



## A Third Generation Privacy Preserving Location Proof Updating System

Remya P V, Rijin I K  
CSE Department, MIT Anjarakandy,  
Kannur University, India

---

**Abstract**— Nowadays most mobile users have devices capable of finding their locations. Location based application require the user to provide location proofs at a particular time in use. There are some cases were users may cheat on their locations. So we need a secure technique to provide locations to applications. In the base paper- 'APPLAUS', the technology used for communication is Bluetooth. Due to all limitations in Bluetooth, here we propose 'A third generation privacy preserving location proof updating system' where individual users evaluate their location privacy levels and decide whether to accept or reject the location proof requests. In proposed system I am considering the network which includes Smartphone having android OS with 3G facility and GPS module. The server is implemented on a laptop. It stores the uploaded location proof records. This work is cost effective and can be implemented feasibly when compared to other technologies used. This implementation is expected to arrive in better speed of communication, low average delay, improved process delivery ratio and lowered overhead ratio.

**Keywords**— Location proof, Parse, Prover, Pseudonyms, Witness

---

### I. INTRODUCTION

Location-based services take advantage of user location information and provide mobile users with various resources and services. Today, many 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 mainly location-sensitive since location proof plays a critical role in enabling these applications. There are several location-sensitive applications. One kind is location-based access control. For example, a hospital [1] may allow patient information access only when doctors or nurses can prove that they are in a particular room of the hospital. There is another class of location-sensitive applications [2] which requires users to provide past location proofs such as auto insurance quote in which the auto insurance companies offer discounts to drivers who can prove that they take safe routes during their daily commutes, 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 of 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 motive to cheat on their locations. The Location-sensitive applications would require users to prove that they really are (or were) at the claimed locations. Most mobile users have devices capable of discovering their locations, but in some cases 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. Such a solution which could be used to generate un-forgable geo-tags for mobile content such as photos and video, It would however depend on the expensive trusted computing module on mobile devices to generate proofs. Cellular service providers in real-time tracking services that will help to verify the positions of mobile users, the accuracy are not good enough and proven history. In the proposed system, we are just extending previous work APPLAUS by including 3G as the communication and Parse mobile application service. Parse allows the developers to connect their app to back end cloud storage. It also provides features such as user management, push notifications etc. We are registering our app in Parse, so that the request and response are in the form of push notification through 3G network. We protect the privacy of each other devices and server, by using pseudonyms. Therefore, the performance of many applications the use of pseudonyms is concerned is just as good as using real identities The proposed model consists of 3G enable Smartphone having android OS generate location proofs and send updates to a location proof server which verifies the location with the help of verifier.

The rest of the paper is organized as follows. We first introduce related work in section 2 and then proposed system in section 3. Section 4 discuss result and conclude the paper in section 5.

### II. RELATED WORK

The technology today allows more and more content to be provided by the mobile user. The user-generated content is provided in the form of podcasts, blogs etc. Vincent Lenders et al. proposed Location-Based Trust for Mobile User-Generated Content [3]. It mainly deals with how to establish the authenticity of content created by users. The system

consists of three entities: content producers, content consumers, and a location/time verification service. When a content producer has some content that it wants to have geographically certified, it issues a request to the localization/certificate authority (step 1). This request includes a hash over the content it wants to have certified. Then, it replies back to the content producer with a Data-Location-Time (DLT) certificate (step 2) that binds the location of the content producer with the current time and the hash of the content. The content producer now has the option to publish its content with the issued certificate (step 3). When the content consumer retrieves the content, it can now verify the origin location and time of the content by verifying the authenticity of the certificate (step 4), i.e., by verifying the signature of the certificate using the public key of the localization/certificate authority. The problems in this work are expensive trusted computing module on mobile devices to generate location proofs, location history cannot be verified, mobility attack and delay attack.

