Location Proof Updating System Using Bluetooth Devices

S.Adhikesavan[1] S.Ammathul Fareedha[2]
[1,2]Assistant Professor
[1,2]Information Technology
[1,2]Ganadipathy Tulsi's Jain Engg. College, Vellore, India.

Abstract— Many Location Based services rely on user's mobile device to determine the current location. This allows attackers to access a restricted resource or provide fake evidence by cheating on their locations. To overcome this, A Privacy-Preserving Location proof Updating System is developed 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. A user-centric location privacy model is developed in which individual users evaluate their location privacy levels and decide whether and when to accept the location proof requests. In order to resist the colluders ranking and a correlation scheme is used to detect outliers.

Key words**:** Location-based service, location proof, location privacy, pseudonym, colluding attacks.

I. Introduction

LOCATION-BASED services take advantage of user location information and provide mobile users with various resources and services. Nowadays, more and more location-based applications and services require users to provide location proofs at a particular time. For example, "Google Latitude" and "Loopt" are two services that enable users to track their friends' locations in real time. These applications are location-sensitive since location proof plays a critical role in enabling these applications.There are many kinds of location-sensitive applications. One category is location-based access control. For example, a hospital may allow patient information access only when doctors or nurses can prove that they are in a particular room of the hospital. Another class of location-sensitive applications require users to provide past location proofs , such as auto insurance quote in which auto insurance companies offer discounts to drivers who can prove that they take safe routes during their daily commutes, police investigations in which detectives are interested in finding out if a person was at a murder scene at some time, and location-based social networking in which a user can ask for a location proof from the service requester and accepts the request only if the sender is able to present a valid location proof. The common theme across these location sensitive applications is that they offer a reward or benefit to users located in a certain geographical location at a certain time. Thus, users have the incentive to cheat on their locations.

Location-sensitive applications require users to prove that they really are (or were) at the claimed locations. Although most mobile users have devices capable of discovering their locations, some users may cheat on their locations and there is a lack of secure mechanism to provide their current or past locations to applications and services. One possible solution is to build a trusted computing module on each mobile device to make sure trusted GPS data is generated and transmitted.

For example, Lenders et al. proposed such a solution which can be used to generate unforgeable geotags for mobile content such as photos and video; however, it relies on the expensive trusted computing module on mobile devices to generate proofs. Although cellular service providers have tracking services that can help verify the locations of mobile users in real time, the accuracy is not good enough and the location history cannot be verified. Recently, several systems have been designed to let end users prove their locations through WiFi infrastructures. For example, Saroiu and Wolman proposed a solution suitable for third-party attestation, but it relies on PKI and the wide deployment of WiFi infrastructure.In this paper, we propose A Privacy-Preserving LocA-tion proof Updating System (APPLAUS), which does not rely on the wide deployment of network infrastructure or the expensive trusted computing module. In APPLAUS, Bluetooth enabled mobile devices in range mutually generate location proofs, which are uploaded to a untrusted location proof server that can verify the trust level of each location proof. An authorized verifier can query and retrieve location proofs from the server. Moreover, our location proof system guarantees user location privacy from every party. More specifically, we use statistically updated pseudonyms at each mobile device to protect location privacy from each other, and from the untrusted location proof server. We develop a user-centric location privacy model in which individual users evaluate their location privacy levels in real time and decide whether and when to accept a location proof request. In order to defend against colluding attacks, we also present betweenness ranking-based and correlation clustering-based approaches for outlier detection. The rest of the paper is organized as follows: We first introduce preliminaries of our scheme in Section 2, and then present our location proof updating scheme in Section 3. Section 4 discusses colluding attacks and countermeasures. Finally, described real time implementation in section 5 related work in Section 6 and conclude the paper in Section 7.

