WORAL: A Witness Oriented Secure Location Provenance Framework for Mobile Devices

Ragib Hasan, Rasib Khan, Shams Zawoad, and Md Munirul Haque

Abstract—Location based services allow mobile device users to access various services based on the users' current physical location information. Path-critical applications, such as supply chain verification, require a chronological ordering of location proofs. It is a significant challenge in distributed and user-centric architectures for users to prove their presence and the path of travel in a privacy-protected and secure manner. So far, proposed schemes for secure location proofs are mostly subject to tampering, not resistant to collusion attacks, do not offer preservation of the provenance, and are not flexible enough for users to prove their provenance of location proofs. In this paper, we present WORAL, a complete ready-to-deploy framework for generating and validating witness oriented asserted location provenance records. The WORAL framework is based on the Asserted Location Proof protocol [1] and the OTIT model [2] for generating secure location provenance on the mobile devices. WORAL allows user-centric, collusion resistant, tamper-evident, privacy protected, verifiable, and provenance preserving location proofs for mobile devices. The paper presents the schematic development, feasibility of usage, comparative advantage over similar protocols, and implementation of WORAL for Android device users including a Google Glass based client for enhanced usability.

Index Terms—Location Assertion; Location Proof; Location Provenance; Location Security; Witness Endorsement; WORAL

1 Introduction

Mobile devices have enhanced the use of location-based services (LBS) using the geographical locations of the devices [3]. LBS use location tags, such as in social networks, shopping coupons, traffic alerts, and travel logs. However, LBS dependent on location proofs collected by the user have more interesting features and applications. An auditor can later verify the claim of presence with respect to the user's identity, the location in question, and the time when the user was present at that location. However, untrustworthy location reporting have implications ranging from trivial cases, such as, cheating in social-games [4], to national security issues [5].

Self-reported location presence using Global Positioning System (GPS) coordinates, cell triangulation in mobile phones, and IP address tracking are all susceptible to manipulated and false location claims [6]. Continuous tracking of users by service providers including third-party applications violates the users' privacy, allows traceable identities, and makes the users defenseless against untrusted service providers [7]. The service providers may also sell the location data of their users taking advantage of the small-text in the service agreements [8]. Buggy and insecure implementations aggravate the situation even further.

Provenance of information is important for tracing the authenticity of the data back to its source [9, 10]. The provenance of location is a crucial requirement in path critical scenarios. A valid claim of travel path needs to be verified in terms of the location provenance. The integrity of a product may be highly justified by the supply chain and the intermediate locations which the product travels through [11]. Provenance for location is a continuous process and is required to be preserved as the user travels around collecting location proofs. Unlike general data items, the sequence in which the locations are traveled needs to be preserved in chronological order within the provenance chain. As a result, location provenance portrays a greater challenge than that for general data items [2].

There have been numerous proposals for allowing user initiated location proof generation [3, 12–15]. A localization authority covering the area utilizes some secure distance-bounding mechanism to ensure the user's presence when the user requests for a location proof [16–18]. However, existing mechanisms overlook collusion attacks as well as the provenance of the location proofs. Related works thus far have not considered third-party endorsement and the chronological ordering for secure location proofs together, which makes the schemes vulnerable to collusion attacks and tampering with the order of the proofs [3, 6, 7, 12–25]. The following illustrates the practicality of a secure and asserted location provenance framework.

Bob is an engineer at a construction company. The company requires Bob to travel to the construction sites and create a daily report of the project status. Unfortunately, Bob is charged with negligence towards his job when the company suffered a major loss due to an accident. The inspection report that Bob presented was discarded for being a false document as the company claimed that Bob did not visit the construction site and the accident was a result of his negligence. In an alternate scenario, Bob collects location provenance records as he visits each of the

• *Ragib Hasan (ragib@cis.uab.edu), Rasib Khan (rasib@cis.uab.edu), Shams Zawoad (zawoad@cis.uab.edu), and Md Munirul Haque (mhaque@cis.uab.edu), SECuRE and Trustworthy computing Lab (SECRETLab), Department of Computer and Information Sciences, University of Alabama at Birmingham, AL 35294-1170, USA.*

construction sites, which are asserted by the site engineer as a witness. Therefore, Bob can then prove his regular visits and the order of visit to each of the sites based on the secure location provenance records.

In this paper, we present the Witness ORiented Asserted Location provenance (WORAL) framework. The system is based on the Asserted Location Proof (ALP) protocol [1] and incorporates the OTIT model for secure location provenance [2]. The WORAL framework is a complete suite of production-ready applications, featuring a web-based service provider, a desktop-based location authority server, an Android-based user app, a Google Glass-based client, and a desktop-based auditor.

Contributions: The contributions in the paper are as follows:

1) We have introduced a novel solution for obtaining user-centric, witness endorsed, provenance preserving, and secure location proofs for mobile devices without the requirement of having a centralized model.
2) We have presented the WORAL framework implementation; a complete ready-to-deploy suite of applications, supporting Android based devices to collect and export location proofs, including wearable peripheral devices, such as Google Glass. We developed the secure protocol for WORAL based on our earlier work on secure location proofs and augmented the protocol using secure location provenance preservation [1, 2].
3) We have also presented a discussion and a comparative analysis of similar protocols and a practicability analysis based on the suitable application areas.

The rest of the paper is organized as follows. We present the possible applications of location proof mechanisms in Section 2. We discuss related works and their limitations in Section 3. Section 4 introduces the key terminologies and the system and threat models. The WORAL framework architecture, based on ALP [1] and enhanced with the OTIT model [2] is presented in Section 5. A comparative and design analysis is included in Section 6. The implementation of the ready-to-deploy WORAL framework and its components are described in Section 7. Ongoing research for future enhancement and the conclusions are presented in Section 8 and Section 9 respectively.

2 Applications

Assertion oriented location provenance schemes can be effectively used in a variety of real-life scenarios. Our solution emphasizes the device's presence, and can be a highly applicable technology for equipment handling businesses. At present, most high end devices come with networking features and built-in memory. Hence, these expensive devices could easily be monitored for presence at their particular locations. The concept of location provenance and witnesses can also be applied to other domains, such as in preserving the integrity of supply chain information for different products and services [11].

An interesting application can be made at organizations who have traveling clientèle or employees. Travelers can collect the asserted location provenance items on their mobile devices. Later, they can utilize the proofs to simplify subsequent processes, such as, travel expense claims and itinerary management, in a secure and reliable fashion.

The whole mechanism of asserted proofing could be utilized in a reversed witness oriented application. Instead of a user presenting the proofs as evidence of presence, witnesses can present notarized records as a proof of specific users visiting a certain location. Taking the example of insurance agents, construction site inspectors, and relief workers, the presence of these people are more concerned in their respective fields of action. Witnesses at the particular sites can provide their endorsements as proof of visit for the agents on the field.

Extending the concept of locations and asserted proof of presence, social networks and such community oriented platforms have opportunities for implementing such schemes as well. A secure proof of presence with provenance preservation can be employed to form ad-hoc social networks and community networks. Therefore, a secure, automated, and non-intrusive location proof generation scheme fits perfectly as the underlying mechanism for all such LBS.

3 Related Work

Ardagna *et al.* presented a work on location-based access control (LBAC) [26], where, the requester, the access control engine, and the location service allows evaluation of LBAC policies for accessing resources and services, according to the location of the user with respect to a particular area. El Defrawy *et al.* proposed ALARM, a location-aided routing protocol, which uses current location of nodes to construct the network topology and forward data in mobile ad-hoc networks [27]. In another similar work, El Defrawy *et al.* proposed PRISM, a secure and privacy-preserving on-demand reactive location-based anonymous routing protocol for mobile ad-hoc networks [28]. Traditional Global Positioning Systems (GPS) [29] are not suitable in terms of security and indoor tracking. Gabber *et al.* [30] utilized multi-channel information from Caller-ID, GPS, cellular telephony, and satellite ranging, in a combined approach to determine the movement and location of user devices. Unfortunately, malicious entities can bypass such combinatorial schemes [3, 14]. GPS signatures [31] are not useful since they are open to spoofing attacks [14]. Bauer *et al.* have shown how localization algorithms are vulnerable to non-cryptographic attacks using a low-cost directional antenna [32]. The proposed schemes also do not consider preserving the order in which the location proofs were obtained by the user.

