International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 3, Issue 10 (October '2017)

ENABLING PRIVACY-PRESERVING FOR MOBILE USERS GENERATING LOCATION PROOFS

L.Soundharya,
M.Phil Research Scholar,
Department of Computer Science,
Mahendra Arts and Science College(Autonomous),
Kalippatti ,Tamilnadu,India.

Dr.S.Kumaravel,
Head,
Department of Computer Science,
Mahendra Arts and Science College(Autonomous),
Kalippatti ,Tamilnadu,India.

Abstract: Location-based services are quickly becoming immensely popular. In addition to services based on users' current location, many potential services rely on users' location history, or their *spatial-temporal provenance*. Malicious users may lie about their spatial-temporal provenance without a carefully designed security system for users to prove their past locations. In this paper, we present the Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP) scheme. STAMP 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. STAMP ensures the integrity and non-transferability of the location proofs and protects users' privacy. A semi-trusted Certification Authority is used to distribute cryptographic keys as well as guard users against collusion by a light-weight entropy-based trust evaluation approach. Our prototype implementation on the Android platform shows that STAMP is low-cost in terms of computational and storage resources. Extensive simulation experiments show that our entropy-based trust model is able to achieve high collusion detection accuracy.

Keywords: Location Proof, Mobile Applications,STAMP,Spatial Temporal Provenance

I. INTRODUCTION

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 users' 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. In addition to users' current locations, there is an increased trend and incentive to prove/validate mobile users' past geographical locations. This opens a wide variety of new location-proof based mobile applications [1] . Saroiu *et al.* described several such potential applications in .Let us consider three examples: (1) A store wants to offer discounts to frequent customers. Customers must be able to show evidence of their repeated visits in the past to the store. (2) A company which promotes green commuting and wellness may reward their employees who walk or bike to work. The company may encourage daily walking goals of some fixed number of miles. Employees need to prove their past commuting paths to the company along with time history. This helps the company in reducing the healthcare insurance rates and move towards sustainable lifestyle. (3) On the battlefield, when a scout group is sent out to execute a mission, the commanding center may want every soldier to keep a copy of their location traces for investigation purpose after the mission. The above applications require users to be able to obtain proofs from the locations they visit. Users may then choose to present one or more of their proofs to a third-party verifier to claim their presence at a location at a particular time. 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*. Many works in literation have referred to such a proof as location proof [2][3].

In this paper, 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. Other terminologies have been also used for similar concepts, such as location claim, provenance proof , and location alibi. Today's location-based services solely rely on users' devices to determine their location, e.g., using GPS. However, it allows malicious users to fake their STP information. Therefore, we need to involve third parties in the creation of STP proofs in order to achieve the integrity of the STP proofs. This, however, opens a number of security and privacy issues. First, involving multiple parties in the generation of STP proofs may jeopardize users' location privacy. Location information is highly sensitive personal data. Knowing where a person was at a particular time, one can infer his/her personal activities, political views, health status, and launch unsolicited advertising, physical attacks or harassment. Therefore, mechanisms to preserve users' privacy and anonymity are mandatory in an STP proof system. Second, authenticity of STP proofs should be one of the main design goals in order to achieve integrity and non-transferability of STP proofs. Moreover, it is possible that multiple parties collude and create fake STP proofs. Therefore, careful thought must be given to the countermeasures against collusion attacks[4].

II. RELATED WORK

The notion of unforgeable location proofs was discussed by Waters *et al.* They proposed a secure scheme which a device can use to get a location proof from a location manager. However, it requires users to know the verifiers as a prior. [5]Saroiu *et al.* proposed a secure location proof mechanism, where users and wireless APs exchange their signed public keys to create timestamped location proofs. These schemes are susceptible to collusion attacks where users and wireless

APs may collude to create fake proofs. VeriPlace is a location proof architecture which 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 formation), a TTPU (Trusted Third Party for managing User information) and a CDA (Cheating Detection Authority). Each trusted entity knows either a user's identity or his/her location, but not both. VeriPlace's collusion detection works only if users request their location proofs 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 [5][6].

Hasan *et al.* proposed a scheme which relies on both location proofs from wireless APs and witness 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. All the above systems are centralized, that is, they all require central infrastructures (wireless APs) to act as the location authorities and generate location proofs. However, we want to design a framework that can also work for distributed scenario where users are far from any trusted AP [7]. In Davis *et al.*'s alibi system, their private corroborator scheme relies on mobile users within proximity to create alibi's (i.e., location proofs) for each other. The security and privacy of the system is achieved based on a cryptographic commitment scheme. However, they do not deal with any collusion attacks.

