Approximation of Mobile User Location by Distance Bounding Scheme

[1]Kalaimathi.T, [2]Mrs., Prabula
III MCA, Lord Jegannath College of Engineering & Technology
Assistant Professor/Department of Computer Applications,
Lord Jegannath College of Engineering & Technology

ABSTRACT

AS LOCATION ENABLED mobile devices proliferate, location based services are rapidly becoming immensely popular. Most of the current location based services for mobile devices are based on user's 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. Our distance bounding scheme 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. Our distance bounding scheme ensures the integrity and non -transferability of the location proofs and protects user's privacy. Our prototype implementation on the Android stage shows that our distance bounding scheme is Low cost in terms of computational and storage possessions.

Index Terms- Location proof, privacy, spatial temporal provenance, trust.

1. INTRODUCTION

1a.The propagation of small mobile devices has sparked an interest in systems that can determine the location of a device with high precision. While researchers have achieved important results in this area, they have given less attention to the security of these location determining schemes. Integrity and privacy are both significant elements of a location proving systems security. The integrity of a location system is important because often a user in control of s device will have incentive to falsify the report of its location. A system that maintains proper security should protect against an attack from such a user. Privacy is also critical to a location system. In many applications the location of a device is privileged information .A location proving system should protect against unwanted parties learning this information.

The project presents a system that allows a device to securely prove its location to another party. We believe we are the first to consider a location proving system that provides both integrity and privacy.

1b. The present a model in which a party know as the verifier is interested in the location of a device to which it does not have immediate access. The verifier might trust the device if it were manufactured to be tamper resistant ,but does not trust the environment surrounding the device, including the user in possession of the device.

This model is motivated by practical situations. For example, lenders of customer equipment will often find themselves in the position of the verifier. Universities that lend laptops to the students might

wish to have the laptops remain within the confines of the university campuses. Operators of electronic home arrest monitoring systems have a similar problem [6]. In these systems have a tamper resistant device is attached to the ankle of person under house arrest. The verifier wants to make sure that the device (and thus the person) is at the house during certain hours of the day.

In many circumstances the verifier will not have control of the networking infrastructure at the location of the device. The verifier will then need to turn to a third party that will help the device prove its location. One possibility is to use a large global system such as the Global Positioning System (GPS) or the cell phone network. While these systems have been useful for many applications their usefulness in proving location is limited. These large systems did not have location proving as original design objectives and to adapt them for such as purpose would be costly and complex. Additionally the coverage these systems does not reach many indoor areas where location proving might be desirable.

We turn our attention to small wireless networks where each small network can vouch for the presence of devices in a small area that it covers. We call the access points of these networks as location managers is that the coverage of location proving networks can grow incrementally.

The use of several location managers presents challenges in the design of a location proving system. A verifier must decide who it will trust to be a location manager for a given area. The location manager must be trustworthy and have a networking infrastructure capable of facilitating location proofs in the area. The challenge of choosing a suitable location manager becomes difficult when the number of locations that a device might potentially visit is large and the verifier is unable to investigate each one individually.

A large privacy issue exists in this model. The device will possibly be proving its location with several different location managers. As discussed the identities of the device and verifier are privileged information in many applications. It is important that this information does not leak out to eavesdroppers or even the location managers facilitating the proofs.

2. Related Work

2a. The Global positioning system is probably the most widely recognized location determining system [1]. However its use in proving the positioning of devices seems limited as false input of GPS signals can be generated by users in possession of the devices [4, 5]. There exists a military segment of GPS known as the Precise Positioning Service (PPS).In PPS signals are encrypted so that they cannot be forged by a user. This service has not been open to the commercial sector. Even if PPS were commercially available this would entail trusting every device that this service to hold a global encryption secret. The use of a global secret seems very unlikely to work with a large deployment of devices.