Ardagna *et al.* presented obfuscation-based techniques to enable different degrees of location privacy based on varying the radius of a particular area [33]. Dunne *et al.* presents an interesting approach for dealing with user privacy utilizing an adapted mediated identity based cryptography system to allow a single private key to be used with multiple public keys [34]. However, the solutions provided do not solve the problem of accountable identity ownership by

the users. Grutesar *et al.* [35] proposed a central trusted anonymity server to enable spatial and temporal cloaking of the identity for mobile devices. Secure location provenance also require verifiability and thus obfuscation strategies are not exactly applicable in this context. Hardware oriented localization techniques employ mechanisms specific to the additional functionality of devices [36–38]. Such localization techniques measure signal attenuation to verify the presence of a certain user device in the vicinity [39–41]. Other approaches use asynchronous measurement of round trip times between the user devices and access points [16, 42]. Unfortunately, location reporting mechanisms using signal attenuation can easily be manipulated by an attacker, suffer from channel noise, and has limitations with line-of-sight. Dunne *et al.* proposed a three-party architecture for location-based services utilizing an operator-oriented trusted party [34]. Such centralized architectures impose a bottleneck and complexity due to the centralized mode of operation.

Secure and unforgeable location proofs was discussed by Waters *et al.* [15]. Lenders *et al.* [43] proposed a secure geo-tagging service which allows the verification of the location and timestamp for user-generated content. However, these schemes require highly coupled entities with a monolithically centralized architecture as the cardinal block for operation. Another approach for creating secure location proofs has been described by Saroiu *et al.* [3]. Signed public keys of users and access points are applied in creating timestamped location proofs. Trusted Platform Module (TPM) and virtual machine based attestation for trusted sensor readings have been proposed by Saroiu *et al.* [44] and Gilbert *et al.* [13] respectively. Luo *et al.* [14] have presented a method to generate privacy-preserved location proofs utilizing a random nonce commitment, which is used instead of the public keys for all communications in that session. Other methods of secure localization include utilizing different channels of information, such as social networks [19], or combination of wireless medium, such as WiFi and Bluetooth [20].

Zhu *et al.* [45] proposed APPLAUS, a similar work on a collusion resistant location proof updating system using co-located Bluetooth devices. Wang *et al.* proposed STAMP for providing spatial-temporal probabilistic provenance assurance for mobile users [46]. The collusion detection for such protocols are not 100% effective and the protocol does not itself guarantee any collusion resistance. Additionally, only a limited number collusion models are considered in the threat model instead of an exhaustive analysis [1, 2].

Secure provenance have been proposed for data items, file systems, database systems, grid and distributed systems [10, 47–49]. However, none of these schemes provide solution for secure location provenance. Ananthanarayanan *et al.* [22] presented StarTrack, a framework where the sequence of a user's location and time entries are stored in tracks. While tracks are similar to location provenance chains, security issues are not considered here making tracks vulnerable to attacks by malicious users. Zugenmaier *et al.* introduced the notion of location stamps [23] for cell phones to provide proof of the location for the user at a certain time. Gonzalez-Tablas *et al.* developed the notion of Path-stamps [24] for creating a hash-chain of location proofs. Manweiler *et al.* [25] proposed the SMILE protocol, where two mutual strangers can establish shared knowledge and later prove that they have met before. However, none of these works define the requirements for secure location provenance and/or are dependent on specialized hardware features.

4 Modeling the WORAL Framework

In this section, we present the terminologies and the models for developing the WORAL framework for provenance preserving secure location proofs. In this context, we define security as ensuring the integrity and privacy of the location provenance records that has been generated at a specific location for a user.

4.1 Terminologies

We have introduced certain terminologies in the description of our models and for designing the WORAL architecture. The **Service Provider** *SP* is the trusted entity providing the secure location provenance service to mobile users, based on decentralized and certified location authorities and verified auditors. A **User** *U* is an entity who visits a location and uses a mobile device to request and store location provenance records. A **Site** *S* is a physical region with a valid address within a finite area under the coverage of one location authority. A **Location Authority** *LA* is a stationary entity, certified by the *SP*, identified using a unique identifier, and is responsible for providing location provenance records for a particular site. A **Witness** *W* is a spacio-temporally co-located mobile user who has volunteered to assert a location provenance record for the presence of another mobile device user at the given location. A **Witness List** *WL* provides the listing of all registered witnesses under the coverage of the location authority at a given time. A **Crypto-Id** *CID* is a cryptographic identity for the user (who is also a witness), used in all phases of the protocol, ensuring privacy of the entities participating in the process. A **Location Proof** *LP* is a token of evidence received by a user when visiting a specific site, and an **Asserted Proof** *AP* is a location proof *LP* asserted by a valid witness using his Crypto-ID. **Location Provenance** is the guarantee of the chronological ordering of the asserted location proofs in a tamper-evident chain of records based on a particular **Provenance Scheme** *PS*. Finally, an **Auditor** is an *SP* verified authority who is presented with a chain of asserted location proofs and confirms the legitimacy of the user's claim of presence at the particular site and the order of visits.

4.2 Witnesses and Assertions

In real-life, two parties considering each other as untrustworthy necessitates the involvement of a witness. A witness provides a notarization of a statement between two parties. The endorsed statement implies a greater truth value of the content and is then redistributed among the two parties.

We utilize the same concept to create location proofs and have the proof asserted by a co-located witness. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

this context, a witness is a spatio-temporally co-located entity with the user and the location authority. A witness will assert proofs only when willing to do so and can de-register as a witness at any time. In a commercially deployed scenario, the incentive of the witness can be based on awarded 'points' depending on valid assertions. The 'points' would add to the trust value of a witness and may be redeemed for membership benefits from the service provider. The assertions may also be used by the witness to prove co-location with the user.

4.3 Threat Model

The threat model for WORAL is based on the previously described entities and is described as follows:

- The location information within the asserted location proof corresponds to a particular identity of a user and an adversary should not be able to create a location proof for a location that the user has not visited.
- The time at which the particular user visited the given site and collected the asserted location proof should not be modifiable by an attacker to create a proof for a different (local) time than the actual time of visit.
- The identity and location privacy of users and witnesses are protected and an attacker may not create a dossier of users visiting a given location and learn the location history and identities of other users.
- The chronological ordering of the proofs should be preserved and an attacker should not be able to modify the order of proofs in the provenance records.
- The privacy of information within a proof is exposed according to the desire of the user and an attacker or auditor should not be able to view any private information not intended to be exposed by the user.
- A user intending to expose a subset of the location provenance records should not be revealing more than what is required for the desired segment of the chain.
- A malicious user should not be able to hide a temporary off-track movement from the claimed location provenance.
- A malicious user may want to overload the auditor with a high computational requirement for the secure location provenance verification process.

Next, we describe the attacker capabilities for our threat model based on the contexts, assumptions, functionality, and possible intents for each of the entities.

- Unlike previous works [3, 14, 15], we do not consider the location authorities as trustworthy. We assume that the location authorities as well as the requesting and witnessing user present at the site and participating in the proof generation protocol can all be malicious.
- Users, location authorities, and witnesses can collude with one another, driven by social, monetary, or any other form of illicit mutual benefits.
- After a proof is collected for a particular site, the user can delete or tamper with location proof and provenance records which are saved on the device.
- The location authority or the user can create a puppet witness to produce false asserted proofs or relay the assertion requests to a remote witness who is not co-located at the given site at the time of visit.
- Users, *LA*, and witnesses, each own a public/private key-pair, which has been signed by the *SP* at the time the entities register for the service, and no entity shares their private keys at any point.
- We assume that a three-way (all-party) collusion scenario does not exist as it is highly unlikely all three participants will be fraud at a given scenario.
- We expect that mobile devices are non-shareable private properties and the physical security of the phone depends on the user himself.
- Attacks such as MAC address fingerprinting are prevented via known techniques such as MAC address cloning [50].
- According to the description of the protocol, we assume the presence of at least one witness at the given site who is willing to provide an assertion.

4.4 System Model

We assume that mobile devices carried by users are capable of communicating with other devices and *LA*s over WiFi networks. The devices have local storage for storing the provenance items. The user has full access to the storage and computation of the device, can run an application on the device, and can delete, modify, or insert any content in the data stored on the device. The user, *LA*, and witness can access each others' public key from the *SP*.

The *LA* is a fixed server with higher computation and storage capability than a mobile device. A location runs a WiFi network, and the *LA* is directly connected to the network. Any user interested to receive an asserted location provenance record obtains the address of the *LA* from the site via network broadcasts. Similarly, a user can obtain the address of the location authority, and register as an interested witness. The location authority periodically updates the available witness list. When required, the location authority chooses a witness from the list at random and sends a request to the selected witness to assert a location proof.