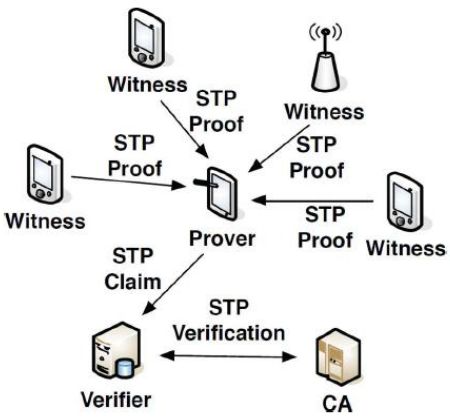

Figure 1: An illustration of system architecture

III. PROBLEM STATEMENT

Existing Model : Today's location-based services solely rely on users' devices to determine their location, e.g., using GPS. However, it allows malicious users to fake their STP information. Therefore, we need to involve third parties in the creation of STP proofs in order to achieve the integrity of the STP proofs. This, however, opens a number of security and privacy issues. Hasan *et al.*proposed a scheme which relies on both location proofs from wireless APs and witness 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. In Davis *et al.*'s alibi system, their private corroborator scheme relies on mobile users within proximity to create alibi's (i.e., location proofs) for each other.

Drawbacks:

- ❖ Most of the existing STP proof schemes rely on wireless infrastructure (e.g., 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 examples certainly cannot be obtained from wireless APs.
- ❖ Most of the existing schemes require multiple trusted or semi-trusted third parties.

Proposed System :

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*. In this paper, we propose an STP proof scheme named Spatial-Temporal provenance Assurance with Mutual Proofs (STAMP). STAMP aims at ensuring the integrity and non-transferability of the STP proofs, with the capability of protecting users' privacy. We propose an entropy-based trust model to detect the collusion scenario. A distributed STP proof generation and verification protocol (STAMP) is introduced to achieve integrity and non-transferability of STP proofs. No additional trusted third parties are required except for a semi-trusted CA.STAMP is designed to maximize users' anonymity and location privacy. Users are given the control over the location granularity of their STP proofs. An entropy-based trust model is proposed to detect users mutually generating fake proofs for each other. STAMP uses a entropy-based trust model to guard users from prover-witness collusion. This model also encourages witnesses against selfish behavior.

Advantages:

- Target a wider range of applications.
- STAMP is based on a distributed architecture.
- STAMP 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.
- 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).
- STAMP requires low computational overhead.
- A security analysis is presented to prove STAMP achieves the security and privacy objectives.

IV. METHODOLOGY

THE STAMP SCHEME

A. Preliminaries

1) Location Granularity Levels: We assume there are granularity levels for each location, which can be denoted by , where represents the finest location granularity (e.g., an exact Geo coordinate), and represents the most coarse location

granularity (e.g., a city). Hereafter, we refer to location granularity level as *location level* for short. When a location level is known, we assume it is easy to obtain a corresponding higher location level where . The semantic representation of location levels are assumed to be standardized throughout the system.

2) Cryptographic Building Blocks: STAMP uses the concept of *commitments* to ensure the privacy of provers. A commitment scheme allows one to commit to a message while keeping it hidden to others, with the ability to reveal the committed valuelater. The original message cannot be changed after it is committed to. A commitment to a message can be denoted as where is a nonce used to randomize the commitment so that the receiver cannot reconstruct , and the commitment can later be verified when the sender reveals both end .

3) Distance Bounding: 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. Different types of distance bounding protocols have been studied and proposed. A most popular category is based on *fast-bit-exchange* : one party sends a challenge bit and another party replies with a response bit and vice versa. By measuring the round-trip time between the challenge and the response, an upper bound on the distance between the two parties can be calculated. This fast-bit-exchange phase is usually repeated a number of times. One of the most challenging problems in distance bounding is the Terrorist Fraud attack, i.e., the P-P collusion scenario. The Terrorist Fraud attack is hard to defend against because a fast-bit-exchange process demands no processing delay (or at least extremely small processing delay) at the prover end between receiving a challenge bit and replying a response bit.