RF Technologies, Inc's local positioning system (LPS) is used for tracking and locating office equipment [10]. A novel aspect of their work is that they use the round trip latency of signal propagation to measure the distance from a tag placed on a mobile object to an antenna at a fixed position. A signal is sent from the antenna and the tag uniquely transform the signal the antenna then reads the processed signal from the tag and records the latency. The processing time of the signal in the tag is fixed so the component of latency due to signal propagation can be isolated. The system uses multiple antennas to triangulate on the position of the tag. The designers achieved an accuracy of approximation 2m using a 40MHZ clocking rate of the chip. This project demonstrates the feasibility of using radio round trip latency to estimate the distance from a mobile object to fixed base station. The project was not designed to be robust against malicious attacks.

2b. everyplace [2] is a location proof architecture with is designed with privacy protection and collusion resilience. However it requires three different trusted entities to provide security and privacy protection: a TTPL (Trusted Third Party for managing Location in information) a TTPU (Trusted Third Party for managing user information) and a CDA (Cheating Detection Authority).Each trusted entity know either user identity or his/her location, but not both. Everyplace collusion detection works only if users request their location proof very frequently so that the long distance between two location proofs that are chronologically close can be considered as anomalies .This is not a realistic assumption because users should have the control over the frequency of their requests.

Has a et al [4] proposed a scheme which relies on both location proofs from wireless APs and wireless endorsements from Bluetooth enabled mobile peers, so that no users can forge proofs without colluding with both wireless APs and other mobile peers at the same time. It eliminates the necessity of multiple trusted parties. Two privacy preserving schemes based on hash chains and bloom filters respectively are described for protecting the integrity of the chronological order of location proofs.

3. Power Framework

Proposed system

In this project we propose an STP proof scheme named Spatial -temporal Provence Assurance with distance bounding scheme. Our distance bounding scheme aims at ensuring the integrity and non transferability of the STP proofs, with the capability of protecting user's privacy. Most of the existing STP proof schemes rely on wireless infrastructure (eg.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 battlefield example certainly cannot be obtained from wireless APs. To target a wider range of applications, our distance bounding scheme 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 not eliminate the possibility of utilizing wireless infrastructure as more trusted proof generation source. In addition ,in contrast to most of the existing schemes which require multiple trusted or semi trusted third parties, Our distance bounding scheme requires only a single semi trusted third party which can be embedded in a Certificate Authority (CA).We design our system with an objective of protecting users anonymity and location privacy.

As we explained, 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.

Architecture Diagram:

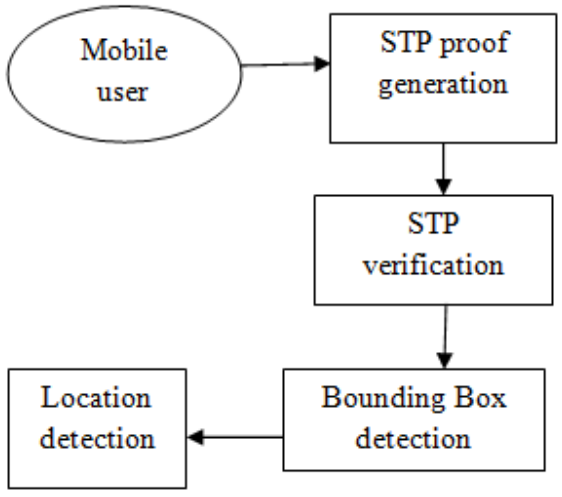


Fig.1

Architecture Diagram

In addition, the deployment cost would be high if we require a large number of wireless APs to have the capacity of generating STP proofs. Therefore, we think a distributed STP proofs 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.

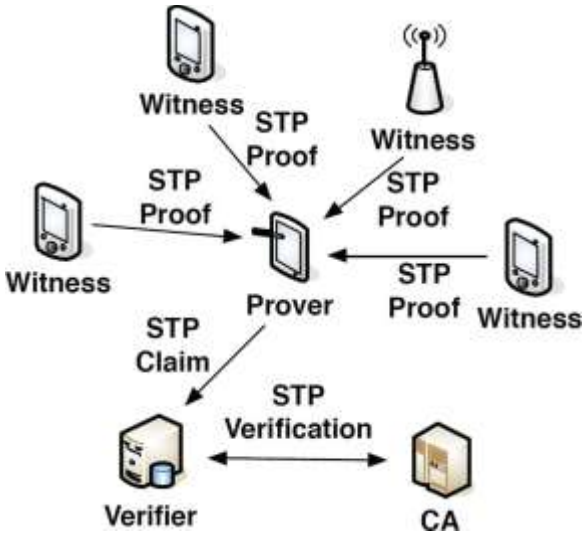
Fig.1 illustrates the architecture of our system. There are four types of entities based on their roles:

Prove: A prove is a mobile device which tries to obtain STP proofs at a certain location.