Upon completion of a schematic communication between the entities, the user obtains a provenance preserving location proof from the *LA*, which has been asserted by a witness, and is stored on the user's device. At a later time, the user presents location proofs as a claim of presence for certain locations and the path of travel. The auditor uses the location-ID and the yielded assertion to validate the claim of presence and the chronological order of the proofs.

5 WORAL Architecture

Four entities are involved in the WORAL framework: the WORAL mobile device users (user/witness), the *LA*, auditor, and the *SP*. In the secure asserted location provenance protocol, a user *U* visits a site *S*, which is maintained by an *LA*. Additionally, there are a number of witness devices *W*, which are registered with the LA, and are willing to serve in asserting the location provenance items. The *SP* is the only centralized entity in the WORAL architecture, which is responsible to manage the accounts of the other three

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

entities, provide authentication, and distribute public keys. Figure 1 depicts the overview of the proposed architecture.

Communications between LA and mobile users are done over TCP. All messages are signed using the private key of respective entities and verified using the public key. Signature of an entity E for a message M is refereed as $S_E(M)$. An entity can receive the public key of another entity from the SP. All communications with the SP occur through the public network using REST [51] and HTTPS.

The different steps and phases of the protocol have been designed, such that, to ensure the location proof is resistant to collusion attacks and the provenance of the location proofs is preserved. Hence, we designed WORAL based on the secure location proof collection scheme presented in [1] and is enhanced using secure location provenance schemes presented in [2]. In the following subsections, we present the different components and work flows of the framework.

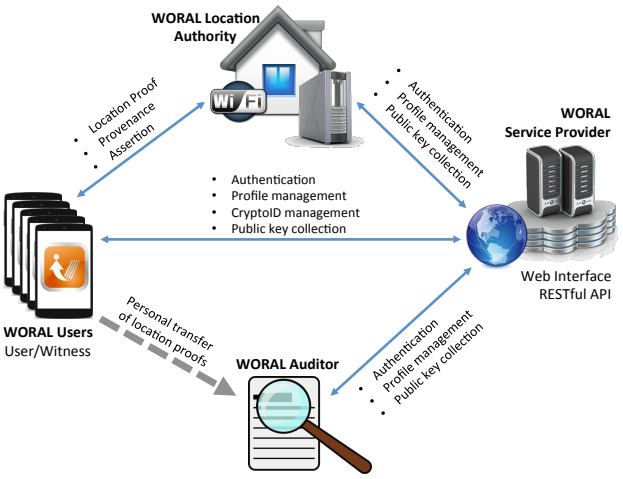


Fig. 1: Overview of WORAL Work Flow

5.1 Dependencies on Service Provider

Account Creation and Authentication: In the WORAL framework, users, witnesses, *LA*s, and auditors need to create an account with the *SP* using a unique identification criteria. Such systems can include the Social Security Number, passport number, driving license, trade license, or anything else which unambiguously identifies the person or the organization. While setting up the account, each entity needs to provide a unique username/password, which is later used as login credentials for all the entities.

As the *LA* and auditor needs to be authorized entities, there is an account verification stage for these two entities. The *SP* verifies the *LA* and auditor account requests and sends them a service code. *LA*s and auditors cannot access their accounts until the accounts are activated using the service code received from the *SP*.

CryptoID and Key Distribution: The SP is responsible for providing access to public keys in different stages of the protocol. There are two different approaches to generate the private-public keypair for *LA*s and for users (user/witness).

An LA needs to provide a human readable unique identity (location-ID) at the time of account creation. Once the account gets activated, the *SP* generates a private-public keypair, which is identified by the the location-ID. LAs need to collect the private key and store it on the local server. Upon receiving a request for the public key for a particular *LA* (location-ID), the *SP* sends the appropriate public key to the requestor.

Privacy is crucial for users (user/witness) to ensure non-traceable provenance against an attacker. In WORAL, we use a cryptographic identity (Crypto-ID) for users. The Crypto-ID hides the actual identity of user/witness within the location provenance records. A user can create multiple Crypto-IDs for WORAL and the user can chose a different one at different times on the mobile device while requesting the location proof. Hence, an external attacker cannot track the location of user/witness from a list of location provenance records. Users (user/witness) can generate a Crypto-ID on the mobile device and a private-public keypair will be created and saved for the Crypto-ID on the mobile device. The user/witness needs to upload the public key to the SP, which will be identified by the corresponding Crypto-ID. Later, a request for the public key of user/witness for a particular Crypto-ID will be served by the *SP*.

5.2 Location Authority Discovery

The user and witness need the IP address of the LA to establish a TCP connection with the *LA*. They also require the unique location-ID to access public key of the LA. The IP and identifier is made available to the user and witness through the *LA* discovery protocol using broadcast messages.

When a user or witness needs the LA's information, it broadcasts a UDP packet to a specific port requesting the information of LA. The LA always listens for new UDP broadcast packets. If the packet matches with some certain criteria (in our case, request for LA's information), the LA sends a UDP packet as a response that contains its location ID. After receiving the response sent by the LA, the user/witness can extract the identity and IP address of the LA from the received UDP packet.

5.3 Witness Registration

The LA needs to maintain a list of available co-located WORAL mobile users who are interested to serve as witnesses. The registration process is shown in Figure 2. A WORAL mobile user express his willingness to serve as a witness by sending a witness registration message $WReg$ to the LA and is defined as:

$$WReg =< CID_{\mathrm{W}}, t_{\mathrm{W}}, S_{\mathrm{W}}(CID_{\mathrm{W}}, t_{\mathrm{W}}) > \quad (1)$$

where CID_W is the Crypto-ID of the witness and t_W, is the timestamp from the witness' mobile device.

After receiving $WReg$ from the witness, the LA adds the witness information (CID_{W} and witness' IP address) to the available witness list (WL) and sends an acknowledgement message $RegAck$ to the witness.

$$RegAck =< R, t_{\mathrm{L}}, S_{\mathrm{L}}(R, t_{\mathrm{L}}) > \quad (2)$$

Here, $R \in [YES, NO]$, and t_L is the timestamp of LA.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

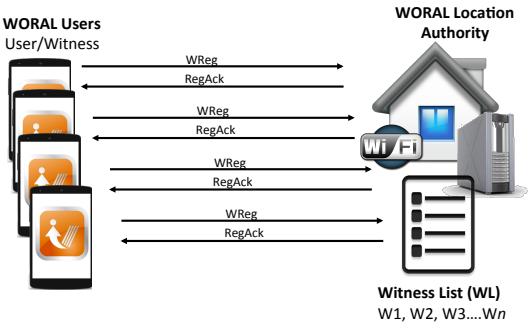


Fig. 2: Witness Registration with Location Authority

5.4 Secure Location Provenance Protocol

The sequence of interaction among the entities for creating an asserted location proof with provenance preservation is illustrated in Figure 3 and described as follows:

a) Location proof request: The user obtains the identity of the *LA* and sends a location proof request *PReq* to the *LA*, as shown in Expression 3.

$$pReq =< CID_{\mathrm{U}}, t_{\mathrm{U}}, PS, LProv_{\mathrm{cur}}, S_{\mathrm{L}}(CID_{\mathrm{U}}, t_{\mathrm{U}}, PS, LProv_{\mathrm{cur}}) > \quad (3)$$

Here, in Expression 3, CID_U is the Crypto-ID of the user U represented by the public key [3], or by anonymized identifiers [14], t_U is the timestamp from the user's mobile device, PS is the provenance scheme selected by the user, and $LProv_{cur}$ is the current head of the location provenance chain. Retrieving the current head of the provenance chain does not depend on the selected provenance scheme.

b) Location proof generation: The LA generates the location proof *LP* as shown in Expression 4 and sends the LP to the user.

$$LP =< CID_{\mathrm{U}}, L, t_{\mathrm{L}}, LProv_{\mathrm{new}}, S_{\mathrm{L}}(CID_{\mathrm{U}}, L, t_{\mathrm{L}}, LProv_{\mathrm{new}}) > \quad (4)$$

The LP includes the Crypto-ID of the user CID_U, the location-ID *L*, the local timestamp for the visit at the *LA*, and the new entry for the location provenance chain, $LProv_{\mathrm{new}}$. In general, $LProv_{\mathrm{new}}$ for all the provenance schemes will be generated using $LProv_{\mathrm{cur}}$, CID_{U}, L, and t_{L}. The provenance scheme PS selected by the user will define how the information will be used to create the new provenance entry [2]. For example, for Bloom Filter based chaining, CID_{U}, L, and t_{L} will be inserted to the existing bit array for $LProv_{\mathrm{cur}}$. The bit array with the newly added information will be the $LProv_{\mathrm{new}}$ [2]. For hash-chain, the new provenance entry is generated as $LProv_{\mathrm{new}} =< Hash(LProv_{\mathrm{cur}}, CID_{\mathrm{U}}, L, t_{\mathrm{L}}) >$ [2]. For RSA chaining, the new provenance entry is defined as $LProv_{\mathrm{new}} = LProv_{\mathrm{cur}}^{\mathrm{Hash(CID_U,\,L,\,t_L)}} \% N$ [2].

