A Survey on Privacy Preserving Location Transfer for mobile users

Safiya M K
Department of Computer Science and Engineering
MES College of engineering
Kuttippuram, India
Safiyasalam3112@gmail.com

Anil K Jacob
Department Of Computer Science and Engineering
MES College of engineering
Kuttippuram, India
akanjama@hotmail.com

Abstract

Location-based services are quickly becoming immensely popular. In addition to services based on users current location, many potential services rely on users location history, or their spatial-temporal provenance. Literature survey deals with study of three method dealing with privacy preserving location transfer for mobile users; they are VeriPlace is a location proof architecture which is designed with privacy protection and collusion resilience Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme. STAMP is designed for ad-hoc mobile users generating location proofs for each other in a distributed setting. However, it can easily accommodate trusted mobile users and wireless access points. A Privacy-Preserving Location proof Updating System (APPLAUS) in which co-located Blue tooth enabled mobile devices mutually generate location proofs and send updates to a location proof server. The method using for privacy preservation in above method are achieved through encryption with suitable protocol.

Keywords- Location proof, privacy, spatial temporal provenance, trust.

1. Introduction

Location-based services are quickly becoming immensely popular. In addition to services based on users current location, many potential services rely on users location history. Location-based services have seen lot of interest recently, both from companies and from academia. For instance, Four-square provides a location-based social networking platform through which users can "check in" at places that they are visiting and learn their friends' location. Users can benefit from being at a given location. For example, Four-square has the user who checks in the most often at a location become the mayor of the location. In turn, the owner of the location might provide a reward to this user. If there is a benefit for being at a location, people get an incentive for claiming that they are at a location even though they are not. Location proofs are an alternative approach to let location based services verify a user's location. Formally, a location proof is an electronic document that certifies someone's presence at a certain location at some point in time. A location proof architecture is a mechanism with which users obtain location proofs and with which services validate proofs. A crucial challenge is to ensure that users' privacy does not get violated while collecting and using location proofs. These proofs are spatial-temporal provenance (STP) of the user, and a digital proof of user's presence at a location at a particular time as an STP proof. The location based service providing scheme has following challenges;

A. Security: The security of STP proofs are two fold: integrity and non-transferability. The integrity property requires that no prover can create fake STP proofs by himself/herself or by collaborating with one or more other untrusted parties in the system. The non-transferability property requires that no prover can claim the ownership of another prover's legitimate STP proofs.

B. Privacy: Location privacy is an extremely important factor that needs to be taken into consideration when designing any location based systems. Revealing both identity and location information to an untrusted party poses threats to a mobile users, the prover, either. Pseudonyms:
Pseudonyms are often used to provide anonymity. Nevertheless, if the same pseudonym is used by a mobile user, it is possible for an adversary to track the user's location. Therefore, we need to develop mechanisms to secure location transfer.

**A. VeriPlace**: A location proof architecture enables users to collect proofs for being at a location and services to validate these proofs. It is essential that this proof collection and validation does not violate user privacy. We introduce VeriPlace, a location proof architecture with user privacy as a key design component. In addition, VeriPlace can detect cheating users who collect proofs for places where they are not located. We also present an implementation and a performance evaluation of VeriPlace and its integration with Yelp.

**B. APPLAUS**: A Privacy-Preserving Location proof Updating System (APPLAUS) in which collocated Bluetooth enabled mobile devices mutually generate location proofs and send updates to a location proof server. Periodically changed pseudonyms are used by the mobile devices to protect source location privacy from each other, and from the untrusted location proof server. We also develop user-centric location privacy model in which individual users evaluate their location privacy levels and decide whether and when to accept the location proof requests. In order to defend against colluding attacks, we also present betweenness ranking-based and correlation clustering-based approaches for outlier detection. APPLAUS can be implemented with existing network infrastructure, and can be easily deployed in Bluetooth enabled mobile devices with little computation or power cost.

**C. STAMP**: Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme. STAMP is designed for ad-hoc mobile users generating location proofs for each other in a distributed setting. It can easily accommodate trusted mobile users and wireless access points. STAMP ensures the integrity and non-transferability of the location proofs and protects users privacy. A semi-trusted Certification Authority is used to distribute cryptographic keys as well as guard users against collusion by a light-weight entropy-based trust evaluation approach.

**veriplace**

A location proof architecture enables users to collect proofs for being at a location and services to validate these proofs. It is essential that this proof collection and validation does not violate user privacy. We introduce VeriPlace[3], a location proof architecture with user privacy as a key design component. In addition, VeriPlace can detect cheating users who collect proofs for places where they are not located. We also present an implementation and a performance evaluation of VeriPlace and its integration with Yelp. Location proofs are an alternative approach to let location based services verify a user's location. Formally, a location proof is an electronic document that certifies someone's presence at a certain location at some point in time. A location proof architecture is a mechanism with which users obtain location proofs and with which services validate proofs. A crucial challenge is to ensure that users' privacy does not get violated while collecting and using location proofs.