Witness: A observer is a device which is in proximity with the prove and is willing to create an STP proof for the prove and upon receiving his/her request. The observer can be entrusted or trusted, and the trusted witness can be mobile or stationary (wireless APs) collocated mobile users are entrusted.

Verifier: A verifier is the party that the prove 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 (entrusted for privacy protection, see section IV-C for details) which issues, manages cryptographic credentials for the other parties.CA is also responsible for proof verification and trust evaluation.



A proves and a witness communicate with each other via Bluetooth or Wi-Fi in ad hoc mode. A peer discovery mechanism for discovering nearby witness is required and preferably provided by underlying communication technology instead of our protocol. The proof generation system of prove is presented a list of available witness. When there are multiple witnesses willing to corporate, the prove initiate protocol with them sequentially STP claims are sent to verifiers from prove via a LAN or internet and verifiers are assumed to have internet connection with CA. Each user can act as a prove or a witness depending on their roles at the moment. We assume the identity of a user is bound with his/her public key, which is certified by CA. Users have unique public /private key pairs, which are established during the user registration with CA and stored on users personal devices. There are storing incentives for people not to give their privacy away completely, even to their families or friends, so we assume a user never gives his/her mobiles device or private key to another party.

4. Framework Design

4a.Threat Model:

Proves: A malicious proves seeks to create fake STP proofs without physically being present at a location. This includes creating fake STP proofs by himself ,lying to a witness about his/her location, tampering with the spatial temporal information in his/her existing proofs, as well stealing and using another users STP proofs. Moreover, a malicious proves also attempts to obtain a witness's identity information in the entire process of STP proof generation.

Witness: A malicious witness goals include acquiring a prove identity information and repudiating an STP proof that is generated by him/her.

Verifier: A verifier is often a service provider or an authority that is trying to validate a prove STP claim. A

proves has to present both his/her identity and STP information to the verifier in order to get a service or simply prove his/her alibi. We assume that a verifier is trusted in terms of privacy leakage that is a verifier never leaks a prove identity or STP information to any other parties. However, a prove should be able to only give a verifier his/her STP information that is necessary. In other words, a prove should have the control over which STP proofs and what location granularity of the STP proofs are revealed to a verifier.

We assume CA is trusted but curious, in the sense that it is only trusted in term of correctly performing its functions i.e., user registration key and credential management and trust assessment for STP proofs. Also, CA does not intentionally leak any information it stores to other individuals users. However CA may intend to use any thus a potential privacy abuse may happen at CA.

Collusion: We specifically tackle two different collusion scenarios in this work.(1) A witness can collude with a prove by creating an STP proof for him/her even though one or both of them are not at the location as claimed in the STP proof. We name this collusion scenario as **W-P collusion.** To the best of our Knowledge there is no good solution to detect this type of collusion yet. (2)A prove who is at a specific location to masquerade as him/her and generate a fake STP proof. while we suppose A does not give his/her personal key to B it is possible for A and B to have a hidden communication tunnel during the STP proof generation process so that B could relay message to A, A signs on them and returns them to B in real time. This kind of collusion attack is a type of Wormhole attack [10], which has been more commonly referred to as the Terrorist Fraud attack [7] in location verification. It is one of the most challenging attacks to protect against in location verification. Applied to our context, we name this collusion scenario as P-P collusion.

4b.Protocol Description:

We outline the protocol as a sequence of steps taken by the following parties, the verifier (v) location manger (LM), and Device (D).We assume that the verifier and location manager both have asymmetric encryption key parries associated with them and the public keys for these parries will be noted as Kv and Klm respectively. The Device and Location Manager will also each have a signature key the private keys will be noted as KD and Klm respectively. Additionally all encryption and signature operations are assumed to be properly randomized.