II. Preliminaries

In this paper, we focus on mobile networks where mobile devices such as cellular phones communicate with each other through Bluetooth. In our implementation, mobile devices periodically initiate location proof requests to all neighboring devices through Bluetooth. After receiving a request, a mobile node decides whether to exchange location proof, based on its own location proof updating requirement and its own privacy consideration. Given its appropriate range (about 10 m) and low power consump-tion, Bluetooth is a natural choice for mutual encounters and location proof exchange.

A. 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, every mobile node i registers with the CA by preloading a set of M public/private key pairs K_i^{Pub} is used to serve as the pseudonym of node i. The private key K_i^{Prv} enables node i to digitally sign messages so that the receiver can validate the signature authenticity. Due to the broadcast nature of wireless communication, probes are used for mobile nodes to discover their neighbors. When a node i receives a probe from another node, it checks the certificate of the public key of the sender and the physical identity, e.g., Bluetooth MAC address. After that, i verifies the signature of the probe message. Subsequently, if confidentiality is required, a security association is established (e.g., with Diffie-Hellman).

B. Threat Model

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, or individual devices may collude with each other to generate false proofs. We assume that the number of colluders is small compared with that of valid devices. In the worst case, the adversary could compromise the location proof server to get the stored location proof records. However, it is not able to take control of the server to work as a colluder, since once compromised, the attack will be detected promptly and the location proof server will be replaced by a back-up server. The same assumption applies to the CA. By passive, we assume the adversary cannot perform active channel jamming, mobile worm attacks or other denial-of-service attacks, since these attacks are not related to location privacy. By global, we assume the adversary can monitor, eavesdrop, and analyze all the traffic in its neighboring area, or even monitor all the traffic around the server.

In practice, the adversary can thus be a rogue individual, a set of malicious mobile nodes, or eavesdropping devices in the network. In the worst case, it is possible that the untrusted location proof server may be compromised by the adversary and the location information can then be easily inferred by examining the records of location proofs, e.g., the adversary could apply statistical testing such as K-S test to identify a user although no real identity is included. Therefore, we need to appropriately design and arrange the location proof records in the untrusted server so that no private information related to individual users will be revealed even after it is compromised. Hence, the problem we address in this paper consists of collecting a set of location proofs for each peer node and protecting the location privacy of peer nodes from each other, from the adversary, or even from the untrusted location proof server to prevent other parties from learning a node’s past and current location information.

C. 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 linkability. 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 requirement. If this node changes its pseudonym at least once during a time period (mix zone), a mix of its identity and location occurs, and the mix zone becomes a confusion point for the adversary.Consider a mobile network composed of N mobile nodes and each node has M pseudonyms. At time t, for each node i there are a group of mðtÞ pseudonyms observed at the location proof server.

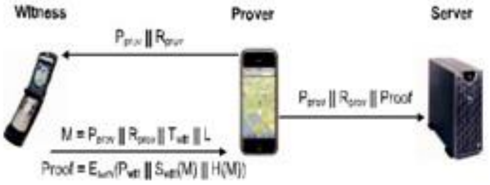

Witness
Prover
Server
$P_{prov} \| R_{prov}$
$P_{prov} \| R_{prov} \| Proof$
$M = P_{prov} \| R_{prov} \| T_{wit} \| L$
$Proof = E_{sym}(P_{wit} \| S_{wit}(M) \| H(M))$

Fig 2 : Location Proof Updating Protocol

Each pseudonym among the mðtÞ pseudonyms can involve multiple location proofs across various locations l_1; l_2; . .; l_n at different time t_1; t_2; . . . ; t_n. An adversary is able to correlate the location and time distribution of each pseudonym to see if two pseudonyms belong to the same node. For example, the adversary can observe a series of location proofs with mðT Þ pseudonyms during time T . He then compares the distribution of location proof set B of pseudonym b with the distribution of location proof set D of pseudonym d to determine if the two pseudonyms can be linked. Let $p_{d¼b}$ ¼ Pr (distribution D of pseudonym corresponds to distribution B of pseudonym.

