An Entropy Based Privacy Preserving Authentication Scheme For Mobile Users

Aavani.N[1], Vijesh Mundokalam[2], Supriya Sarkar[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- *Today's location-sensitive service relies on user's mobile device to determine the current location. This allows malicious users to access a restricted resource or provide bogus alibis by cheating on their locations. However, as computation and communication capacities become ubiquitous with the large-scale adoption of smartphones by individuals, we propose to leverage on these resources to solve this issue in a collaborative and private manner. In this paper, we present a spatial-temporal provenance (STP) proof, which ensures integrity and non-transferability of the location proofs, as well as guard users against collusion by a lightweight entropy-based trust evaluation approach. When verifying the authenticity of location provided by a user, some adjacent nodes with higher credits are selected as witnesses. The location information provided by the witnesses is compared with that presented by the user to verify whether user's location is authentic or not. Through the security analysis, we prove that this scheme can effectively prevent location cheat in LBS. Our experiments show that our entropy-based trust model is able to achieve high collusion detection accuracy.*

Keywords- Location-based service, location proof, Location authentication, spatial-temporal provenance, Witness.

I. INTRODUCTION

In recent years, mobile social networking applications has been developing rapidly, many application provides location service interface for the user. Through this interface, the user can know his position and may get reward by sharing their location with location service provider. This has led to location cheating attack, where some dishonest users may provide a false position to defraud the location service provider to get reward. For example, in scenario of calling taxi, users use the call-taxi software to share their location, the taxi drivers will provide service if they got the user's sharing location information. However, during the rush hour, in order to improve the success rate of taking a taxi, a person may publish fictional position to help his friend calling a taxi. When the driver saw there are many customs in some particular position, he will give preference to these positions, which is unfair to normal users. Also, once the user fakes their location information or the location information is tampered with other malicious software, it will increase the operating costs of the driver or cause inconvenience to normal users. Therefore, location verification becomes one of the most important issues in LBS.

To counter this threat, LBS should require its users to prove their actual or past position before granting them access to resources. This notion has been formalized through the concept of location proof (LP), attesting the position of a user at a specific moment in time.

In recent years, most research in LBS has focused on location privacy, few efforts have been put into countering location cheating attacks. In [1].ZhichaoZhu, etc. proposed a location verification method based on location proof generated by witnesses. In this method, an additional dedicated server named location proof server is equipped to store the location proof from witnesses and there are no security mechanism to ensure the trust of the location proof.

In this paper, we define the past locations of a mobile user at a sequence of time points as the spatial-temporal provenance (STP) of the user, and a digital proof of user's presence at a location at a particular time as an STP proof. Here we consider the two terms interchangeable. We prefer "STP proof" because it indicates that such a proof is intended for past location visits with both spatial and temporal information.

II. RELATED WORK

Formerly we initiated our work by an idea that adjacent users are used verify the authenticity of user's location. The said model is Fig.1:

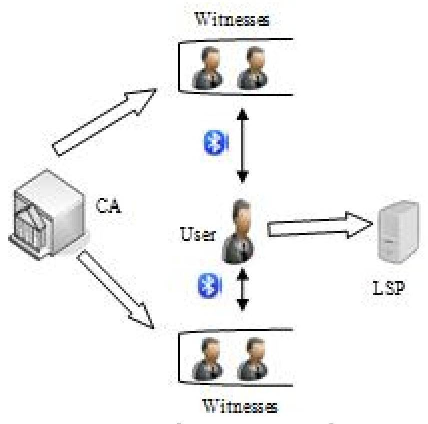

Fig.1: Authentication Scheme

- User: Who enjoys location services by a smartphone.
- Witness: Witness is an adjacent user of the user and he can generate location proof for the user.
- Location Server Provider (LSP): A server providing location service for users and store the related location data.
- Certificate Authority: As a third trust party, the CA issues certificates for every witness.

The main verification steps in our scheme as follow:

Step 1. The user claimed that he is in a certain position and he want to adjacent users help him to verify his position.
Step 2. The user open the Bluetooth of mobile device to find the adjacent user.
Step 3. Use witnesses we selected to verify the authentication . The adjacent users should generate their position coordinates .
***Step* 4.** After the digital signature, the witnesses append his own trust value and the timestamp together, send it to the user.
Step 5. When LSP received the information, it will compare the digital signature of users to determine whether the information is modified or not.