The next approach was proposed by T. Xu and Y. Cai in the paper Feeling-based Location Privacy Protection for Location-based Services [4]. The model allows a user to express her privacy requirement by specifying a public region, which the user would feel comfortable if the region is reported as her location. The popularity of the public region, measured using entropy based on its visitors' footprints inside it, is then used as the user's desired level of privacy protection. Collect location samples from cellular phone users. These location samples, each called a footprint, can then be used to measure the popularity of a spatial region. Assume mobile clients communicate with LBS providers through a trusted central location depersonalization server (LDS) managed by the clients' cellular service carriers. The LDS randomly generates a service session ID and contacts the service provider. After establishing a service session, the service user periodically reports her current location to the LDS. The problems in this work are, only prevent an adversary from correlating anonymous location information with restricted spaces such as home or office. Observation implication attack – adversary has direct observation over the region.

The next approach was Enabling New Mobile Applications with Location Proofs [2] developed by Stefan Saroiu, Alec Wolman. Location proof is a piece of data that certifies a receiver to a geographical location. Location proofs are incrementally deployable – any cell tower or Wi-Fi access point can start to support them with very limited coordination with other parts of the infrastructure. This coordination is limited to the proof verifier needing a trust relationship with the proof provider (i.e., the public key). A location proof has five fields: an issuer, a recipient, a timestamp, a geographical location, and a digital signature. Wi-Fi access points broadcast beacon frames to announce their presence. Upon receiving a beacon, a client can decide whether to explicitly request a location proof from the respective AP. To request a proof, the client extracts the beacon's sequence number to use it in the request for the location proof. The request for a location proof contains the client's public key and the signed AP's sequence number. The client signs the sequence number to protect their integrity and to make it hard for clients to impersonate other devices. Upon receiving the request, the AP checks whether the signature is valid and whether the sequence number is a current one. In case of a valid request, the AP creates a location proof with a current timestamp and designates the client as the recipient. After creating the location proof, the AP broadcasts it. The AP does not check whether the client received the location proof. The problems in this work are, physical attacks pose a significant threat to location proofs. For example, an AP can be stolen and relocated, or it can be broken into to change its latitude and longitude coordinates. Also AP's should monitor client continuously, and very expensive to maintain Wi-Fi infrastructure.

A Privacy-Preserving Location proof Updating System [5] developed by Zhichao Zhu, and Guohong Cao next used, in which co-located 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 [6][7] from each other, and from the un-trusted location proof server.

### III. PROPOSED SYSTEM

Based on roles of different devices in location proof updating system, they are categorized as Prover, Server, Witness, Certification Authority (CA) and Verifier. Every mobile node  $i$  will have to register with the CA by preloading a set of  $M$  public/private key pairs before entering the network. The public key will be used to serve as the pseudonym of node  $i$ . The private key will enable node  $i$  to digitally sign messages so that the receiver will be able to validate the signature authenticity. The architecture is shown in fig 1. When the device login is successful, the device that needs to collect location proofs from its neighbouring nodes act as Prover. When a location proof is needed at a time  $t$ , the prover will broadcast a location proof request to its neighbouring nodes. This request is in the format of a Push notification through Parse [8] which contains prover's current pseudonym  $P_{prov}$ , and a random number called  $R_{prov}$ . Here a neighbouring node when agrees to provide location proof for the prover, now this node will become a witness of the prover. Now the witness node will generate a location proof and send it back to the prover. The location proof contains prover's pseudonym  $P_{prov}$ , prover's random number  $R_{prov}$ , witness's current time stamp  $T_{witt}$ , witness's pseudonym  $P_{witt}$ , and their shared location  $L$  (Longitude and Latitude). The Location proof is now encrypted using the server's public key to prevent from traffic monitoring. Here, after getting 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 the verification purpose.

Server our goal is not only to monitor real-time locations, but also to retrieve history of location proof information when needed, a location proof server is here necessary for storing the records of history 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 here stored as pseudonyms, the location proof server is untrusted in the category 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. Now Prover can send a request to verifier. This request includes prover's pseudonym Pprov, prover's random number Rprov.