III. THE LOCATION PROOF UPDATING SYSTEM

In this section, we introduce the location proof updating architecture, the protocol, and how mobile nodes schedule their location proof updating to achieve location privacy in APPLAUS.

A. Architecture

In APPLAUS, mobile nodes communicate with neighboring nodes through Bluetooth, and communicate with the untrusted server through the cellular network interface. Based on different roles they play in the process of location proof updating, they are categorized as Prover, Witness,Location Proof Server, Certificate Authority or Verifier. The architecture and message flow of APPLAUS is shown in Fig. 1.

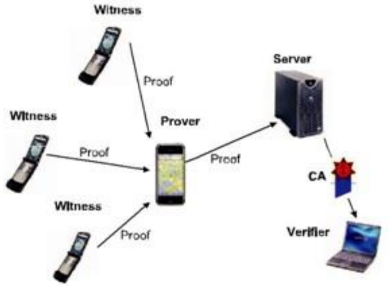


Fig 1: Location Proof Updating Architecture

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. If no positive response is received, the prover will generate a dummy location proof and submit it to the location proof server.

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 history records of the location proofs. It communicates directly with the prover nodes who submit their location proofs. As the source identities of the location proofs are stored as pseudonyms, the location proof server is untrusted in the sense that even though it is compromised and monitored by attackers, it is impossible for the attacker to reveal the real source of the location proof.

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. CA is the only party who knows the mappingbetween the real identity and pseudonyms (public keys), and works as a bridge between the verifier and the location proof server. It can retrieve location proof from the server and forward it to the verifier.

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

B. Protocol

When a prover needs to collect location proofs at time t, it executes the protocol in Fig. 2 to obtain location proofs from the neighboring nodes within its Bluetooth communication range. Each node uses its M pseudonyms $P_i{}^{M}{}_{¼1}$ as its identity throughout the communication.

[1] The prover broadcasts a location proof request to its neighboring nodes through Bluetooth according to its update scheduling. The request should contain the prover's current pseudonym P_{prov}, and a random number R_{prov}.

[2] The witness decides whether to accept the location proof request according to its witness scheduling. Once agreed, it will generate a location proof for both prover and itself and send the proof back to the prover. This location proof includes the prover's pseudonym P_{prov}, prover's random number R_{prov}, witness's current time stamp T_{witt}, witness's pseu-donym P_{witt}, and their shared location L. This proof is signed and hashed by the witness to make sure that no attacker or prover can modify the location proof and the witness cannot deny this proof. It is also encrypted by the server's public key to prevent from traffic monitoring or eavesdropping.

[3] After receiving the location proof, the prover is responsible for submitting this proof to the location proof server. The message also includes prover's pseudonym P_{prov} and random number R_{prov}, or its own location for verification purpose.

[4] An authorized verifier can query the CA for location proofs of a specific prover. This query contains a real identity and a time interval. The CA first authenticates the verifier, and then converts the real identity to its corresponding pseudonyms during that time period and retrieves their location proofs from the server. In order not to expose correlation between pseudonyms to the location server, CA will always collect enough queries from k different nodes before a set of queries are sent out.

[5] The location proof server only returns hashed location rather than the real location to the CA, who then forwards to the verifier. The verifier compares the hashed location with the claimed location acquired from the prover to decide if the claimed location is authentic.

In order to prevent the CA from knowing locations of a real identity, the location proof server calculates the hash of each location and only sends the hashed locations to the CA in step 5. In this way, the following property can be achieved.

Definition 1 (Separation of privacy knowledge). The knowledge of the privacy information is separately distributed to the location proof server, the CA, and the verifier. Thus, each party only has partial knowledge.The privacy property of our protocol is ensured by the separation of privacy knowledge: the location proof server only knows pseudonyms and locations, the CA only knows the mapping between the real identity and its pseudonyms, while the verifier only knows the real identity and its authorized locations. Attackers are unable to learn a user's location information without integrating all the knowledge. Therefore, compromising either party of the system does not reveal privacy information.