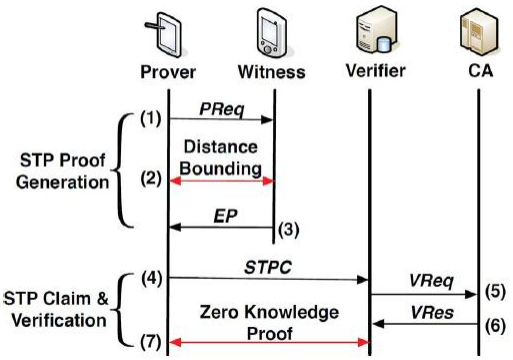

Figure 2: An illustration of STAMP protocol.

both bit values, he/she can never learn about . After the fast-bit-exchange, the location verifier de-commits and verifies the corresponding bit commitments in and (only for the received bits) by asking the prover to provide the nonces used for those commitments. In the third *zero-knowledge proof* stage, the prover convinces the verifier that he/she knows through a zero-knowledge proof. It is not possible for a user to give away the values of and , which would mean that is given away. Because of this, the protocol is not vulnerable to the Terrorist Fraud attack. In the scenario we are considering, a witness does not know the identity of a prover, we therefore cannot rely on the witness only to authenticate the prover via the zero-knowledge proof. We integrate the Bussard-Bagga protocol into STAMP by breaking up its execution and have the witness and verifier jointly authenticate the prover.

B. Protocol

Our protocol consists of two primary phases: *STP proof generation* and *STP claim and verification*. Figure 2 gives an overview of the two phases and the major communication steps involved. When a prover collects STP proofs from his/her co-located mobile devices, we say an *STP proof collection event* is started by the prover. 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. When a prover encounters a verifier (the frequency of such encounters is specific to the application scenarios) and he/she intends to make a claim about his/her past STP to the verifier, the STP claim and verification phase takes place between the prover and the verifier. A part of the verification job has to be done by CA. Therefore, communication between the verifier and CA happens in the middle of the STP claim and verification phase. In Figure 3, the two arrowed lines in red color represent the latter two stages of the Bussard-Bagga protocol. These stages require multiple interactions between the two involved parties, and thereby are represented by doubly arrowed lines. The preparation stage of the Bussard-Bagga protocol does not need to be executed for every STP proof generation and thus is not shown. Users could run the preparation stage before each STP proof collection event or pre-compute and store several sets of the bit commitments and primitives, and randomly choose one set ofthem when needed. Subsequently, we present the details of the STAMP protocol.

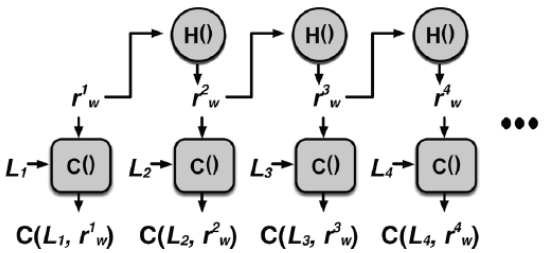

Figure 3: Construction of location level commitments.

5.1 Performance Evaluation

In this section, we analyze the security properties of the STAMP protocol and prove that the protocol can achieve our security goals.

Proposition 1: A prover cannot create a legitimate without a witness. Since users do not give away their private keys, a prover has no access to another user's private key. A plaintext

STP proof has to be signed by a legitimate witness to create a legitimate . If a prover uses his/her own private key or an illegitimate private key to create a signature for . CA will be able to detect it.

Proposition 2: Without colluding with a witness, a prover cannot create a legitimate without being present at the claimed location at the claimed time. Based on Proposition 1, a prover has to ask a witness to create a legitimate . Let us now consider two attacks: (1) a prover asserts a false location/time in a ; (2) a prover establishes a hidden communication tunnel with a proxy at the intended location and ask the proxy to send a for him/her (i.e., P-Pcollusion). When a legitimate witness receives a , he/she can easily check if in is within an acceptable range from the current time. Subsequently, the execution of the distance bounding stage enables the witness to determine if the party who sent the is within an acceptable distance. Since no signal travels faster than the speed of light, a prover who communicates with the witness from a distant location will be detected by the fastbit- exchange in the distance bounding stage. Hence, Attack can easily be detected by the witness. Based on the Bussard-Bagga protocol, the zero-knowledge proof stage is able to guarantee that a party who ran the distance bounding stages with the witness in fact has the private key corresponds to the committed in a . That means, a prover has to give his/her private key to the proxy in order to pass both the distance bounding stage with the witness and the zero-knowledge proof stage with the verifier. Assuming a user never gives away his/her private key, our protocol ensures that Attack cannot succeed.

