outputs $\omega_d := (\mathfrak{F}_{K_1}(id''), \mathfrak{G}_{K_2}(id''), \mathfrak{P}_{K_3}(id''), id'')$.

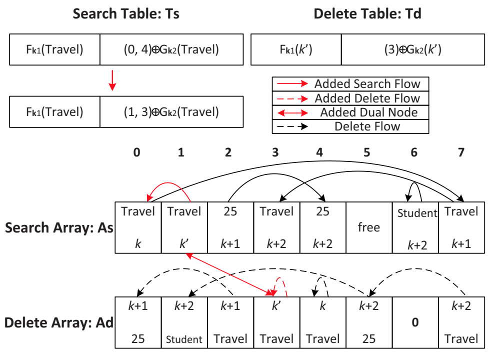


Fig. 4. Dynamic User Update Process

$\mathsf{Del}(\mathcal{J}, \{C_k\}_{k=1}^{\mathbb{N}}, \omega_d)$: BS deletes user id'' and remains other mobile users stored on the untrusted storage for further search.

1. Compute $\mathsf{addr}_d(\mathsf{D}_1) := \mathsf{T}_d[\mathfrak{F}_{K_1}(id'')] \oplus \mathfrak{G}_{K_2}(id'')$. Then, decrypt D_1 by computing $(\mathsf{D}_1, r) := \mathsf{A}_d[\mathsf{addr}_d(\mathsf{D}_1)]$ and $(\mathsf{Q}[1], ..., \mathsf{Q}[6], \mathfrak{F}_{K_1}(id'')) := \mathsf{D}_1 \oplus H_2(\mathfrak{P}_{K_3}(id''), r)$.
2. Delete D_1 and fill $\mathsf{A}_d[\mathsf{addr}_d(\mathsf{D}_1)]$ with $(6\log|\mathsf{A}_d| + |\xi|)$-bit random strings. Update the last free entry in T_s as $\mathsf{Q}[4]$, and keep index k'' as a record. Then, set $\mathsf{A}_s[\mathsf{Q}[4]] := (\phi', \mathsf{addr}_d(\mathsf{D}_1))$, where ϕ' is the original first node of free entry in A_s.
3. First, to update the node in A_s, compute $(\mathsf{R}[1], \mathsf{R}[2]) := \mathsf{A}_s[\mathsf{Q}[5]] \oplus H_1(\mathfrak{P}_{K_3}(m_{\imath.\jmath}), r_{-1})$. Then, replace $\mathsf{R}[2]$ as $\mathsf{R}[2] \oplus \mathsf{Q}[4] \oplus \mathsf{Q}[6]$, where the address of N_{i+1} of N_{-1} is updated to N_{i+1}.
4. Use the similar approach to update the pointer of N_{-1} and N_{+1} in $\mathsf{A}_d[\mathsf{Q}[2]]$ and $\mathsf{A}_d[\mathsf{Q}[3]]$.
5. Update the current node address as $\mathsf{Q}[1]$ and continue to update other attribute values of id''. Remove $C_{i''}$ from $\{C_k\}_{k=1}^{\mathbb{N}}$, and remove $\mathfrak{F}_{K_1}(id'')$ from T_d.

We show the deletion process in Fig. 4, where the untrusted server deletes a user with index $k+2$ by searching T_d to find the first node stored in $\mathsf{A}_d[7]$. Then, it can sequentially search the following nodes and update the corresponding dual nodes stored in $\mathsf{A}_s[3]$, $\mathsf{A}_s[4]$, and $\mathsf{A}_s[6]$. Finally, the dynamic user update process helps maintain the searchable arrays and tables stored on the untrusted server without causing extra updates on unchanged mobile users.

V. Protocol Evaluation

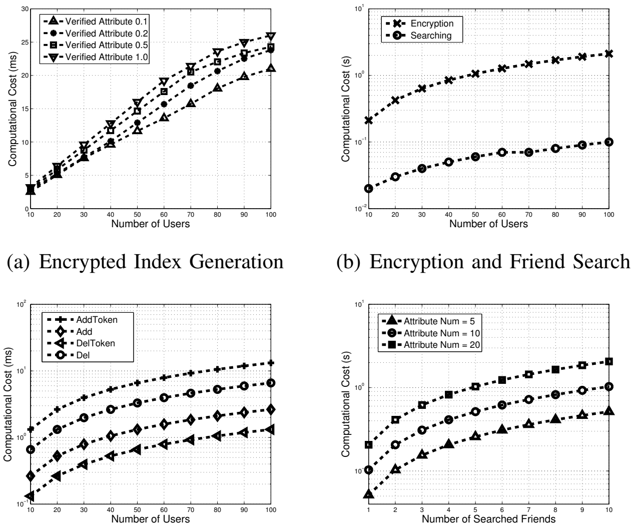
A. Security and Privacy Analysis

We discuss the privacy-preserving mechanism together with the possible attacks listed in the adversary model. First, for the active attacker, the most possible attack would be using fabricate spatial and temporal information to have friend search as good mobile users. However, in our scheme, the location infrastructure issues private keys based on the current spatial and temporal information, which prohibits malicious users from knowing the search tokens. Also, instead of directly issuing the keys K, the location infrastructure issues $\omega_s := (\mathfrak{F}_{K_1}(m_{\imath.\jmath}), \mathfrak{G}_{K_2}(m_{\imath.\jmath}), \mathfrak{P}_{K_3}(m_{\imath.\jmath}))$ to mobile users after it checks the validity of their attribute values.

