Secured Location Proof Sharing Mechanism for Android Application

Aavani N[1], Vijesh Mundokalam[2],Vikas D,Thombre[3]
[1, 2, 3]*Dept of Computer Engineering,*
[1, 3]*SKN Sinhgad Institute of Technology & Science, Savitribai Phule Pune University, Lonavala, Maharashtra, India*
[2] *Gharda Institute of Technology, University of Mumbai, Lavel, Maharashtra, India*

ABSTRACT

Recently many applications provide location service interface to user. STAMP relies on mobile devices in vicinity to mutually generate location proofs or uses wireless APs to generate location proofs. Integrity and non-transferability of location proofs and location privacy of users are the main design goals of STAMP. In order to ensure that a device is who it says it is, every device must have an encrypted Pseudonym. Because probes are used to discover their neighbors, a neighbor can check a private key it receives against the public key for the corresponding physical identity (MAC address) of the device it is trying to authenticate. At application layer, communication of resource constrained devices is expected to use constrained application protocol (CoAP).Communication security is an important aspect of IoT environment. The Internet of Things (IoT) is next generation technology that is intended to improve and optimize daily life by operating intelligent sensors and smart objects together. It is extended to use another authentication and access control system like Kerberos along with the CoAP protocol. Optimized version of ECDSA is implemented within smart things which provide efficient privacy.

Keyword: *Privacy, Location Proof, STAMP, CoAP.*

1. INTRODUCTION

Location based services have become very popular in recent years. In many scenarios, access control, authentication, and other important decisions can be made based on a user's current and past physical locations. Today's location-based services solely rely on users' devices to determine their location, e.g., using GPS.LBS requires location proof, attesting the position of a user at a specific moment in time. Location history is important for all LBS applications.

A location proof is a piece of data that certifies a geographical location. Access points (APs) embed their geographical location in location proofs, which are then transmitted to designated recipient devices. A location proof has five fields: an issuer, a recipient, a timestamp, a geographical location, and a digital signature. We use latitude and longitude coordinates to specify a geographical location. We use public keys to represent the identities of the issuer and the recipient present in the proof. Finally, the digital signature covers all the fields of a location proof except the AP's public key. The recipient uses the AP's public key to verify the integrity of the location proof.But here it recommends STP proof. Because it indicates that such a proof is intended for past location visits with both spatial and temporal information.

Location information is highly sensitive personal data. Malicious users can attempt to cheat by modifying location proofs. To prevent these types of attacks, it relies on digital signatures to protect the integrity of the proofs created by the APs. The server validates that the time intervals on each activity proof do not overlap before summing them up. Note that as the server does not allow users to declare activities that overlap in time, a user cannot use twice a collected distance proof. Users could also try to claim other user's location proofs as their own. However, the server can detect such cheating attempts by checking that the pseudonyms included in the proofs belong to the user claiming them. A malicious user could also ask other users to collect proofs on her behalf by sharing her pseudonyms. To discourage such attempts, pseudonyms include sensitive information. Also note that pseudonyms are sent encrypted to the APs with their group public key to prevent eavesdropping by other parties and that our threat model assumes honest-but-curious APs(i.e., they do not abuse the information on the pseudonyms).

Using a centralized time stamping service is not scalable, and also becomes a single point of failure. To make location provenance trustworthy, we must ensure the integrity of the chronological order of the location proofs and prevent collusion attacks that create false history. At the same time, we need to balance the tradeoff between the need to verify location history versus user privacy.

1.1 Constrained Application Protocol (CoAP)

CoAP is designed to meet specific requirements such as simplicity and low overhead in resource-constrained environments. Security is particularly important for the Things as they are connected to the untrusted Internet.