c) Proof assertion request: The LA randomly selects a witness *W* from the *WL* and then sends an assertion request *AReq* to the selected *W*, where $AReq = LP$.

d) Asserted message creation: The witness *W* verifies the information in the *AReq* message. Upon successful verification of all the information, the asserted location proof *ALP*, as shown in Expression 5, is sent to the *LA*.

$$ALP =< LP, CID_{\mathrm{W}}, CID_{\mathrm{U}}, L, h(LP), t_{\mathrm{W}}, S_{\mathrm{W}}(CID_{\mathrm{W}}, CID_{\mathrm{U}}, L, h(LP), t_{\mathrm{W}}) > \quad (5)$$

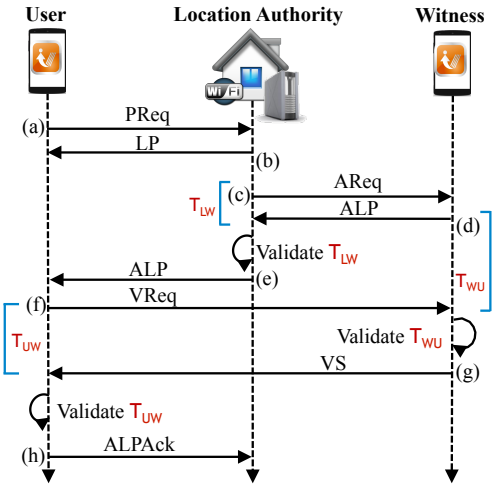


Fig. 3: Sequence Diagram for the WORAL

In Expression 5, CID_W and CID_U are the Crypto-IDs for the *W* and the *U* respectively, and t_W is the signed asserted timestamp from the mobile device used by the witnessing device. *W* also includes *h(LP)*, a hash of the *LP* to ensure the integrity of the location proof.

e) Assertion verification and relay: The LA receives and verifies the *ALP* for the assertion provided by the *W*. The *LA* also verifies the time lapse between sending an assertion request *AReq* and receiving the asserted location proof *ALP*, i.e., difference between t_L available from *ALP*, and the current time at the LA. This time difference is referred as T_{LW} in Figure 3. The LA enforces a maximum threshold for the T_{LW} to detect any proxy forwarding delay by the witness. The process of identifying the appropriate value for the T_{LW} is presented in [1]. Upon successful verification, the LA relays the *ALP* to the user *U*.

f) Verification request: Once *U* has received both the *LP* and the *ALP*, he directly communicates with *W*, and sends a verification request *VReq*, as shown in Expression 6.

$$VReq =< ALP, LP, h(ALP, LP), t_{\mathrm{u}} > \quad (6)$$

Here, *U* had already received *LP* (Expression 4) and the *ALP* (Expression 5) from the *LA*. The user then includes a signed timestamp t_u for the current time on *U*'s device, and *h(ALP,LP)*, a cryptographic hash function on both the *LP* and the *ALP*.

g) Verification response: *W* receives the *VReq* from *U* and checks to see if the assertion has been tampered or not. *W* calculates the difference between the time t_W, available in the ALP, with the current time on the witness' device. This time difference is referred as T_{WU} in Figure 3. A maximum acceptable value for the T_{WU} ensures that *U* is not trying to collect the *ALP* through a proxy. After successful verification, *W* creates a verification statement *VS*, as shown in Expression 7, and sends it to the user *U*.

$$VS =< R, t_{\mathrm{WV}}, S_{\mathrm{W}}(R, t_{\mathrm{WV}}) > \quad (7)$$

Here, in Expression 7, $R \in$ [*YES*, *NO*], and t_{WV} is the response timestamp for the *W*'s verification.

h) Location proof receipt: After receiving the *VS* from *W*, the user verifies the time difference between the time in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

$VReq\ t_u$ and the current time on the user's device when it receives *VS*. In Figure 3, this time difference is referred as T_{UW}. A maximum threshold for the T_{UW} ensures that *W* is not proxying the assertion and the verification requests. *U* then creates an acknowledgement *ALPAck* as follows:

$$AS =< LP, CID_{\mathrm{W}}, CID_{\mathrm{U}}, L, h(LP), t_{\mathrm{W}} > \quad (8)$$
$$ALPAck =< S_{\mathrm{U}}(LP, AS), h(LP, AS), t_{\mathrm{t}} > \quad (9)$$

The *ALPAck* shown in Expression 9 includes *S_U(LP,AS)*, a signature from *U*, on the *LP*, and an assertion statement *AS*, which is similar to the visible information on the *ALP*. This is then sent to the LA to be stored as a receipt for the asserted location provenance item received by *U*.

The user then stores the proof information on his device for the specific site *S* and hence, completes the secure location provenance protocol. Subsequently, the *LA* stores the receipts for the location proofs sent from the users. The *LA* maintains a publicly visible list of these tickets. At every epoch, it publishes the current state of this list along with a signature. The purpose of this publicly available list is to prevent back-dating and future-dating attacks.

5.5 Proof and Provenance Verification

When the location of *U* at a certain time is in question, *U* needs to send the location proofs stored in her device to an *SP* verified auditor. An exported proof by *U* from the mobile device contains the following items:

Plain-text information:$< CID_{\mathrm{U}}, CID_{\mathrm{L}}, Location, t_{\mathrm{U}}, PS >$
LA-Signature: $< S_{\mathrm{L}}(CID_{\mathrm{U}}, L, t_{\mathrm{L}} LProv_{\mathrm{new}}) >$
Witness-Signature: $< S_{\mathrm{W}}(CID_{\mathrm{W}}, CID_{\mathrm{U}}, L, h(LP), t_{\mathrm{W}}) >$.

Granularity of the location that appears in the exported proof is based on *U*'s selection. As *U* has control over the stored information, a malicious user can try to tamper with the plain-text information. However, even when the user has colluded with the *LA* or the witness, the user cannot the change both the signatures. While verifying the location proofs provided by the user, the auditor compares the plain-text information with the information that is signed by the LA and the *W*. Any discrepancies, with the signed information can be easily detected by an auditor.

An auditor also checks the provenance and chronological order when multiple location proofs have been presented. First, the $LProv_{\mathrm{new}}$ is extracted from each proof. Next, depending on the selected provenance scheme PS, the auditor will run the appropriate provenance verification algorithm, which are presented in [2], and verify the location provenance claimed by the user *U*.

6 Analysis

The protocol design and performance evaluation was performed and presented in details in the Asserted Location Proof paper [1]. The performance evaluation and comparison for the different provenance models were presented in OTIT [2]. This section presents a discussion on the proposed protocol including a comparison to other similar technologies.

Notation	Attack(s)
U L W	No collusion
$\bar{U}$ L W	False proof, reordering, DoS, proof switch, relay attack
U $\bar{L}$ W	DoS, implication
U L $\bar{W}$	False endorsement, privacy
U $\bar{L}\bar{W}$	Implication, relay attack, replay attack
$\bar{U}$ L $\bar{W}$	False endorsement, relay attack, Sybil attack [52]
$\bar{U}\bar{L}$ W	False location proofs, relay attack, replay attack
$\bar{U}\bar{L}\bar{W}$	False proofs.

TABLE 1: Collusion Models and Corresponding Threats

6.1 Collusion Attacks

We define the following symbols: honest and malicious users *U* and $\bar{U}$, honest and malicious location authorities *L* and $\bar{L}$, honest and malicious witnesses *W* and $\bar{W}$. The eight different combinations and corresponding possible collusion attacks are presented in Table 1. WORAL enforces mutual communication and detection of any colluded fake proof generation. A security analysis of WORAL for each collusion model is presented as follows [1].