IV. COLLUDING ATTACKS AND COUNTERMEASURES

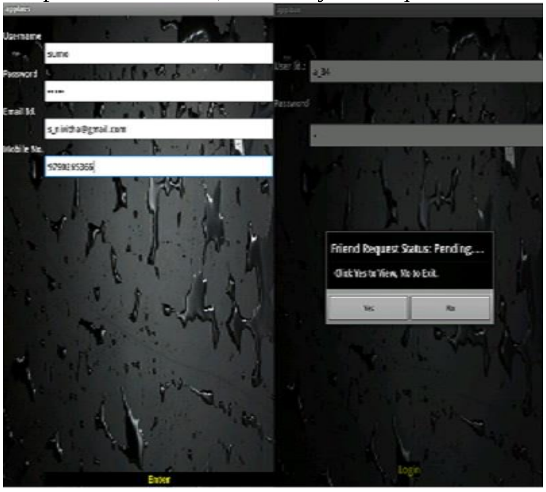
The joint issues of location proof and location privacy have been studied in [33], but the threat of colluding attack is still an open issue. This threat exists when two nodes collude with each other to generate bogus location proofs. For example, when a dishonest node C_1 from San Francisco needs to prove herself in New York City (NYC), she can have another colluding node C_2 to generate bogus location proofs for her, with location tag of New York City. Generally speaking, such attacks can be identified by looking into the location traces and examining the interac-tions between colluders as well as the time and location consistency along the moving trajectory. We first consider statistical threshold based solution in which the system requires the prover to obtain a number of witness nodes, no matter what their real identities are. As we know, the location proof server has information about the number of pseudonyms at a particular time and location. This information can be used to estimate whether the prover lies about not finding enough peers or always finding the same peer based on some statistical techniques.

More specifically, the server-level detection is performed on individual location proof based on its embedded time stamp and location information, where all concurrent and co-located location proofs from other nodes (pseudonyms) are used to verify its trust level.Calculating the trust level of a location proof involves the examination of its surrounding location proofs for both prover and witness, as well as large amount of redundant calculations between individual location proofs. To address this problem, we develop techniques that can perform verifications on a set of location proofs which are relevant in time and space, rather than individual proofs. We present two approaches to detect suspicious location proofs and pseudonyms: betweenness ranking and correlation clustering. The betweenness ranking approach calculates the betweenness of each pseudonym in a graph and then ranks these pseudonyms based on their betweenness value. The pseudonyms with low ranking are considered as suspicious nodes. The correlation clustering approach takes into account the time delay between two neighboring location proofs, and uses a modified correlation clustering algorithm on a temporal-weighted graph to rule out outlier clusters, which are considered as suspicious location proofs. Both approaches use undirected graph to reflect the relationship between pseudonyms or between location proofs.

V. IMPLEMENTATION

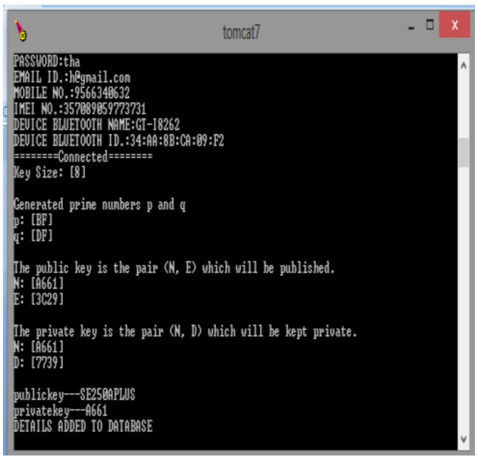
User Registration and Friend Requisition in Social Network

The users in the APPLAUS network have an initial level registration with the server to make use of its services. In response, the server generates an userid for the user. Every mobile node is been loaded with a set of public/private key pairs before entering the network. The user's Bluetooth device name and device id are also been stored at the back end of the server. The user thereafter makes a login, and is been forwarded social network where he finds new users to make a friendship. This requested user's request is displayed to the third party user when he makes a login into his account. The third party user can accept and form a friendship in the network, else deny the request.