CoAP is a web protocol that runs over the unreliable UDP protocol and is designed primarily for the IoT. CoAP is a variant of the most used synchronous web protocol, HTTP ,and is tailored for constrained devices and machine-to machine communication. CoAP is a relatively simple request and response protocol providing both reliable and unreliable forms of communication. A CoAP-enabled device may be acting in a client role, a server role, or both, or sending non conformable messages without response. The reasons that a new protocol is defined for constrained IP networks, instead of simply reusing HTTP, is to greatly reduce overhead in implementation complexity (code size) and to reduce the bandwidth requirements. Such data reduction also helps to increase reliability (by reducing link layer fragmentation) and reduce latency.

Contributions: The contributions of this paper are as follows:

1. We introduce a witness-endorsed collusion-resistant scheme for generation of location proofs.
2. We present a scheme for designing private location proofs that allows users to reveal their location history only at the desired granularity.
3. 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.
4. We evaluate the performance of our system on Android-based mobile phones.
5. We can correctly resist and detect user collusion attacks in real time compared with that of previous work.
6. We reduce the overhead to generate a location proof compared with that in previous work.
7. We achieve a higher level of user location privacy compared with that of previous work.

2. RELATED WORK

With the development of wireless technology, the user can enjoy the location services easily. Some companies like Google provide users a lot of the Map interfaces, we can display the location result through the Google Map interface. Android also provide a rich API, many of them can be integrated. All of these will make our work become easy. So it is feasible for a user to collect the information from witness to verify their location.

The proposed scheme can prevent location cheating attack. Firstly, In order to check the authentication of user's claimed location, we need to calculate the distance between that location provided by the user and that in every location proof. The location contained in the location proof cannot be modified or replaced by malicious users because it is signed by the witness with his private key. So a malicious user can't fake his location without being detected by the LSP. Secondly, if a malicious user, who possess a set of location proof collected from witnesses in location A sometime ago, claims that he is now in location A while in fact he is in other location B by replaying the location proof, this attack will not achieved because the location proof contains a timestamp, which is signed by the witness with his private key and also cannot be modified or replaced by malicious users. If the timestamp is not valid, the location service provider can detect it. Therefore, the proposed scheme can prevent location cheating attack.

Location privacy is an important issue when it comes to tracking the location provenance. We need to ensure the privacy of location information. While a user wants to reveal her location to the verifier, she should not be required to reveal all of her locations unless she wants to do so. A verifier should only be able to access location information that the user has authorized the verifier to do. To allow this, any scheme to provide location provenance should allow revealing any subset of location history to the verifier. At the same time, the verifier needs to be able to verify the order of the user's location when given any arbitrary subsequence of the user's location provenance. The next privacy issue arises from the privacy of users visiting a location. While the user might need endorsement from other

users in finding its location, such witnesses should not be forced to reveal their identities. Witnesses should be able to anonymously endorse location information of other users. Finally, when a user asks for location information or an endorsement from a location, the location should not learn anything about the previous location of the user.

Secure CoAP is the basic need of resource constrained devices in real IoT environment. DTLS is the standard protocol to enable secure CoAP (CoAPs). Compressed DTLS for CoAPs is efficient in energy consumption of nodes, memory requirement and network response. This system can be deployed in real world IoT environment containing smart sensors, constrained devices and smartphones etc with real time application. Such deployment helps to deeply study and evaluate significance of this system with confidential application.

3. SYSTEM MODEL

We assume that users carry mobile devices capable of communicating with other devices and locations over Wi-Fi. A peer discovery mechanism for discovering nearby witness is required and preferably provided by underlying communication technology. The proof generation system of prover is presented a list of available witnesses. Devices can also check the presence of other devices, or the devices advertise their presence to neighboring devices. Communication between a device and the location authority or another device happens over wireless channels. Each user can act as a prover 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 user's personal devices.

There are mainly four entities:
Witness: a neighboring node that agrees to provide location proof for the prover.
Prover: Server that stores all location data, in Pseudonym form to ensure security of data.
Certificate Authority: The third party server that maps Pseudonyms to real names.
Verifier: The service that needs to verify the Prover's location.

Fig-1: System Architecture

Authentication Process (AuP)
Public login service is created on CoAP NAS by using Kerberos.