The scheme is designed to solve two security problems: 1) Fake location and 2) Replay attack. The scheme includes two processes: location proof generation and location verification. In the process of location proof generation, the user broadcasts location proof request to nearby witnesses through Bluetooth. Upon receiving the request, every witness generates a location proof containing his location information and the timestamp, and sent it back to the user. After receiving location proof from witnesses, the user select some location proof which is provided by witnesses with credit above a threshold ,and send the location proof together with his claimed location to the location service provider. In the process of location verification, the location service provider check the validity of all location proof from the user , and if all the location proof is valid, the location service provider computes the distance between the user's claimed location and the location contained in every valid location proof to check the authenticity of the claimed location.

The model suffers from remote attacks, where the user send the location proof (via Bluetooth, a drawback) from witnesses to a remote malicious user and the latter then fake his location to cheat the location service provider, We extended this scheme by STP proof architecture and improved its practicability.

We proposed a distributed STP proof architecture, i.e., mobile users obtaining STP proofs from nearby mobile peers, would be more feasible and appropriate for a wider range of applications. Fig. 2 illustrates the architecture of our system. There are few more entities, which add to this model, are:

- ***Prover:*** A prover is a mobile device which tries to obtain STP proofs at a certain location.
- ***Verifier:*** A verifier is the party that the prover wants to show one or more STP proofs to and claim his/her presence at a location at a particular time.

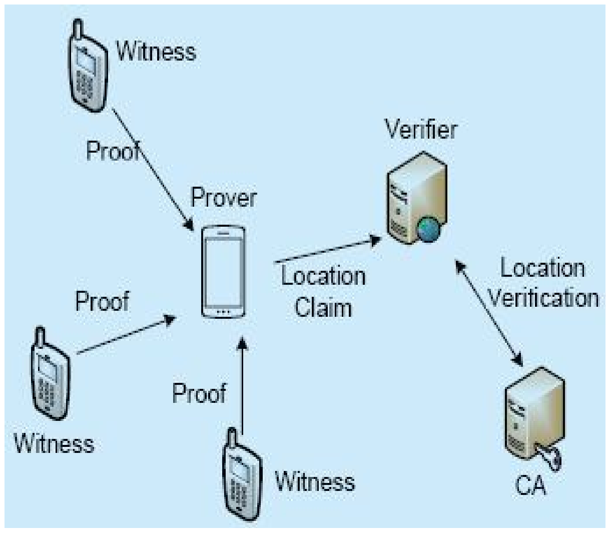

Fig.2: STP proof architecture

A prover gets a final location proof based on a few temporary location proofs created and endorsed by witnesses around him in a P2P manner,and claims the final proof to the verifier. When there are multiple witnesses willing to cooperate, the prover initiate protocol with them sequentially. STP claims are sent to verifiers from provers via a LAN or Internet, and verifiers are assumed to have Internet connection with CA. 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 users' personal devices. There are strong 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 mobile device or private key to another party.

III. SYSTEM MODEL

The Model named as Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) aims at ensuring the integrity and non-transferability of the STP proofs, with the capability of protecting users' privacy. STAMP is based on a distributed architecture. Co-located mobile devices mutually generate and endorse STP proofs for each other, while at the same time it does not eliminate the possibility of utilizing wireless infrastructures as more trusted proof generation sources. In addition, in contrast to most of the existing schemes which require multiple trusted or semi-trusted third parties,

STAMP requires only a single semi-trusted third party which can be embedded in a Certificate Authority (CA).

Fig. 3 gives an overview of the STAMP Protocol, we design our system with an objective of protecting users' anonymity and location privacy. No parties other than verifiers could see both a user's identity and STP information (verifiers need both identity and STP information in order to perform verification and provide services). Users are given the flexibility to choose the location granularity level that is revealed to the verifier.