- Following are the main design challenges of veriplace scheme:
  - Privacy: Privacy is of central importance to mobile users. Namely, we must prevent issuers and verifiers of location proofs from violating a user's privacy. Issuers of a location proof have knowledge of a user's location at the time the proof is issued to the user. If proof issuers can also learn the user's identity, they will be able to track users by their location. Therefore, shielding user identity from proof issuers is essential.
  - Security: Mobile users may lie about their location to use services that they are not qualified for. A challenge is to incorporate a cheating detection mechanism into the location proof architecture so as to (1) make cheating difficult for users, (2) enable services to spot dishonest users who submit incorrect location information nonetheless, and (3) detect cheating without compromising user privacy.
  - Flexibility: A location proof must satisfy the requirements of the service that it is handed to. Embedding service-specific data (e.g., the service's public key [29]) in a location proof reduces the applicability of the proof because users would have to request a proof for each service that they want to interact with. The situation worsens as the number of services increases. Therefore, a challenge is to produce general location proofs that are acceptable by various services.
Deployability: To be widely deployable, a location proof architecture must purely utilise the most common features of existing mobile devices or APs (assuming APs issue location proofs) and not require dedicated hardware. Requiring dedicated hardware is one of the factors that prevents previously proposed architectures from being widely deployed.

System Model
VeriPlace targets services provided by third parties (such as applications in Apple's App store), not services offered by cellphone providers (maybe in collaboration with third parties). VeriPlace requires three types of trusted entities that are run by different parties to avoid collusion. To protect users' privacy, each trusted entity knows either a user's identity or her location, but not both of them. Third parties are:

- A TTPL (Trusted Third Party for managing Location information) is responsible for issuing location proofs by creating a new proof that includes all the information contained in intermediate location proofs, with some information transformed to another representation.
- A TTPU (Trusted Third Party for managing User information) is in charge of storing encrypted location information associated with users. More specifically, a TTPU stores triples of the form (IDuser T; E), where T represents the time when user IDuser requested a proof and E represents the encrypted identity of the AP that issued the proof.
- A CDA (Cheating Detection Authority) carries out cheating detection. After receiving encrypted AP information from the TTPU, the CDA decrypts this information and checks whether any two APs are far apart, which is a sign of cheating (because the same user can not request location proofs at two far-apart places simultaneously). The CDA notifies the TTPU in case cheating is detected.

AppALUS
A Privacy-Preserving Location proof Updating System (AppALUS) [2] in which collocated Bluetooth enabled mobile devices mutually generate location proofs and send updates to a location proof server. Periodically changed pseudonyms are used by the mobile devices to protect source location privacy from each other, and from the untrusted location proof server. We also develop user-centric location privacy model in which individual users evaluate their location privacy levels and decide whether and when to accept the location proof requests. In order to defend against colluding attacks, we also present betweenness ranking-based and correlation clustering-based approaches for outlier detection.

System Preliminaries
- Pseudonym: As commonly used in many networks, we consider an online Certification Authority (CA) run by independent trusted third party which can pre-establish credentials for the mobile devices. Similar to many pseudonym approaches, to protect location privacy.
- Threat Mode: We assume that an adversary aims to track the location of a mobile node. An adversary can have the same credential as a mobile node and is equipped to eavesdrop communications. We assume that the adversary is internal, passive, and global. By internal, we mean that the adversary is able to compromise or control individual mobile device and then communicate with others to explore private information.
- Location Privacy Level: In this paper, we use multiple pseudonyms to preserve location privacy; i.e., mobile nodes periodically change the pseudonym used to sign messages, thus reducing their long term link ability. To avoid spatial correlation of their location, mobile nodes in proximity coordinate pseudonym changes by using silent mix zones or regions where the adversary has no coverage. Without loss of generality, we assume each node changes its pseudonyms from time to time according to its privacy.

System Model
AppALUS system contains the following entities:

- Prover: the node who needs to collect location proofs from its neighboring nodes. When a location proof is needed at time t, the prover will broadcast a location proof request to its neighboring nodes through Bluetooth.
- Witness: Once a neighboring node agrees to provide location proof for the prover, this node becomes a witness of the prover. The witness node will generate a location proof and send it back to the prover.
Location proof server: As our goal is not only to monitor real-time locations, but also to retrieve history location proof information when needed, a location proof server is necessary for storing the location history.

Certificate authority: As commonly used in many networks, we consider an online CA which is run by an independent trusted third party. Every mobile node registers with the CA and pre-loads a set of public/private key pairs before entering the network.

Verifier: a third-party user or an application who is authorized to verify a prover’s location within a specific time period. The verifier usually has close relationship with the prover, e.g., friends or colleagues, to be trusted enough to gain authorization.

**Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP)**

Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme [1]. STAMP is designed for ad-hoc mobile users generating location proofs for each other in a distributed setting. However, it can easily accommodate trusted mobile users and wireless access points. STAMP ensures the integrity and non-transferability of the location proofs and protects users' privacy. A semi-trusted Certification Authority is used to distribute cryptographic keys as well as guard users against collusion by a light-weight entropy-based trust evaluation approach. Most of the current location-based services for mobile devices are based on users’ current location. Users discover their locations and share them with a server. In turn, the server performs computation based on the location information and returns data/services to the users. In addition to users' current locations, there is an increased trend and incentive to prove/validate mobile users' past geographical locations. This opens a wide variety of new location-proof based mobile applications. Saroiu et al. described several such potential applications in. Let us consider three example: A store wants to offer discounts to frequent customers. Customers must be able to show evidence of their repeated visits in the past to the store.

STP proof scheme named Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP). STAMP aims at ensuring the integrity and non-transferability of the STP proofs, with the capability of protecting users' privacy. Most of the existing STP proof schemes rely on wireless infrastructure (e.g., WiFi APs) to create proofs for mobile users. However, it may not be feasible for all types of applications, e.g., STP proofs for the green commuting and battle field examples certainly cannot be obtained from wireless APs. To target a wider range of applications, STAMP is based on a distributed architecture. Co-located mobile devices mutually generate and endorse STP proofs for each other, while at the same time it does

wireless infrastructure may not be available everywhere and hence a system based on wireless APs creating STP proofs would not be feasible for all scenarios. In addition, the deployment cost would be high if we require a large number of wireless APs to have the capability of generating STP proofs. Therefore, we think a distributed STP proof architecture, i.e., mobile users obtaining STP proofs from nearby mobile peers, would be more feasible and appropriate for a wider range of applications. We design a generic decentralized protocol, and then show how it can work well for centralized case also. Above figure illustrates the architecture of our system. There are four types of entities based on their roles:

- **Prover**: A prover is a mobile device which tries to obtain STP proofs at a certain location.
- **Witness**: A witness is a device which is in proximity with the prover and is willing to create an STP proof for the prover upon receiving his/her request. The witness can be untrusted or trusted, and the trusted witness can be mobile or stationary (wireless APs). Co-located mobile users are untrusted.
- **Verifier**: A verifier is the party that the prover wants to show one or more STP proofs to and claim his/her presence at a location at a particular time.
Certificate Authority (CA): The CA is a semi-trusted server which issues, manages cryptographic credentials for the other parties. CA is also responsible for proof verification and trust evaluation.

Performance Analysis

A comparison of different schemes for privacy preserving location transfer are shown in following table. Different features and limitations of each schemes are compared. The comparison study is based on the design challenges, security, privacy, dealing with collusion attack and computational overhead. In terms of security, STAMP is preferable compared to other two approaches. Privacy is a major issue in VeriPlace method than APPLAUS and STAMP. Location transfer between applications require fair security, better privacy, should deal with collusion attack and also a low level of computational overhead. The performance analysis of the

<table>
<thead>
<tr>
<th>METHODS</th>
<th>Security</th>
<th>Privacy</th>
<th>Collision Attack</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeriPlace</td>
<td>Less</td>
<td>Medium</td>
<td>No</td>
<td>High</td>
</tr>
<tr>
<td>APPLAUS</td>
<td>Medium</td>
<td>Medium</td>
<td>Less</td>
<td>Less</td>
</tr>
<tr>
<td>STAMP</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>Less</td>
</tr>
</tbody>
</table>

Performance analysis

Location based services for mobile users are commonly available now. Those services are not only based on users current location, but also based on location history of user. There are three basic approaches for privacy preserving location transfer for mobile users. VeriPlace, APPLAUS, and STAMP. Each approach has a unique way to deal with privacy preserving location transfer. But in terms of security we can conclude that STAMP is more suitable scheme for privacy preserving location transfer. Also, STAMP deals with collusion attack and it is more efficient than the other two schemes.

Acknowledgments

1.1.1.1

Acknowledgment

I thank GOD almighty, who showered his abundant grace on me to make this journal a success. I express my heartfelt thanks and deep sense of gratitude to our Principal Dr. A. S. Varadarajan. I extend my deep gratitude to our Head of the Department Dr. Sasidaran Sreedharan, for his valuable help and support. I would also like to thank my seminar guide Mr. Dr. Anil K Jacob, Professor, Department of Computer Science and Engineering for his invaluable advice and whole hearted co-operation without whom I could not take this seminar successfully. I would also like to express my sincere gratitude to our seminar co-ordinator Dr. Govindaraj E. Assoc. Professor of Computer Science and Engineering Department for his guidance and support. Gracious gratitude to all faculty of the department for their valuable advice, encouragement and cooperation.

I express my sincere gratitude to all the staff members of Computer Science and Engineering Department and my beloved family members who helped me with their timely suggestions and support. I also express my sincere thanks to all my friends who helped me throughout the successful completion of this work.

References


First Author Presently pursuing Post Graduation in Computer Science and Technology, completed Batchlor Degree in Computer Science and Technology in 2015 from the MG University.

Second Author Working as Associate professor in Computer Science and Technology Department, MES College of Engineering, Kuttippuram, India.