- Ticket is unique per user and service.
- Ticket has particular timeout, after that it becomes invalid.
- Ticket length could be configurable according to the requirement.

Access Control Process (AcP)

- Verifies and Checks Validity of ticket granting ticket of AuP.
- Ticket granting service is then granted for authentic requests.
- Data integrity is met by ECDSA encryption scheme which does not increase normal size of packet too much and improves the privacy.

3.1 Threat model

An adversary should not be able to create a location proof for a location that the user has not visited. Also, even if the user has visited the location, an adversary should not be able to create a proof for a different (local) time than the actual time of visit. A false location proof is one that attests to the user's presence in a location not visited by the user, or the presence at a different time than the real time of visit. The order information is also critical and must be secured.

Internal: Attacker has internal control of a device, and access to private information, as well as the ability to collude with similar devices.

Passive: Attacker cannot perform active channel jamming, mobile worm attacks or other, denial-of- service attacks.

Global: The adversary can monitor, eavesdrop, and analyze all the traffic in its neighboring area, or even monitor all the traffic around the server.

Table 1: Threat model

Description	**Threat/Attack**
Everyone is	No collusion.
Malicious user	False location proofs, reordering, denial or presence, proof switching.
Malicious	Denial of service, implication.
Malicious	False endorsement, privacy.
User and	False location proofs.
User and	False endorsement.
Location and	Implication.
Everyone	False proofs.

4. SECURITY GOALS

Our security goals are also consistent with those of STAMP, which are to create authentic location proofs and protect user location privacy. To create authentic location proofs, provers are required to honestly create and submit location proofs, which represent their specific location information. This implies that a prover can neither create a location proof with an arbitrary location or time, nor claim another legitimate user, final proof as his.

Protecting user location privacy has different connotations for different entities:

— Location privacy among prover and wit-nesses. As a proof is created in a P2P manner, if a prover and witness directly use their real identities to interactively create the proof their identities would be known to each other. Then their locations will be revealed. Thus, both prover and witness should hide their real identity when they interactively create the proof.

— Location privacy against venfier. Since a prover's precise location and time will be embedded in a proof, the verifier should not lean the prover's and his witnesses' precise location from the prover, proof

— Location privacy against CA. As the CA knows each prover and his witnesses' identity information the CA is not allowed to obtain a prover, location information from his final proof.

5. SECURITY ANALYSIS

Correctness: The completeness property is trivial. Once an LP is received by the prover, he can verify that the spatio-temporal information contained within it is valid. The spatial and temporal soundness are ensured because revealing a geolocated context that does not match the one contained in the LP will be detected during the verification process. Thus, a malicious prover cannot alter the integrity of a LP and fool the verifier by claiming a different location than the one contained in the LP.

1. The request should contain the prover's current pseudonym and a random number .
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 , prover's random number, witness's current time stamp , witness's pseudonym and their shared location. 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 and random number, 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 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.

Our design for location proofs has some security properties, as follows.

1. Integrity: A location proof is signed by the access point that issued it. Thus, a proof cannot be modified by anyone other than the piece of infrastructure where it originated from.
2. Non-transferability: Once a location proof is issued, it cannot be transferred from one user to another. When requesting a proof, the user incorporates in the request a signed version of the access point's sequence number. This ensures that the user making the request is the holder of the appropriate private key that corresponds to the public key that appears in the request. When the location proof is issued, it incorporates the client's public key signed by the access point, thereby designating this client as the recipient of the location proof.Once location proofs are issued, clients can transfer them to others only by sharing their private keys. While this is possible (e.g.,collusion attacks), the feasibility and ease of such attacks are just a function of the identity scheme used by the location proofs. In some identity schemes, the cost for mounting a collusion attack is lower than others. For example, when using e-mail addresses as identities, a collusion attack requires two users to share the passwords of their e-mail accounts. Instead, when using PGP identities, a collusion attack requires the users to share their PGP identities; this sharing is likely to be detected by their circle of "friends" – others than have vetted their identities by signing them. There are other possible forms of mounting a collusion attack that do not require users to share their private keys; for example, users can collude when requesting location proofs from the infrastructure.
3. Privacy: To reduce the privacy risks, any user can choose when to ask for a location proof and when to present their location proofs to any applications. An alternate implementation is one in which the infrastructure itself monitors the mobile devices and can vouch for the location of a device without any explicit participation.
4. Eavesdropping attack: IPsec and DTLS protect the system against this type of attacks. But to increase the security, the proposed platform could generate a new ticket for each message (plus hashing it with other parameters like Message ID), thus increasing the difficulty for a malicious user to predict a valid ticket.
5. Man in the middle attack: IPsec will encrypt the data and will use time outs for each IP connection, increasing the difficulty to guess the valid password to decrypt all packets.
6. Identity Spoofing attack: The use of false IP addresses is not going to work over the IPsec layer.