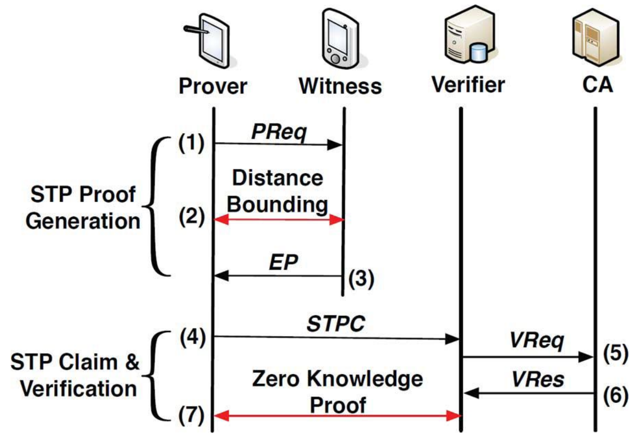

Fig.3: An illustration of STAMP protocol.

Our protocol consists of two primary phases: STP proof generation and STP claim and verification. An STP proof generation phase is the process of the prover getting an STP proof from one witness. Therefore, an STP proof collection event may consist of multiple STP proof generations. The prover finally stores the STP proofs he/she collected in the mobile device. A location proof system needs a prover to be securely localized by the party who provides proofs. A distance bounding protocol serves the purpose. A distance bounding protocol is used for a party to securely verify that another party is within a certain distance. The Bussard-Bagga protocol proposed is based on a zero-knowledge proof technique, and it allows the prover to be authenticated via a private/public key pair. We integrate the Bussard-Bagga protocol into STAMP by breaking up its execution and have the witness and verifier jointly authenticate the prover.

A. STP Proof Generation:

Suppose a prover wants to start an STP proof collection event at time , the prover first broadcasts an STP proof request (denoted as PR_{eq}) to other nearby mobile devices and waits for responses. A PR_{eq} is constructed as follows:

$PR_{eq} = C(ID_p, r_p) \mid L_1 \mid t$

where ID_p is the prover's ID, r_p is a random nonce generated by the prover for the commitment to ID_p, $L_1$and is the lowest level of the current location.

If the distance bounding stage succeeds, the witness starts creating an STP proof for the prover. The witness first creates an STP record (denoted as STPR):

$STPR= C(L_1, r^1_w)/\dots/C(L_n, r^n_w)/t$

A plaintext STP proof (denoted as) is then created as follows:

$P=C(ID_p, r_p)/STPR/z$

P is finally endorsed by the witness and encrypted using CA's public key. The endorsed STP proof (denoted as EP) is given by:

$EP=E^{K+}CA(ID_w/P/E^{K-}{}_w(H(P)))$

B. STP Claim and Verification:

At the beginning of an STP claim and verification
phase, the prover extracts the necessary data from his/her corresponding STP proof entry and creates an STP claim (denoted as STPC) as follows:

$STPC = EP_1 /\dots/EP_m/r^x_{w,1} /\dots r^x_{w,m} /IDP_p\ /r_p\ /L_x\ /t$

After receiving the prover's STPC , the verifier needs CA's assistance in verifying the STPC . The verifier now constructs a verification request (denoted as VReq) by extracting the following information from the STPC:

VReq= EP1 /…/EPm/IDp /rp

If all the EPs fail the verification or the P-W collusion detection returns a positive result, CA sends back a verification response (denoted as VRes) with a one-bit failure notification to the verifier. Otherwise, CA creates a VRes as follows and sends it back to the verifier:

$V\ Res= E^{K-}\ CA\ (STPR_1\ / \dots/\ STPR_m\ / z)$

C. P-W Collusion Detection:

If a prover colludes with a witness, it is easy for the witness to give the prover a legitimate STP proof with fake spatial-temporal information. Since the STP proof generation process is done in an opportunistic manner and we do not assume a trusted party (e.g., a location authority or a trusted witness) in this process, a P-W collusion cannot be prevented or detected with a 100% certainty. As a countermeasure against P-W collusions, we proposed an entropy-based trust model which measures the likelihood of such an attack. The trust evaluation is done by CA, which requires CA to keep track of the STP proof transaction history between any two users. A user's STP proof transactions include both the STP proofs he/she getsas a prover and the STP proofs he/she creates as a witness.