Proposition 3: A prover cannot change the spatial and/or temporal information in an . The location levels are committed by the witness in an . The is in turn encrypted by CA's public key in an . The prover does not have CA's private key, and thus cannot decrypt an and see the location level commitments.

Proposition 4: A prover cannot use an created for another prover. By the binding property of commitments, a prover's ID is bound with the , which is in turn encrypted in an. A prover therefore cannot change the bound with an. If a prover claims to a verifier with his/her own and another prover's , CA will detect that the in the does not agree with the in the sent by the verifier. If a prover claims to a verifier with another prover's and , hoping to get services without showing his/her own identity, the verifier will detect that the prover does not possess the private key corresponding to via the zero-knowledge
proof stage.

Proposition 5: A witness cannot repudiate a legitimate created by him/her. A legitimate contains . Based on the assumption that no user gives away his/her private key, assures the non-repudiation property of an .

Proposition 6: A prover and a witness cannot find out each other's identity. During an STP proof generation process, the prover's identity is committed. Since is not known to the witness, he/she cannot de-commit and obtain . The witness's identity is enclosed in , which is encrypted by CA's public key. Since the prover does not possess CA's private key, he/she cannot decrypt and obtain . Furthermore, based on the Bussard-Bagga protocol, the distance bounding stage does not reveal the two parties' identities to each other.

VI.CONCLUSION

In this paper we have 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. Integrity and non-transferability of location proofs and location privacy of users are the main design goals of STAMP. 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-Bagga distance bounding protocol into the design of STAMP. 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.

VII.REFERENCE

[1]. S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *Proc. ACM HotMobile*, 2009, Art. no. 3.
[2]. W. Luo and U. Hengartner, "VeriPlace: A privacy-aware location proof architecture," in *Proc. ACM GIS*, 2010, pp. 23–32.
[3]. 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. 2011.
[4]. N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. ACM WiSe*, 2003, pp. 1–10.
[5]. R. Hasan and R. Burns, "Where have you been? secure location provenance for mobile devices," *CoRR* 2011.
[6]. B. Davis, H. Chen, and M. Franklin, "Privacy preserving alibi systems," in *Proc. ACM ASIACCS*, 2012, pp. 34–35.
[7]. I. Krontiris, F. Freiling, and T. Dimitriou, "Location privacy in urban sensing networks: Research challenges and directions," *IEEE WirelessCommun.*, vol. 17, no. 5, pp. 30–35, Oct. 2010.
[8]. Y. Desmedt, "Major security problems with the 'unforgeable' (feige)- fiat-shamir proofs of identity and how to overcome them," in *Proc. SecuriCom*, 1988, pp. 15–17.
[9]. L. Bussard and W. Bagga, "Distance-bounding proof of knowledge to avoid real-time attacks," in *Security and Privacy in the Age of Ubiquitous Computing*. New York, NY, USA: Springer, 2005.
[10]. B. Waters and E. Felten, "Secure, private proofs of location," Department of Computer Science, Princeton University, Princeton, NJ, USA, Tech. Rep., 2003.
[11]. X. Wang *et al.*, "STAMP: Ad hoc spatial-temporal provenance assurance for mobile users," in *Proc. IEEE ICNP*, 2013, pp. 1–10.
[12]. A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity-a proposal for

terminology," in *Designing Privacy Enhancing Technologies*. New York, NY, USA: Springer, 2001.
[13]. Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole attacks in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 370–380, Feb. 2006.
[14]. S. Halevi and S. Micali, "Practical and provably-secure commitment schemes from collision-free hashing," in *Proc. CRYPTO*, 1996, pp. 201–215.
[15]. I. Damgård, "Commitment schemes and zero-knowledge protocols," in *Proc. Lectures Data Security*, 1999, pp. 63–86.
[16]. I. Haitner and O. Reingold, "Statistically-hiding commitment from any one-way function," in *Proc. ACM Symp. Theory Comput.*, 2007, pp. 1–10.
[17]. D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *Proc. IEEE MASS*, 2005.
[18]. J. Reid, J. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing-based protocols," in *Proc. ACM ASIACCS*, 2007, pp. 204–213.
[19]. C. Kim, G. Avoine, F. Koeune, F. Standaert, and O. Pereira, "The Swiss-knife RFID distance bounding protocol," in *Proc. ICISC*, 2009, pp. 98–115.
[20]. H. Han *et al.*, "Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 727–735.