Server Requisition for Witness

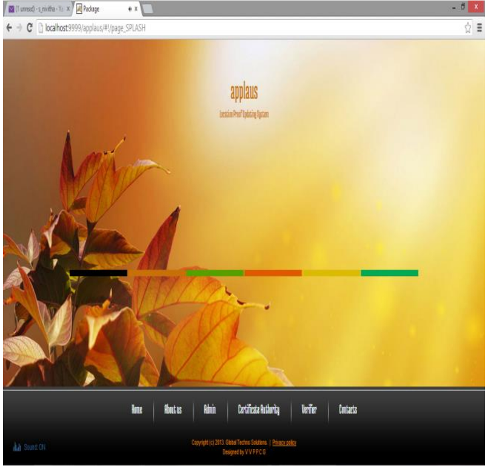
A third-party user or an application who is authorized to verify a prover's location within a specific time period having close relationship with the prover, e.g., friends or colleagues, to be trusted enough to gain authorization forwards his request to the server. The Server forwards a message to the prover in which our application automatically starts to run in his mobile phone. The Prover now analyses his mobile's paired Bluetooth device names and forwards it to the server, just to know whether they too belong to our network. The Server receives the request and checks the users who belong to our network and forwards those filtered names to the user back.



```
PASSWORD:tha
EMAIL ID.:h@gmail.com
MOBILE NO.:9566340632
IMEI NO.:357089059773731
DEVICE BLUETOOTH NAME:GT-I8262
DEVICE BLUETOOTH ID.:34:AA:8B:CA:09:F2
========Connected========
Key Size: [8]

Generated prime numbers p and q
p: [BF]
q: [DF]

The public key is the pair (N, E) which will be published.
N: [A661]
E: [3C29]

The private key is the pair (N, D) which will be kept private.
N: [A661]
D: [7739]

publickey---SE250APLUS
privatekey---A661
DETAILS ADDED TO DATABASE
```

Location Proof Updating System

The prover, who needs to collect location proofs from its neighboring nodes at time t, will broadcast a location proof request to its neighboring nodes through Bluetooth. This request includes the prover's pseudonym (prover's mobile Bluetooth device name, device id., user information) and a randomly generated number. 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. The witness turns on his GPS and generates his current latitude and longitude positions and then forwards its response back to the prover. This response includes the prover's pseudonym, prover's randomly generated number, witness's pseudonym and witness's randomly generated number. If no positive response is received, the prover will generate a dummy location proof and submit it to the location proof server. These location informations will be viewed by the Verifier in the form of a map.


(1 unread) - s_nivitha - Y
Package
localhost:9999/applaus/#!/page_SPLASH
applaus
Location Proof Updating System
Home
About us
Admin
Certificate Authority
Verifier
Contacts
Sound: ON
Copyright (c) 2013. Global Techno Solutions. | Privacy policy
Designed by VVPPCG

VI. RELATED WORK

Recently, several systems have been proposed to provide end users the ability to prove that they were in a particular place at a particular time. The solution in relies on the fact that nothing is faster than the speed of light in order to compute an upper bound of a user's distance. Capkun and Hubax propose challenge-response schemes, which use multiple receivers to accurately estimate a wireless node location using RF propagation characteristics. In, the authors describe a secure localization service that can be used to generate unforgeable geotags for mobile content such as photos and video. However dedicated measuring hardware or high-cost trusted computing module are required. Saroiu and Wolman propose a solution suitable for third-party attestation, but it relies on a PKI and the wide deployment of WiFi infrastructure. Different from these solutions, APPLAUS uses a peer-to-peer approach and does not require any change to the existing infra-structure. SmokeScreen introduces a presence sharing mobile social service between colocated users which relies on centralized, trusted brokers to coordinate anonymous communication between strangers. SMILE allows users to establish missed connections and utilizes similar wireless techniques to prove if a physical encounter occurred. However, this service does not reveal the actual location information to the service provider thus can only provide location proofs between two users who have actually encountered. APPLAUS can provide location proofs to third-party by uploading real encounter location to the untrusted server while maintaining location privacy.