1. D->LM: E klm (start, reply, E kV (Dev ID)) the device sends an encrypted message to the location manager. The messages contain two randomly generated nonce's (start and reply).Each nonce is long enough so that the chance of an adversary randomly guessing them is negligible. The Device also sends its encrypted ID to the Location Manager. The ID is encrypted with the verifiers public key so that the Location Manager will not be able to read it.(Recall that encryption is properly randomized so the cipher text will be unique each time).

2. The Location Manager starts its timer.

3. LM->D: **start, end)**

The Location sends the nonce **Start** and a new nonce **echo** to the Location Manager.

4. D->LM**: reply, echo**

The Location Manager sends the nonce's **reply** and **echo** to the device.

5. Upon receiving reply and **echo** back the location manager immediately stops its timer and records the round trip latency.

6. LM->D: Sklmsign (**latency, current time**, Ekv (**DevID**)) The Location Manager signs a message containing the latency, current time, and the encrypted Devices ID. The Location Manager will subtract out this fixed

internal processing delay from the latency measure. The Device will check that the encrypted ID matches what he gave step 1 and that the current time is correct.

7. D->V: E kV (**skdsign (Dev ID, Loc ID**, Sklmsign (**latency, current time**, Ekv (**DevID**))))

The Device signs its ID the Location Managers ID and the measured latency along with the signed message it received from the Location Manager the previous step. It them encrypt this with the verifier public key and send the encrypted message to the verifier. In the previous steps we assume that the devices and Location Manager have a direct wireless path to communicate on, but for the final step the encrypted message could be sent through any network.

5. Analysis

In this section we provide a security analysis of our protocols.

Integrity:

Suppose that both the device and the location manager behave honestly since the first message is encrypted only the Device and Location Manager will know the nonce's **start** and **reply.** When the Location Manager transmits the **start** nonce the RF signal will propagate to the Device. Once the nonce is transmitted there is nothing that an adversary can do to help it arrive at the Location Manager sooner so the instant an adversary learns the **start** nonce that knowledge becomes useless to him. The same principle applies in the reverse direction. The integrity is derived the fact that no signal can travel faster than the original RF wave. If another slower means such as ultra sound were used for communicating then the protocol would not be secure. Additionally the device will only use the signature from step 6 if it includes his encrypted ID. This ensures that the latency measurement is matched with the correct Device.

Recall the we have assumed the Device to be tamper resistant, meaning that all processing steps of the device in the protocol are done in tamper resistant hardware. If the device is compromised or was not tamper resistant to begin with we would still like to have some level of security. Suppose the device is compromised by an adversary then in order for the adversary. Then in order for the adversary to make the verifier believe that the Device, was a given distance from the Location Manager, the adversary must control a proxy within that the distance. This follows from the fact that the **echo** nonce must be sent back to the Location Manager during the timed phase. If the adversary wishes to reply with the **echo** nonce soon enough he must control a proxy that is close enough to do that. Additionally, the encrypted Device ID is committed to the Location Manager adversary controlling a compromised Device cannot hijack another proof.

Integrity for visibility protocol let's consider the integrity of a proof based upon the alternative definition of network visibility due to the location unless it can communicate with the location manager. However a powerful adversary could execute a proxy attack and effectively extend the visibility of the network to the Device. Such an attack could be executed by a variety of means. For example an adversary could amplify the antenna of a Device to widen its wireless range. Alternatively, the adversary could have two boxes at the Location Manager and the Device. It could the record and reply traffic between the two. The feasibility of such attacks depends on factors such as the physical security of the area around the Location Manager. In many circumstances the cost of proxy attack will be prohibitive and the alternative version of the protocol will suffice.

Privacy:

The verifier might wish to keep it's identify and the Device's identify private even to the Location Manager. To that end the Device does not identify itself to the

Location Manager in any of the steps of the protocol. The final message is encrypted so that only the verifier can read it. An eavesdropper could attempt to determine to whom the Device was sending the message in the final step by sniffing the network. However, there exist several systems that can be used to foil traffic analysis attacks [9, 3, 8]. Eavesdroppers can probably learn that a proof of Location is taking place, but they will not learn who the Device and Verifier are.