In terms of the encryption scheme, we modify the existing scheme [14] through the use of the asymmetric DDH-hard groups. The use of DDH-hard pairing groups requires the symmetric external Diffie-Hellman (SXDH) assumption and bilinear Diffie-Hellman (BDH) assumption. By applying these changes, we achieve the same security level (CCA-Security) and better efficiency performance. For mobile users who have never appeared at a particular location, they fail to obtain check-in users' ids even given the ciphertexts. Also, for the same reason, the untrusted server would not be able to identify the current users due to the lack of the location and temporal keys sk.

B. Efficiency Analysis

In this subsection, we mainly discuss the efficiency of our proposed scheme in terms of computational cost of mobile users, location infrastructure, and geosocial networking server. First, we use the Pairing-based Cryptography (0.5.12) Library to implement our simulation on computational cost. We take Tate pairing as our basic pairing operation. The elliptic curve we use for the our scheme is type D159. A curve of such type has the form of $y^2 = x^3 + ax + b$. The order of the curve is 159 bits, as is F_q, the base field. For the experiments, we use a laptop with an Intel processor 2.8GHz and 2GB RAM under the platform Ubuntu 11.10. All the timing reported below are averaged over 100 randomized runs. In terms of attributes, we apply the Foursquare dataset in our scheme. There are more than 100 different categories to characterize venues in a city. Hence, for the simulation settings, each mobile user has $|\mathcal{S}| = 100$ attributes but she/he may choose part of their attributes to get verified. Also, we observer that the average number of check-ins is 212.6 in Los Angles, and 95.1 in New York City. The numbers of checked users are 99.5 and 39.4 in two cities, respectively. Considering the maximum checked users, we take 100 as the total number of users check in at one particular BS as the simulation settings.



(a) Encrypted Index Generation

(b) Encryption and Friend Search

(c) User Update

(d) Mobile Users' Decryption

Fig. 5. Computational Cost in Dynamic Attribute-based Friend Search

During the dynamic attribute-based friend searching, we mainly consider the computational cost of the base station that creates and maintains the searchable arrays and tables on the geosocial networking server. Meanwhile, we also discuss the computation load of the search process along with the encryption and decryption schemes for both the base station and mobile users. As shown in Fig. 5(a), the computational cost increases when the number of check-in mobile users grows. Also, mobile users may choose different numbers of verified attributes to enable friend search. According to the simulation results, if users choose to upload all their attributes, say, 100 attributes, the computational cost of generating searchable arrays and tables is around 25ms for the base station. In Fig. 5(b), we consider the size of $\mathcal{P}$ is 5 to represent the current spatial and temporal information. To search mobile users with corresponding attributes, it consumes negligible time for the untrusted server compared to the KP-ABE scheme performed by the base station. For the maintenance of mobile users, we can see from Fig. 5(c) that to add a new mobile user with one attribute, the computational cost of 100 mobile users is around 12ms for AddToken, which is the most time consuming part. Since most of computation burden in Add and Del is done on the untrusted server, we can relieve it for both mobile users and base stations. Fig. 5(d) shows the decryption cost for mobile users on different set size of $\mathcal{P}$. BS can choose its own granularity on the spatial and temporal information. It is obvious that the more granularity of the encryption attribute set, the more computational cost for mobile users to decrypt the searched friends.

VI. Conclusion

In this paper, we propose a user-centric privacy-preserving friend search in geosocial networks without relying on trusted servers. The scheme not only guarantees the location privacy for mobile users, but also enables them to search for friends with similar attributes on an untrusted geosocial networking server. Based on the trace-driven protocol evaluation, we show the security and efficiency of the proposed scheme.

References

[1] [Online]. Available: http://foursquare.com/about/

[2] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46 – 55, jan-mar 2003.

[3] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. of ICDCS'05*, 2005, pp. 620–629.

[4] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, ser. VLDB '06. VLDB Endowment, 2006, pp. 763–774.

[5] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-aware multiple mix zone placement for protecting location privacy," in *INFOCOM'12*. IEEE, 2012.

[6] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[7] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of MobiSys'03*, 2003, pp. 31–42.

[8] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proceedings of the 16th ACM conference on Computer and communications security*, ser. CCS '09, 2009, pp. 348–357.

[9] T. Wang and L. Liu, "Privacy-aware mobile services over road networks," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 1042–1053, 2009.

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

[11] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[12] W.-j. Hsu, D. Dutta, and A. Helmy, "Profile-cast: behavior-aware mobile networking," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 1, pp. 52–54, Jan. 2008.

[13] J. Bao, Y. Zheng, and M. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *International Conference on Advances in Geographic Information Systems*, ser. ACM SIGSPATIAL, Redondo Beach, California, 2012.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06, 2006, pp. 89–98.

[15] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *Advances in Cryptology —CRYPTO 2001*, pp. 213–229, 2001.

[16] E.-J. Goh, "Encryption Schemes from Bilinear Maps," Ph.D. dissertation, Department of Computer Science, Stanford University, Sep 2007.

[17] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," *CRYPTO'04, LNCS*, pp. 41–55, 2004.

[18] L. Guo, C. Zhang, J. Sun, and Y. Fang, "PAAS: Privacy-preserving attribute-based authentication system for ehealth networks," in *ICDCS 2012*. IEEE, 2012.

[19] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proceedings of the 19th ACM conference on Computer and communications security*, ser. CCS '12, 2012.

[20] G. Ateniese, J. Kirsch, and M. Blanton, "Secret handshakes with dynamic and fuzzy matching," in *NDSS*, 2007.