- **[ULW]** All honest entities does not have the threat of generating false location proofs.
- **[$\bar{\mathbf{U}}$LW]** $\bar{U}$ can request for false location proofs which will not be signed or endorsed by *L* and *W*. A proxy forwarding delay for a relay attack can be detected in step *(g)* of the protocol and the endorsement will be rejected by *W*.
- **[U$\bar{\mathbf{L}}$W]** $\bar{L}$ cannot create a false proof and will never have the final receipt from the *U*. Additionally, *W* will also not assert a location proof unless it can detect *U*'s presence. *W* will not endorse a proof if the timestamp from $\bar{L}$ differs a lot from its own current system time. Any illegitimate information by the $\bar{L}$ will force *U* or the witness *W* to forfeit the WORAL protocol.
- **[UL$\bar{\mathbf{W}}$]** $\bar{W}$ alone cannot do any harm, other than denial of service (DoS) and privacy violation of *U*. However, the many-to-one Crypto-IDs of *U* does not allow $\bar{W}$ to reveal *U*'s linkable identity. A falsely asserted location proof will be discarded by *L* in step *(e)* of the WORAL protocol.
- **[U$\bar{\mathbf{L}}\bar{\mathbf{W}}$]** $\bar{L}$ and $\bar{W}$ cannot create false location proofs for *U* if *U* never participated in a proof protocol. $\bar{L}$ and $\bar{W}$ can give a user a backdated or a future dated timestamp. $\bar{L}$ can also store an old proof to launch a replay attack. However, *U* can discard the proof by not sending the final receipt in step *(h)*. A relay attack can also be identified by *U* between step *(f)* and step *(h)*.
- **[$\bar{\mathbf{U}}$L$\bar{\mathbf{W}}$]** $\bar{U}$ and $\bar{W}$ cannot create falsely asserted location proofs if *L* is honest. The *L* also doesn't allow $\bar{U}$ and $\bar{W}$ to be the same entities, and hence preventing a Sybil attack. The *SP* enforces a centralized registration system and prevents a user from having multiple profiles on the same device. *L* can also identify a relay attack with a proxy $\bar{U}$ in step *(h)*, and that of a proxy $\bar{W}$ in step *(c)*.
- **[$\bar{\mathbf{U}}\bar{\mathbf{L}}$W]** $\bar{U}$ and $\bar{L}$ can collude to create a false proof with backdated or future-dated timestamp and launch a relay or replay attack. However, *W* will not endorse a false proof and can detect a relay attack in step *(f)*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

-	U	$\bar{U}$	UW	$U\bar{W}$	$\bar{U}W$	$U\bar{W}$
L	✓	✓	✓	✓	✓	✓
$\bar{L}$	✓	✗	✓	✓	✓	✗

TABLE 2: Collusion Models and the Vulnerability Matrix

• **[$\bar{U}\bar{L}\bar{W}$]** All-way collusion is not considered in WORAL. However, backdated and futuredated attacks can still be prevented if an auditor checks the published accumulator by the *LA* for the given epoch. A post-dating attack can be possible if $\bar{L}$ does not publish the futuredated proof created falsely by $\bar{U}$, $\bar{L}$, and $\bar{W}$.

We claim that any distributed security protocol without centralized monitoring requires at least one entity to be valid. The successful completion of any security protocol is protected against the legitimate entity, who plays the role of the situational verifier. Nonetheless, an auditor may impose a stricter proof model involving asserted location proof statements from multiple closely located location authorities to verify the actual presence of the user [1].

6.2 System Vulnerability Analysis

Someone willing to share the private keys in public-key cryptography, or a general internet user willing to publicly share the secret password, does not allow any system to be secure. As a result, it is not very useful to discuss any situation where all the given entities are malicious. Increasing the number of entities in a system also increases the number of attack surfaces. Any two-entity based location proof protocol has four different collusion combinations. A two-party protocol will have atleast one combination which the system will be vulnerable to, where both the parties are malicious. As shown in Table 2, the combination of a malicious location authority $\bar{L}$ and a malicious user $\bar{U}$ will make the protocol invalid. Therefore, any such a protocol is 25% vulnerable in the best case.

Table 1 and Table 2 show the collusion models in our proposed three-party protocol. As discussed earlier, our proposed scheme is resilient to all forms of collusion, except in the case of all-way collusion [1]. Hence, all one-party and two-party attacks are prevented in the design of the secure location provenance generation scheme. Therefore, we can state that our proposed protocol is indefensible in one out of eight combinations, that is, 12.5% vulnerable in the worst case scenario.

6.3 Secure Provenance Generation

Next, we present the security lemmas and propositions for secure location provenance schemes.

Lemma 1: *A location proof is a securely generated data item for user U, which validly verifies the presence of user U at location L_i, where $i \in \{1, 2, ..., n\}$.*

Lemma 2: *A location provenance chain C is a record of location proofs for locations L_i, where $i \in \{1, 2, .., n\}$, and presence at each location L is verified using a location proof Proof(L) for that location.*

Therefore, using **Lemma 1** and **Lemma 2**, we can say that if a user U presents a provenance chain C, which has Proof(L) as one of the elements, this securely verifies the claim that the user U was present at location L. Using the above lemmas, we put forward the following propositions for secure location provenance.

Properties	HC	BC	BF	SH	MH	RC
P1	✓	✓	✓	✓	✓	✓
P2	✓	✓	✓	✓	✓	✓
P3	✓	✓	✓	✓	✓	✓
P4	✓	✓	✓	✓	✓	✓
P5	✗	✗	✗	✓	✗	✓
P6	✗	✗	✓	✗	✗	✗
P7	✗	✗	✗	✗	✗	✓
P8	✗	✓	✓	✗	✓	✗

TABLE 3: Comparison of Location Proof Provenance Approaches: Hash Chains (HC), Block-Hash Chains (BC), Bloom Filter (BF), Shadow Hash Chain (SH), Multi-Link Hashing (MH), and RSA Chaining (RC) [2]

Proposition 1 - Chronological (P1):
If U visited locations (L_{i-1}), (L_i), and (L_{i+1}) in order $(L_{i-1}) \rightarrow (L_i) \rightarrow (L_{i+1})$, the provenance chain C enters the location proofs as Proof(L_{i-1}) + Proof(L_i) + Proof(L_{i+1}).

Proposition 2 - Order Preserving (P2):
If U visited locations (L_{i-1}), (L_i), and (L_{i+1}) in order $(L_{i-1}) \rightarrow (L_i) \rightarrow (L_{i+1})$, given that Proposition 1 holds true at time t, the provenance chain C preserves the order at time $(t + \delta t)$, where δt is a positive value.

Proposition 3 - Verifiable (P3):
An auditor A can verify the order of visits $(L_{i-1}) \rightarrow (L_i) \rightarrow (L_{i+1})$ with the location provenance chain C with Proof(L_{i-1}) + Proof(L_i) + Proof(L_{i+1}) and the individual proofs Proof(L_{i-1}), Proof(L_i), and Proof(L_{i+1}).

Proposition 4 - Tamper Evident (P4):
An auditor can successfully detect the tampering and the falsely claimed order of visits for U.

Proposition 5 - Privacy Preserved (P5):
U can validate his location provenance to an auditor A with no additional information visible to the auditor.

Proposition 6 - Selective In-Sequence Privacy (P6):
U can prove Proof(L_{i-1}), Proof(L_{i+1}), without revealing Proof(L_i) from the provenance chain C to the auditor A.

Proposition 7 - Privacy Protected Chronology (P7):
An auditor can validate the provenance chain as $(L_{i-1}) \rightarrow (?) \rightarrow (L_{i+1})$ if U hides Proof(L_i).

Proposition 8 - Convenience and Derivability (P8):
The complexity of computation for verification is less than O(r), and greater or equal to O(m), where m is the number of proofs to be validated, and r is the maximum range of proofs from n proofs in the provenance chain.

Therefore, given the above security propositions for secure location provenance, Table 3 summarizes the different properties and features for each of the eight different provenance schemes supported in WORAL [2]. The formal proofs for the security of these propositions for each of the provenance schemes are presented in [2].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

Features	PLP	APPLAUS	STAMP	WORAL
Time to generate proof (sec)	$\leq$ 0.5 (10-20m)	$\leq$ 10 (10m)	3 (10-20m)	$\leq$ 1 (10-20m)
Max. distance tested (m)	N/A	10	20	40
Proof size (bits)	$\approx$ 1000	N/A	$\approx$ 1300	$\approx$ 2000
Number of entities involved	2	Multiple	Multiple	3
Malicious LA	No	Partial	Partial	Yes
Vulnerability (%)	75	$\geq$ 75	$\geq$ 75	12.5
Collusion detection rate (%)	N/A	90	90 ($\bar{U}-\bar{W}$), 100 ($\bar{U}-\bar{U}$)	100

TABLE 4: Comparative Evaluation of Protocol Characteristics: *Proactive Location Proof* (PLP) [14], *APPLAUS* [45], *STAMP* [46], and the proposed *WORAL* protocol

6.4 Evaluation of Protocol Characteristics

Different models have tried to solve the location proof problem from different perspectives. A comparison of these characteristics for different location proof models is presented in Table 4. The comparison is based on the most important characteristics for location provenance schemes and are summarized as follows:

Time to generate proof: Time to complete the whole location proof generation process is a very crucial factor in terms of usability and feasibility. The user might stay at some point for a very short period of time. Moreover, the users, and especially the witnesses might lose interest in using any such system if it takes a longer time for completion. Since the time increases with the distance among users and witnesses, we have provided the distance information along with the time to generate the proof.