6. CONCLUSION

The traditional location sharing scheme is vulnerable and not much safer. Here it resists and detect collusion attack and provide anonymity while sharing location. It ensures the authenticity and achieves integrity and non-transferability. For solving the short comings of previous work it is to set up a secure communication channel.

7. ACKNOWLEDGEMENT

The authors would like to thank SKN SITS Management and teaching staff for an enormous support in providing time and valuable suggestions to shape idea of the project.

8. REFERENCES

[1] Xinlei Wang, Amit Pande, Jindan Zhu, Prasant Mohapatra, "STAMP: Enabling Privacy-Preserving Location Proofs for Mobile Users" IEEE/ACM Transactions On Networking Vol. 11, No. 1, January 2016.

[2] H. Han et al., "Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments," in Proc. IEEE INFOCOM, Apr. 2014, pp. 727–735.

[3] I. Afyouni, C. Ray, and C. Claramunt, "Spatial models for contextaware indoor navigation systems: A survey," J. Spatial Inf. Sci., no. 4, pp. 85–123, 2014.

[4] N. Roy, H. Wang, and R. R. Choudhury, "I am a smartphone and I can tell my user's walking direction," in Proc. ACM MobiSys, 2014, pp. 329–342.

[5] Z. Zhu and G. Cao, "Towards privacy-preserving and colluding-resistance in location proof updating system," IEEE Trans. Mobile Comput.,vol. 12, no. 1, pp. 51–64, Jan. 2013.

[6] Jack Brassil, Ravi Netravaliy, Stuart Haber, PratyusaManadhata, and Prasad Rao HP Laboratories."Authenticating a Mobile Device's Location Using Voice Signatures" Jack.2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).Pages:458-465.

[7] Yawen Wei, Yong Guan. Lightweight Location Verification Algorithms for Wireless Sensor Networks. IEEE transaction on parallel and distributed systems, (VOL.24, NO.5).Pages:938-949. MAY 2013.

[8] Xudong Ni, JunzhouLuo, Boying Zhang, Jin Teng, and XiaoleBai. MPSL: "A Mobile Phone-Based Physical-SocialLocation Verification System".Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),V7405 LNCS, pp:488-499, 2012, Wireless Algorithms, Systems, and Applications - 7th International Conference, WASA 2012, Proceedings.

[9] B. Davis, H. Chen, and M. Franklin. Privacy-preserving alibi systems. In 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS, Seoul, South Korea, 2012.

[10] Y. Zhauniarovich, G. Russello, M. Conti, B. Crispo, and E. Fernandes, "MOSES: Supporting and enforcing security profiles on smartphones," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 3, pp. 211–223, May/Jun. 2014.

[11] NashadA.Safa, SaikatSarkar,ReihanehSafavi-Naini,and MajidGhaderi. Secure Localization Using Dynamic Verifiers. 16th European Symposium on Research in Computer Security.Pages: 1-20. 2011.

[12] G. Lenzini, S. Mauw, and J. Pang. Selective location blinding using hash chains. In Security Protocols Workshop, LNCS 7114, 2011.