We use *entropy* to measure the collusion likelihood of a user because of its capability of capturing both the above two factors. In the STAMP system, provers meet witnesses on-the-spot. Entropy is a measure of such unpredictability. Assuming has a total number of *N* different users who had STP proof transactions with him/her, we denote this set of users as $u_1, u_2, ..u_n$. Applying the definition of entropy into our context, 's entropy is given by:

$$E_u = - \Sigma^{p(u,\, u_i)} \log p\,(u, u_i)$$

IV. DISTANCE BOUNDING PROTOCOLS

The main security requirements that have been identified in the literature.

1. A prover should be able to correctly determine its distance from an honest verifier, even when hostile attackers are present.
2. A prover should be able to determine an upper bound for its distance from even a dishonest verifier, as long as the verifier does not collude with other verifiers.
3. A prover should be able to determine an upper bound for its distance from a dishonest verifier even if it does collude.

We fix an interval I_0 that is the expected turnaround time between receiving a challenge and sending a response. Our protocol in Fig.4 proceeds in five steps, four of which involve the sending of messages.

1. The prover P generates a nonce N_P . This, and any other computations that do not involve information from the verifier, can be done in advance of P's participating in the protocol.
2. The verifier *V* requests a distance measurement. This is mainly to warn *P* that a challenge is on the way, and to let *P* know *V* 's identity.
   V sends *V;* request.
3. The verifier *V* sends a nonce as a challenge:
   V sends N_V.
4. The prover *P* sends a response, of the application of a function *F* to N_P , *P*, and *NV* . We refer to this message as the *rapid response*. The only condition that we put on *F* is that the verifier be able to verify that $F(N_V ; P;N_P)$ was constructed using *NV* , *P*, and *NP* . Examples of such functions include N_V , P,N_P ,where denotes concatenation, $N_V ; (P \oplus N_P)$, assuming that names are a distinct recognizable type, and $N_V \oplus h(P,N_P)$, where *h* is a collision-free hash function.
   P sends *F*(*NV ; P;NP*)
   The verifier, on receiving this message, calculates the time elapsed between sending the challenge and receiving the rapid response.

5. The prover sends a message authenticated with a key shared between it and the verifier. We refer to this message as the *authenticated response*. *P* sends $P; PosP ;N_P ;N_V ;MAC_{KPV} (P; PosP ;N_P ;N_V)$

where *PosP* is *P*'s position. *V* , on receiving the message, verifies the MAC. It also computes *F* from the values it receives in the authenticated response, and compares it with the value it received in the rapid response. If the two are the same, and the MAC checks out, it accepts *P*'s response as valid. *V* then subtracts I_0 from the time elapsed between sending the challenge and receiving the rapid response and uses the result to calculate its distance from the prover, that is, the distance is calculated to be v ¢ $(t_2 - t_1 - I_0)/2$.

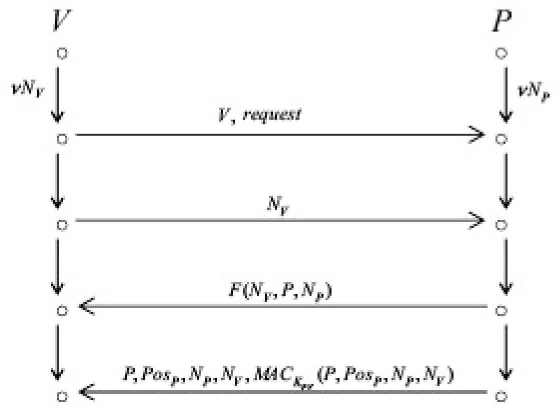

Fig.4: Distance Bounding protocol.

V. EXPERIMENTS AND RESULTS

We implemented a prototype client application on Android with Java. Our experiments are carried out on two Redmi 4G devices equipped with Octa-core Max 1.40 GHz chipset, 3 GB RAM, 32 GB ROM, GPS, and Bluetooth, and running Android OS 6.0. we use RSA key pairs as sub-keys for encryption/decryption operations. We use SHA1 as the one-way hashing function and 128-bit AES as the symmetric key encryption scheme.