Maximum distance tested: Based on the underlying technology being used in the protocol, the maximum distance supported by the system may vary. For example, the maximum possible distance covered by APPLAUS [45] is only 10 meters, since it uses Bluetooth technology. For other protocols, we have provided the maximum distance for which the system has been simulated for testing.

Proof size: Since the location proofs are being generated by mobile devices, the reasonable size of the proof is important for ensuring efficient computation and storage operations.

Number of entities involved: Increased number of entities increases the validity of the proof. But it comes with several trade-offs. Models involving more entities normally require more time. Moreover, it also increases the dimension of threats.

Malicious LA: This is a crucial consideration in terms of secure design. Most models inherently assume that the LA can never be malicious. Though location authorities are a bit more reliable than the volatile nature of the user and witness, it is still a very strong assumption, and is not considered in our design [1].

Vulnerability: We have tried to generate a vulnerability matrix for all given models. For any given model, the vulnerability percentage implies the number of scenarios where generation of invalid proofs is possible. For example, in case of the 2-entity proactive location proof protocol (PLP) [14], there are 4 possible scenarios (UW, $U\bar{W}$, $\bar{U}W$, $\bar{U}\bar{W}$). This protocol guarantees the creation of valid proofs only when both U and W are trusted (UW), and thus having 75% of vulnerability. Since STAMP [46] and APPLAUS [45] can have any number of entities, the exact number of possible scenarios is not fixed and the percentage of vulnerability will vary based on the number of entities involved. If we consider 2 entities, the percentage of vulnerability will be 75% (works for 1 out of 4 possible scenarios); considering 3 entities, it will be 87.5% (works for 1 out of 8 possible scenarios), and so on.

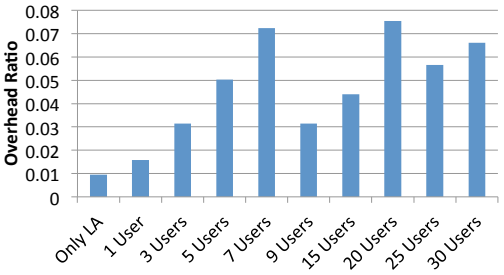


Fig. 4: Approximate (95 Percentile) System Overhead Ratio

Collusion detection rate: Theoretical proof or simulation results are used to illustrate the detection rate in case of different types of collusions, given that an attack has already been executed. In general, a higher detection rate implies a better security model. In summary, vulnerability implies the possibility of attack on a given scenario, while the collusion detection rate signifies the chances of successful detection of the given attack.

6.5 System Overhead for Location Authority

We evaluated the system overhead while running the the WORAL LA server. The LA server was deployed on a dual-core Intel Q9550 2.83GHz desktop PC with 4GB RAM and Ubuntu operating system. We performed the system performance evaluation using Sysbench[1] version 0.4.10, a cross-platform and multi-threaded benchmark tool for evaluating CPU performance.

For calculating the relative performance overhead, we first measured the CPU performance without the LA server running. Subsequently, we measured the CPU performance with the LA server running, and varying the number of consecutive proof requests made to the LA. The relative ratio for the different conditions for the approximate measurements (95 percentile) is shown in Figure 4. The average overhead ratio for all the conditions was at **0.045**, and the maximum value is seen to be at **0.075**. As it can be seen, the *LA* server accounts for a nominal overhead ratio and does not have much changes with the increase of the number of concurrent requests. The results imply that the LA is not a major resource-consuming process and can be handled in regular desktop machines. We posit that the LA can therefore be easily deployed by small businesses and shops, most of whom already own their local computer to run the surveillance system, billing system, etc.

1. https://launchpad.net/sysbench

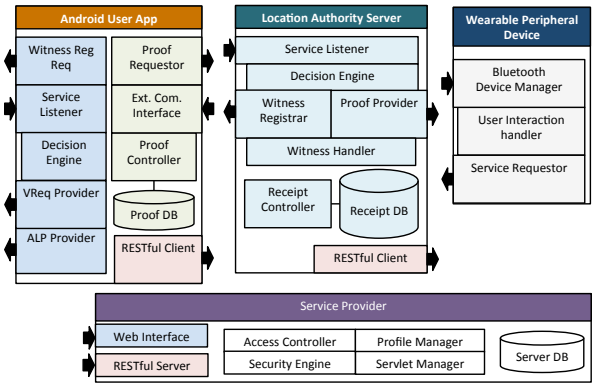


Fig. 5: Component Architecture for the WORAL Framework

7 IMPLEMENTATION

In this section, we present the implementation for a *ready-to-deploy* WORAL framework based on the proposed schematic description for the secure location provenance protocol.

7.1 Component Architecture

The component architecture of the WORAL framework is shown in Figure 5. The inward and outward arrows show the components which are in listening mode for accepting messages or are responsible for sending a message. We used the RSA (2048 bit) for generating signatures and for all encryption and decryption of the packets. Additionally, we used the SHA-2 hash function with digest sizes 256 and 512 for generating the hash messages in the protocol and for storing private information on the databases (e.g. passwords, PIN) respectively.

The user application works as both a proof-requesting user as well as a witness. Here, the 'Proof Requestor' communicates with the 'Service Listener' of the LA server to initiate the WORAL protocol. At a later stage, the 'Proof Requestor' communicates with the 'Service Listener' of another user's witness stack to request for a proof verification. Finally, the asserted location proof is stored into the local proof database. The witness stack on the user application works asynchronously. The 'Registration Requestor' module is responsible for maintaining the active registration at the particular LA. The 'Service Listener' accepts assertion requests from the LA, and verification requests from another user. The witness application also uses a 'Decision Engine', which validates all requests and communicates with the 'ALP Provider', or the 'VReq Provider' accordingly. However, the witness application does not store any of the data from the messages in the protocol. The wearable peripheral device is a client application for the WORAL Android user app. It features a 'Service Requestor' which sends and receives messages via the mobile device.

The LA server is an asynchronous server, listening to proof requests and acknowledgments from the user, as well as the registration and asserted location proof forwarding request from witnesses. The LA server has a 'Decision Engine', which validates all messages throughout the protocol. The 'Registrar' keeps a track of the currently registered witnesses. The 'Proof Provider' generates the location proofs, and also initiates the assertion request sent to the witness. Finally, the 'Receipt Controller' keeps track and stores the final asserted location provenance receipts from the user at the end of the protocol. Both the user application and LA server has RESTful clients to communicate with the SP. The SP is a web based application, with a web-UI based interface and a RESTful server, along with the necessary internal components for security, access control, profile, and servlet management.

Entities	Services
Admin	No registration required (activated via configuration script of web application), Dashboard, View used/unused service codes, Generate new service codes, View registered users/location authorities/auditors, View active inactive location authorities/auditors
User	Registration, Dashboard, View profile settings, View available crypto-IDs, Enable/Disable witness feature, Change password, Update/Save profile, Auto-sync with mobile app
Location Authority	Registration, Dashboard, Profile activation, View profile settings, Profile Activation, Private-key generated during activation, Download private-key, Change password
Auditor	Registration, Dashboard, Profile activation, View profile settings, Profile Activation, Change password

TABLE 5: WORAL Service Provider Web UI Services

7.2 WORAL Service Provider

The WORAL service provider is a web based application built on the JavaServer Pages (JSP) framework. The service provider has a web-based interface for the service provider admin, the WORAL users, location authorities, and auditors. The summary of the service offered over the web interface is presented in Table 5. The service provider also exposes a set of RESTful APIs [51] for the Android application, and the Java desktop applications for location authority and auditor. The RESTful URLs, the required parameters, description of the APIs, and the corresponding responses are summarized in Table 6. Both the web interface and the RESTful APIs are exclusively available via HTTPS. The service provider can be configured for flexible backend database servers via the configuration script.