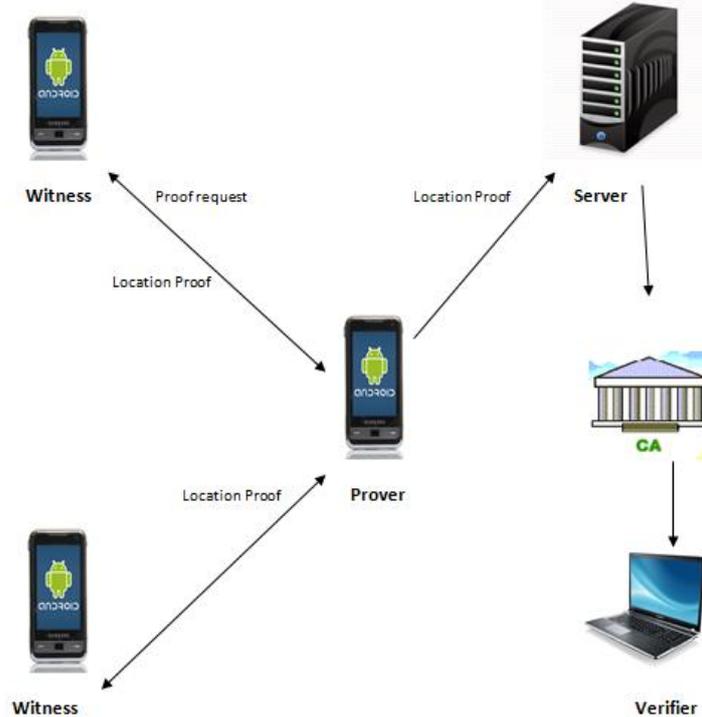


Fig 1 Architecture

Verifier is a third-party user or can be an application that is authorized to verify a prover's location. An authorized verifier can query the CA for the 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 the corresponding pseudonyms during that time period and retrieves their location proofs from the server. We consider an online CA which is run by an independent trusted third party. CA is the only party who knows the mapping between the real identity and pseudonyms (public keys), and works as a bridge between the verifier and the location proof server. 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.

#### IV. RESULTS

Each device must register with CA before entering update process. So CA has device's real identity that is IMEI number and its pseudonym. At the Server side Verifier must be registered. The successful login of device follows location proof request in the form of push notification. The push notification delivered to witness. Witness will generate location proof which is encrypted using RSA and send to Prover. Prover is now ready to submit this proof to Server. Now server has prover's pseudonym, its received encrypted location and prover's location. Location verification page of verifier contains an arraylist of name, IMEI number and verify button. By clicking verify button, decryption followed by verification is performed and that location is displayed in Google map.