The identity of the Device could also be discovered by methods that work outside the basic framework of the protocol. For example, if every Device had a unique Medium Access Channel (MAC) identifier as common Ethernet network cards do then this could be used to identify a Device. Another possible leak of privacy occurs when a user in possession of the Device knows the Device's identify and gives it away through an out of band technique. Implementers of a real system should take care to avoid at least identify these privacy pitfalls.

Denial of service:

Our system does not defend against Denial of Service attacks. An adversary could potentially block the Device from communicating to the outside world. Additionally, an adversary could place a buffer between the Device and the outside world it appear as through the Device was farther away from a Location Manager than it actually was. The severity of such Denial of Service attacks and the methods that should be used to deal with them will depend upon the application for which the system is being used.

Other Issues:

The integrity of our system relies upon both the Device and the Location Manager being able to execute the timed steps of the protocol in a very predictable manner with low variability in the processing times. Additionally, the Location Manager must be able to time this very precisely. A PC with a commercial wireless LAN adapter currently will not be able to meet these performance requirements. However, specialized hardware could perform this task adequately. The Local positioning system is able to determine distances within a few meters by measuring the round trip latency for signal propagation [10].

If the participants of a Location –proving system find the cost of deployment to be prohibitive they may choose a system based on our alternative definition of network visibility. For many applications the risk of a proxy attack might be worth the benefit of easier deployment.

6. Conclusion

6a. Our distance bounding scheme aims at the providing security and privacy assurance to mobile user's proofs for their past location visits. Our distance bounding scheme relies on mobile devices in vicinity to mutually generate location proofs. Integrity and non transfer ability of location proofs and location privacy of users are the main design goals of distance bounding scheme.

We have specifically dealt with two collusion scenarios P-P collusion and P-W collusion. To protect against P-P collusions, we integrated the Bussard-Bagg a distance bounding protocol into the design of our distance bounding scheme. To detect P-W collusion, we proposed an entropy based trust model to evaluate the trust level of claims of the past location visits. Our security analysis shows the distance bounding scheme achieves the security and privacy objectives. Our implementation on android smart phones indicates that low computational and storage resources are required to execute distance bounding scheme. Wide imitation results show that our trust model is able to attain a high balanced correctness with appropriate choices of system parameters.

Future Enhancement

6b. we introduced the in region verification problem. Then we designed a provably secure lightweight protocol to address it named our framework. The frame work does not require cryptography time synchronization or any prior agreement between the prove and verifier making it suitable for low cost devices such as those in sensor networks. It is strong against a hateful challenger with unbounded compute power; the security rests on physical properties of sound and RF signal propagation. We showed that for a reasonable scenario, coverage of 80-90%could be easily achieved the framework could generate in region verification for 80-90% of legitimate location claims. Consequently, we expect our frame work to be a useful contribution in contexts where physical presence is used for access control.

Reference:

1. J.CollinsB.Hofman-Wellenhof, H.Lichtenegger Global Positioning System: Theory and Practice. Springer- Verlag, fourth Edition edition, 1997.
2. W. Luo and U. Hengartner, “Everyplace: A privacy-aware location proof architecture,” in Proc. ACM GIS, 2010, pp. 23–32.
3. Gabber, Gibbons, Matias, and Mayer. How to make personalized web browsing simple, secure, and anonymous. In FC International conference on Financial Cryptography. LNCS, Springervelag, 1997.
4. Eran Gabber and Avishai Wool. On location restricted services. IEEE network, November/December 1999.
5. R.Hasan and R. Burns, “Where have you been? Secure location provenance for mobile devices,”CoRR2011.
6. BI Inc. web site at http://www.bi.com.
7. Y. Desmedt, “Major security problems with the ‘unforgivable’ (feige)-fiat-Shamir proofs of identity and how to overcome them,” in Proc. Security Communication, 1988, pp. 15–17.
8. Legion of the Bouncy Castle Web site at http://www.bouncycastle.org.
9. The Anonymizer. Web site at http://www.anonymizer.com
10. Jay Werb and Colin Lanzl. Designing a positioning system for finding things and people indoors. IEEE Spectrum, 35(9):71-78, September 1998.