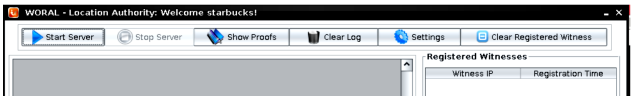
7.3 WORAL Location Authority

The LA server is a Java-based application communicating with the service provider and the user app. The application logs in and displays the service window. The control tabs on the top of the window is illustrated in Figure 6a. The operator can use the buttons to start and stop the server, and view the current list of location proof receipts. The ongoing messages for the protocol is displayed on the logging window. The LA can also use the setting tab to update the local settings, illustrated in Figure 6b. The global settings are downloaded from the SP and are not modifiable once a LA is verified and activated. The local settings are set and saved on the local machine running the LA service. Additionally, we have created a *plug-n-play* LA using Model-B Raspberry Pis[2] with 512 MB RAM, along with a customized Raspian image. The simulation test-bed for WORAL using five *plug-n-play* Raspberry Pi LAs is shown in Figure 7.

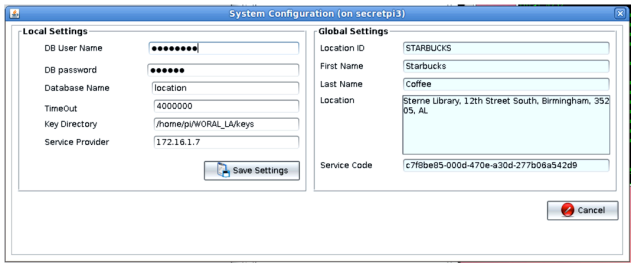
2. *http://www.raspberrypi.org/product/model-b/*

URL *https://ip:8443/woral*	**Parameters**
/Authenticate	*username, password*
Desc: User mobile app, location authority, and auditor uses to login. **Response:** Success or failure (with reason).	
/UserProfile	*username, password*
Desc: User invokes to load profile from server. **Response:** Current user profile in XML.	
/UserProfileUpdate	*username, password, isWitness, provenanceScheme*
Desc: User invokes to sync profile with server after updating on mobile device. **Response:** Success or failure (with reason).	
/LAProfile	*username, password*
Desc: LA invokes to load profile from server. **Response:** Current LA profile in XML.	
/CryptoIDList	*username, password*
Desc: User mobile app uses to download generated crypto-IDs. **Response:** Username and list of crypto-IDs in XML	
/PublicKey	*crypto-ID or location-ID*
Desc: User app, location authority, and auditor uses to collect public keys of users or location authorities. **Response:** Username, modulus, and exponent in XML.	
/PublicKeyUpload	*username, password, crypto-ID, key modulus, key exponent*
Desc: Users generate crypto-ID on the mobile app and uploads the public key. **Response:** Success or failure (with reason).	

TABLE 6: WORAL Service Provider RESTful Services



(a) Top Control Bar



(b) Settings Tab

Fig. 6: Location Authority Application Panels

7.4 WORAL Users

The WORAL Android user application is used for both requesting location proofs as well as for asserting other users' location proofs as a witness. The home screen after the user logs in is illustrated in Figure 8a. The home screen allows the user to select a crypto-ID for the current location proof request or generate new crypto-ID keys, and update/modify the settings. The settings screen for the user app is shown in Figure 8b. In the settings mode allows the user to select the background witness service features, as well as the external communication feature for wearable peripheral devices. The settings are automatically synced with the service provider. The list of currently collected proofs can be viewed as shown in Figure 8c. Additionally, the user can selectively or collectively export or delete the proofs. The exported proofs have the desired level of granularity of information

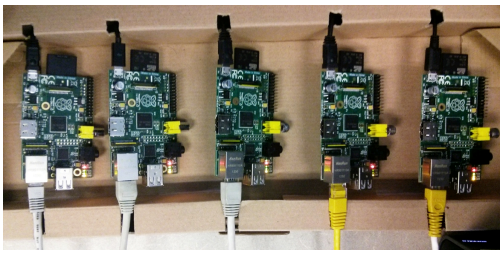
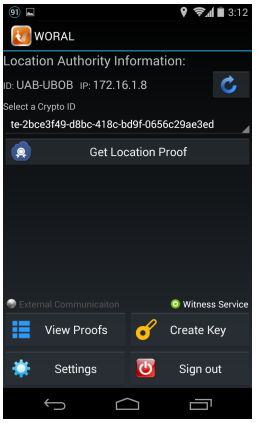
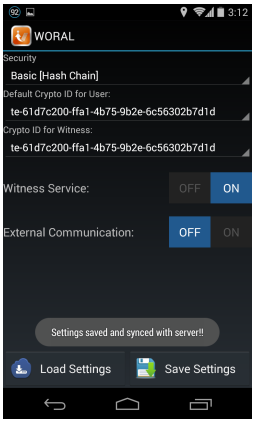


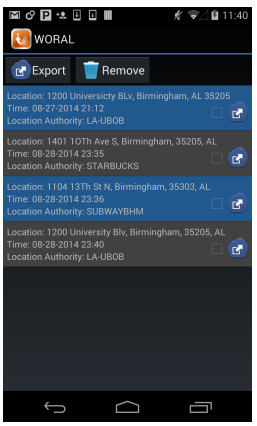
Fig. 7: *Plug-n-Play* Location Authorities using Raspberry Pi-s



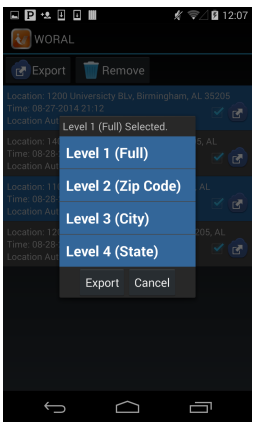
(a) Home Screen



(b) Settings



(c) Proof List



(d) Export Proofs

Fig. 8: Android User Application

as selected by the users and is shown in Figure 8d. The exported proofs is saved as a text file on the mobile device, which can then be sent personally to the auditor by the user (e.g. email, file transfer). We have tested our application on LG Nexus 4, Samsung Galaxy Nexus, Samsung Galaxy S4, Motorola XT875, HTC 1X, HTC Evo 4G, and Motorola Moto G phones with Android version 2.3 and higher.

7.5 WORAL Wearable Device Extension

Wearable peripheral devices, such as the Google Glass[3], are ubiquitous devices with networking capability. Such devices allow seamless interaction and privacy of display for the users. We extended our WORAL framework by implementing a Google Glass based interface for the WORAL Android user app. The wearable device extension

3. *http://www.google.com/glass/start/*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TETC.2015.2401394, IEEE Transactions on Emerging Topics in Computing

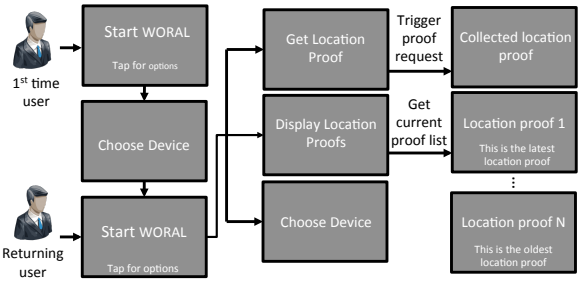


Fig. 9: Google Glass App User Flow

greatly enhances the usability of the system by allowing a user to non-intrusively interact with the WORAL framework without any physical operation on the mobile device. The glassware communicates with the WORAL app running on the paired Android phone over Bluetooth. The user can switch on the *external communication* feature on the mobile app to be able to use the WORAL Google Glass extension. The UI flow for the Google Glass is illustrated in Figure 9. Current implementation allows a user wearing the Google Glass to request for location proofs and display the list of currently available location proofs from the mobile device.

7.6 WORAL Auditor

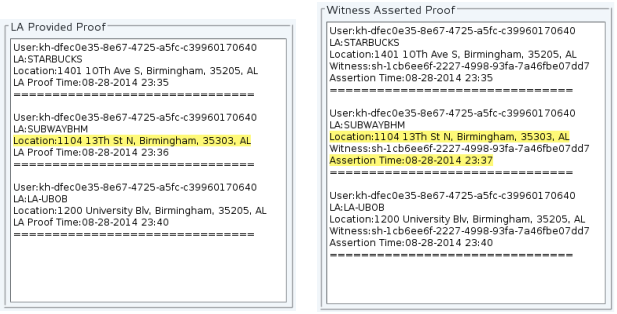
The WORAL auditor is a standalone Java desktop application communicating with the service provider. The user presents an exported proof (or list of proofs) and the auditor imports the file to verify the location proof(s) and their provenance. Two of the panels from the auditor window, for the *LA* provided information and for the witness assertion, is shown in Figure 10a and Figure 10b respectively. Any mismatched information is marked on the corresponding panels, as seen from Figures 10a and 10b. It therefore depends on the auditor to either accept of reject the location provenance claim by the user.