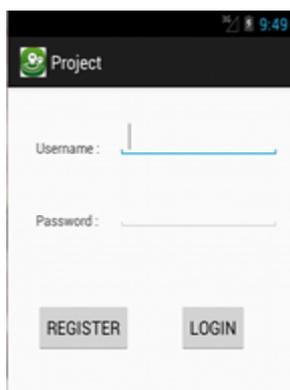


Fig 2 Device Registration

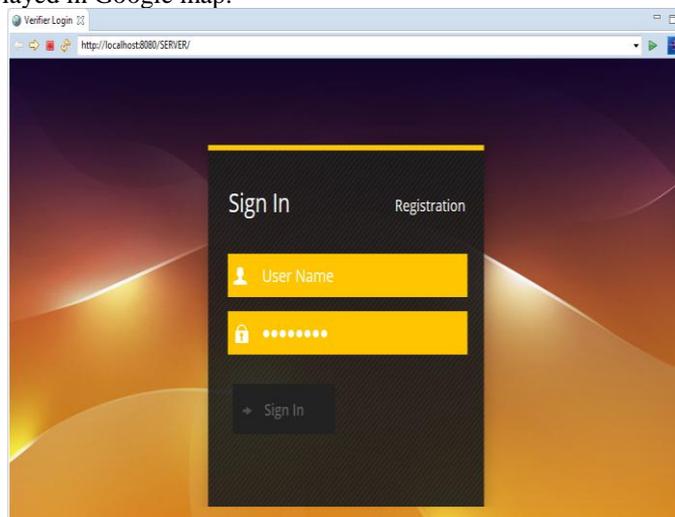


Fig 3 Verifier Registration

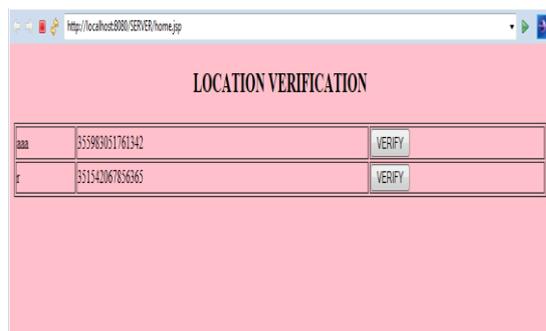


Fig 4 Verification Page of Verifier

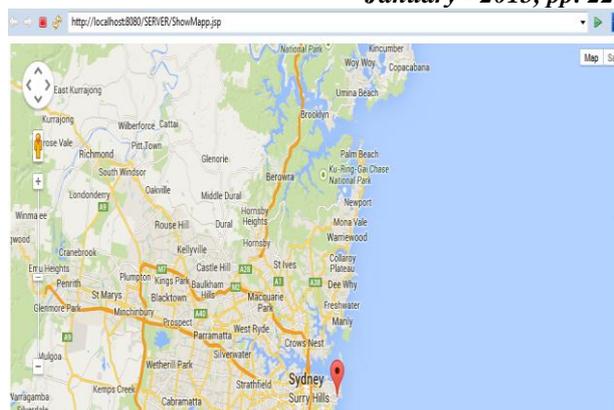


Fig 5 Verified Location in Map

## V. CONCLUSION

In the previous work, the Bluetooth technology was used for communication. Bluetooth is a wireless technology standard for exchanging data over short distances. If attempts to pair one Bluetooth device with another Bluetooth device are unsuccessful, the problem may be caused by one or more of the following: the power switch for the device is turned off, Bluetooth is disabled on the device, the device is not in the discoverable mode, or the Bluetooth module needs to be reseeded. Many connection and communication problems are caused by accidentally selecting the wrong device. Some devices have a time limit on how long they stay in discoverable mode or how long they wait while the passkey is entered. In these cases, if the device is not discovered or the passkey for pairing is not entered within that time period, the connection will fail.

In this paper, we proposed a third generation privacy-preserving location proof updating system, where 3G enabled mobile devices generate location proofs and upload to the location proof server. We used pseudonyms for each device to protect source location privacy from each other, and from the un-trusted location proof server. To protect location privacy, every mobile node  $i$  register with the Certification Authority by preloading a set of  $M$  public/private key pairs before entering the network. We used parse push notifications that simplifies the process and enables powerful targeting on android, ios and windows. 3G-capable Smartphone provide the practical services, exciting features, and fast speeds that users expect in a high-end phone. In future, this work can also be implemented in ios and windows platforms.

## REFERENCES

- [1] W. Luo and U. Hengartner, "Proving Your Location Without Giving Up Your Privacy," *Proc. ACM 11th Workshop Mobile Computing Systems and Applications (HotMobile '10)*, 2010.
- [2] S. Saroiu and A. Wolman, "Enabling New Mobile Applications with Location Proofs," *Proc. ACM 10th Workshop Mobile Computing Systems and Applications (HotMobile '09)*, 2009.
- [3] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi, "Location-Based Trust for Mobile User-Generated Content: Applications Challenges and Implementations," *Proc. Ninth Workshop Mobile Computing Systems and Applications* 2008
- [4] T. Xu and Y. Cai, "Feeling-Based Location Privacy Protection for Location-Based Services," *Proc. 16th ACM Conf. Computer Comm. Security (CCS)*, 2009.
- [5] Zhichao Zhu, Student Member, IEEE, and Guohong Cao, Fellow, IEEE "Toward Privacy Preserving and Collusion Resistance in a Location Proof Updating System", *IEEE TRANSACTIONS ON MOBILE COMPUTING*, VOL. 12, NO 1 JANUARY 2013
- [6] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.C. Herrera, A.M. Bayen, M. Annavaram, and Q. Jacobson, "Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring," *Proc. ACM MobiSys*, 2008.
- [7] M. Li, K. Sampigethaya, L. Huang, and R. Poovendran, "Swing & Swap: User-Centric Approaches Towards Maximizing Location Privacy," *Proc. Fifth ACM Workshop Privacy in Electronic Soc.*, 2006.
- [8] <http://www.parse.com>