Below are some screen shots of the paper:

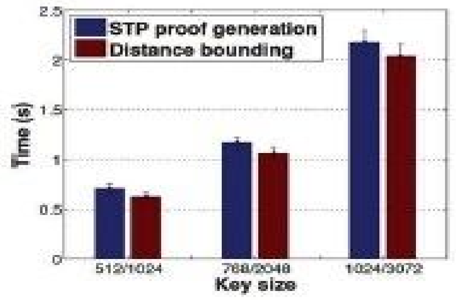

Fig.5(a) Time to generate an STP proof under different key sizes.

Fig. 5(a) shows the time needed for a prover to get an STP proof from a witness and for the portion of this process taken by the Bussard-Bagga distance bounding.

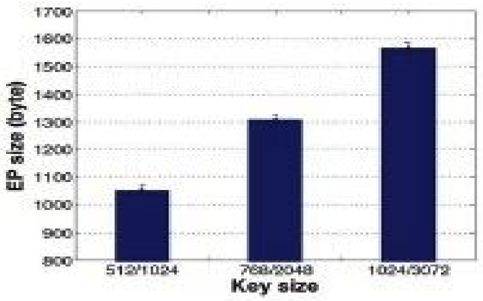

Fig.5(b). Size of EP under different key sizes

Fig. 5(b) shows the size of an that needs to be stored on a prover's mobile device. Since multiple could be received for each STP proof collection event, the size of an is the main factor that determines the storage need for an STP proof entry.

VI. SECURITY ANALYSIS

From our experimental results, we observe that under small key size settings, our scheme works efficiently in terms of both computational and storage resources. However, the computational latency could become rather long when large keys are desired. A major part of computational cost is caused by the Bussard-Bagga protocol, which is known for its expensive computation due to large amount of modular exponentiations. Other than defending against the Terrorist Fraud attack (P-Pcollusion), functionalities of STAMP do not specifically rely on the Bussard-Bagga protocol.

Our P-W collusion detection is supported by entropy-based trust evaluation, instead of complex graph algorithms like the ones used by the APPLAUS system. Therefore, each run of our P-W collusion detection only requires a number of cheap computations. It is much more efficient than APPLAUS where a few hundred seconds are needed to run a detection among a few thousands of users.
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. From (1), we can see that 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.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a location verification scheme. By selecting some appropriate adjacent users as witnesses and collecting the location proof from them, the user can confirm whether his location is authentic or not to the location services provider. Security analysis shows that scheme can prevent location cheating attack. At present, our approach can be used in densely populated environments.

we have also presented STAMP, which aims at providing security and privacy assurance to mobile users' proofs for their past location visits. STAMP relies on mobile devices in vicinity to mutually generate location proofs or uses wireless APs to generate location proofs.

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 that STAMP achieves the security and privacy objectives. Our implementation on Android smartphones indicates that low computational and storage resources are required to execute STAMP. Extensive simulation results show that our trust model is able to attain a high balanced accuracy with appropriate choices of system parameters.

REFERENCES

[1] Zhu, GuohongGao. "APPLAUS: A Privacy-Preserving Location Proof Updating System for Location. "IEEE INFOCOM 2011,pages:1889-1897.

[2] WANG et al.: "Stamp: Enabling Privacy-Preserving Location Proofs For Mobile Users" 1063-6692, 2016 IEEE.

[3] YingpeiZeng,JiannongCao,JueHong,ShigengZhang,LiXie ."Secure localization and location verification in wirelesssensor networks: a survey."Springer Science Business Media.Pages:685-701. 2010.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[4] AdnanVora, Mikhail Nesterenko."Secure Location Verification Using Radio Broadcast." IEEE Transactions on dependable and secure computing,Vol.3,No.4.pp.377-385.OCTOBER-DECEMBER 2006.

[5] Vitaly Shmatikov,Ming- siu Wang. Secure Verification of Location Claims with Simultaneous Distance Modification. Computer and Network Security.Pages:181 -195.2007.

[6] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in Proceedings of the ACM International Conference on Management of Data, 2008.

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

[8] 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.

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] Catherine Meadows, Paul Syverson, and LiWu Chang. Towards more efficient distance bounding protocols. In SecureComm 2006, August 2006.

[16] 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.

[17] R. Vishwanathan and Y. Huang, "A two-level protocol to answer private location-based queries," in IEEE International Conferences on Intelligence and Security Informatics, 2009.

[18] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in Proceedings of the ACM International Conference on Management of Data, 2008.