There are lots of existing works on location privacy in wireless networks. In , the authors propose to reduce the accuracy of location information along spatial and/or temporal dimensions. This basic concept has been improved by a series of works . All the above techniques cloak a node's locations with its current neighbors by trusted central servers which is vulnerable to DoS attacks or to be compromised. Different from them, our approach does not require the location proof server to be trustworthy. Xu and Cai propose a feeling-based model which allows a user to express his privacy requirement. One important concern here is that the spatial and temporal correlation between successive locations of mobile nodes must be carefully eliminated to prevent external parties from compromising their location privacy. The techniques in achieve location privacy by changing pseudonyms in regions called mix zones. In this paper, pseudonyms of each node are changed by the node itself periodically following a Poisson distribution, rather than being exchanged between two untrusted nodes. Identifying a fundamental tradeoff between performance and privacy, Shao et al propose a notion of statistically strong source anonymity in wireless sensor networks for the first time, while Li and Ren and Zhang et al. tried to provide source location privacy against traffic analysis attacks through dynamic routing or anonymous authentication. Our scheme uses similar source location unobservability concept in which the real location proof message is scheduled through statistical algorithms. However, their focus is to generate identical distributions between different nodes to hide the real event source, while our focus is to design distinct distributions between different pseudonyms to protect the real identity.

VII. CONCLUSION

In this paper, we proposed a privacy-preserving location proof updating system called APPLAUS, where colocated Bluetooth enabled mobile devices mutually generate location proofs and upload to the location proof server. We use statistically changed pseudonyms for each device to protect source location privacy from each other, and from the untrusted location proof server. We also develop a user-centric location privacy model in which individual users evaluate their location privacy levels in real time and decide whether and when to accept a location proof exchange request based on their location privacy levels. To the best of our knowledge, this is the first work to address the joint problem of location proof and location privacy. To deal with colluding attacks, we proposed betweenness ranking based and correlation clustering-based approaches for outlier detection. Extensive experimental and simulation results show that APPLAUS can provide real-time location proofs effectively. Moreover, it preserves source location privacy and it is collusion resistant.

REFERENCES

[1] Alastair R. Beresford and Frank Stajano," Location Privacy in Pervasive Computing " IEEE Security and Privacy, 2003.

[2] Srdjan Capkun and Jean-Pierre Hubaux ,"Secure positioning of wireless devices with application to sensor networks" Proc. IEEE INFOCOM, 2005.

[3] Stefan Saroiu, Alec Wolman ,"Enabling New Mobile Applications with Location Proofs" Proc. ACM 10th Workshop Mobile Computing Systems and Applications (HotMobile '09), 2009.

[4] Vincent Lenders, Emmanouil Koukoumidis, Pei Zhang and Margaret Martonosi,"Location-based Trust for Mobile User-generated Content: Applications, Challenges and Implementations" Proc. Ninth Workshop Mobile Computing Systems and Applications, 2008.

[5] Wanying Luo & Urs Hengartner," Proving Your Location without Giving up Your Privacy " Proc. ACM 11th Workshop Mobile Computing Systems and Applications (HotMobile '10), 2010.

[6] Yanchao Zhang, Wei Liu and Wenjing Lou1L ,"Anonymous Communications in Mobile Ad Hoc Networks "Proc. IEEE INFOCOM, 2005.