8 Future Work

WORAL users can obtain multiple Crypto-IDs from the *SP*, which ensures privacy by creating a many-to-one mapping of the Crypto-IDs to the original identity. Our current research includes temporal-anonymizing of the identity for the users. In this new scheme, all interactions among each other at different sites will be based on a temporal identity created by the user on run time. The temporal identity will be based on a chaotic environment and will be utilized for secure collection of a location proof only for that given site. Therefore, the temporal identity will ensure unlinkable user identities across different locations, as well as privacy of the user identity. However, the user will still be accountable for all the temporal identities and will be verifiable by the auditor at a later time.

9 Conclusion

Evolving location-based services have created a need for secure and trustworthy location provenance mechanisms. Collection and verification of location proofs and the preservation of the chronological order has significant real



(a) Location Authority Signature **(b)** Witness Assertion

Fig. 10: Auditor Service Panels

life applications. In this paper, we introduce WORAL, a ready-to-deploy framework for secure, witness-oriented, and provenance preserving location proofs. WORAL allows generating secure and tamper-evident location provenance items from a given location authority, which have been asserted by a spatio-temporally co-located witness. WORAL is based on the Asserted Location Proof protocol [1], and is enhanced with provenance preservation based on the OTIT model [2]. The WORAL framework features a web-based service provider, desktop-based location authority server, an Android-based user application including a Google Glass client for the mobile app, and an auditor application for location provenance validation.

Acknowledgments

This research was supported by a Google Faculty Research Award, the Department of Homeland Security Grant FA8750-12-2-0254, and by the National Science Foundation CA-REER Award CNS-1351038.

References

[1] R. Khan, S. Zawoad, M. Haque, and R. Hasan, "Who, When, and Where? Location Proof Assertion for Mobile Devices," in *Proc. of DBSec*. IFIP, July 2014.

[2] R. Khan, S. Zawoad, M. Haque, and R. Hasan, "OTIT: Towards secure provenance modeling for location proofs," in *Proc. of ASIACCS*. ACM, 2014.

[3] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *Proc. of HotMobile*, 2009, pp. 1–6.

[4] J. VanGrove, "Foursquare cracks down on cheaters." Online at http://mashable.com/2010/04/07/foursquare-cheaters/, April 2010.

[5] I. Maduako, "Wanna hack a drone? possible with geo-location spoofing!" Online at http://geoawesomeness.com/?p=893, July 2012.

[6] N. O. Tippenhauer, K. B. Rasmussen, C. Popper, and S. Capkun, "iPhone and iPod location spoofing: Attacks on public WLAN-based positioning systems," *SysSec Tech. Rep., ETH Zurich*, April, 2008.

[7] A. J. Blumberg and P. Eckersley, "On locational privacy, and how to avoid losing it forever," Online at https://www.eff.org/wp/locational-privacy, August 2009.

[8] J. McDermott, "Foursquare selling its location data through ad targeting firm turn," Online at http://adage.com/article/digital/foursquare-selling-data-ad-targeting-firm-turn/243398/, July 2013.

[9] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Rec.*, vol. 34, no. 3, pp. 31–36, September 2005.

[10] R. Hasan, R. Sion, and M. Winslett, "The case of the fake Picasso: Preventing history forgery with secure provenance," in *Proc. of FAST*. USENIX Association, 2009, pp. 1–12.

[11] R. Khan, M. Haque, and R. Hasan, "A secure location proof generation scheme for supply chain integrity preservation," in *Proc. of HST*. MA, USA: IEEE, 2013, pp. 446–450.

[12] B. Davis, H. Chen, and M. Franklin, "Privacy-preserving alibi systems," in *Proc. of ASIACCS*. ACM, 2012, pp. 34–35.

[13] P. Gilbert, L. P. Cox, J. Jung, and D. Wetherall, "Toward trustworthy mobile sensing," in *Proc. of HotMobile*. ACM, 2010, pp. 31–36.

[14] W. Luo and U. Hengartner, "Proving your location without giving up your privacy," in *Proc. of HotMobile*, 2010, pp. 7–12.

[15] B. R. Waters and E. W. Felten, "Secure, private proofs of location," Technical report TR-667-03, Princeton University, January 2003.

[16] S. Brands and D. Chaum, "Distance-bounding protocols," in *Proc. of EUROCRYPT*. Springer-Verlag New York, Inc., 1994, pp. 344–359.

[17] J. T. Chiang, J. J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proc. of WiSec*. ACM, 2009, pp. 181–192.

[18] K. B. Rasmussen and S. Čapkun, "Realization of RF distance bounding," in *Proc. of USENIX Security*. USENIX Association, Aug 2010.

[19] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," in *Proc. of NDSS*, Feb 2011.

[20] P. Traynor, J. Schiffman, T. La Porta, P. McDaniel, and A. Ghosh, "Constructing secure localization systems with adjustable granularity using commodity hardware," in *Proc. of GLOBECOM*, Dec 2010.

[21] J. Brassil, R. Netravali, S. Haber, P. Manadhata, and P. Rao, "Authenticating a mobile device's location using voice signatures," in *Proc. of WiMob*. IEEE, Oct 2012, pp. 458 –465.

[22] G. Ananthanarayanan, M. Haridasan, I. Mohomed, D. Terry, and C. A. Thekkath, "StarTrack: a framework for enabling track-based applications," in *Proc. of MobiSys*, Jun 2009, pp. 207–220.

[23] A. Zugenmaier, M. Kreutzer, and M. Kabatnik, "Enhancing applications with approved location stamps," in *Proc. of Intelligent Network Workshop*. IEEE, 2001, p. 140.

[24] A. I. González-Tablas, B. Ramos, and A. Ribagorda, "Path-stamps: A proposal for enhancing security of location tracking applications," in *Proc. of Ubiquitous Mobile Information and Collaboration Systems Workshop*. Citeseer, 2003.

[25] J. Manweiler, R. Scudellari, and L. P. Cox, "SMILE: encounter-based trust for mobile social services," in *Proc. of CCS*. ACM, Nov 2009, pp. 246–255.

[26] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Supporting location-based conditions in access control policies," in *Proc. of ASIACCS*. ACM, 2006, pp. 212–222.

[27] K. El Defrawy and G. Tsudik, "Alarm: Anonymous location-aided routing in suspicious manets," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1345–1358, Sept 2011.

[28] K. El Defrawy and G. Tsudik, "Privacy-preserving location-based on-demand routing in MANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 10, pp. 1926–1934, Dec 2011.

[29] P. Enge and P. Misra, "Special issue on global positioning system," *Proc. of the IEEE*, vol. 87, no. 1, pp. 3 –15, jan. 1999.

[30] E. Gabber and A. Wool, "How to prove where you are: tracking the location of customer equipment," in *Proc. of CCS*. ACM, 1998, pp. 142–149.

[31] D. E. Denning and P. F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud & Security*, vol. 1996, no. 2, pp. 12–16, 1996.

[32] K. Bauer, D. McCoy, E. Anderson, M. Breitenbach, G. Grudic, D. Grunwald, and D. Sicker, "The directional attack on wireless localization: how to spoof your location with a tin can," in *Proc. of GLOBECOM*. IEEE Press, 2009, pp. 4125–4130.

[33] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. of DBSec*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 47–60.

[34] C. R. Dunne, T. Candebat, and D. Gray, "A three-party architecture and protocol that supports users with multiple identities for use with location based services," in *Proc. of ICPS*. ACM, Jul 2008, pp. 1–10.

[35] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of MobiSys*. ACM, May 2003, pp. 31–42.

[36] S. Capkun and J. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. of INFOCOM*, vol. 3. IEEE, Mar 2005, pp. 1917–1928.

[37] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. of WiSe*. ACM, Sep 2003, pp. 1–10.

[38] S. Capkun, M. Cagalj, G. Karame, and N. Tippenhauer, "Integrity regions: Authentication through presence in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1608–1621, Nov 2010.

[39] Aruba Networks, Inc., "Dedicated air monitors? you decide." Online at http://www.arubanetworks.com/technology/tech-briefs/dedicated-air-monitors/, 2006.

[40] S. Pandey, F. Anjum, B. Kim, and P. Agrawal, "A low-cost robust localization scheme for wlan," in *Proc. of WICON*. ACM, Aug 2006, p. 17.

[41] P. Tao, A. Rudys, A. Ladd, and D. Wallach, "Wireless lan location-sensing for security applications," *Computing Reviews*, vol. 45, no. 8, pp. 489–490, 2004.

[42] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, "Pinpoint: An asynchronous time-based location determination system," in *Proc of MobiSys*. ACM, Jun 2006, pp. 165–176.

[43] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi, "Location-based trust for mobile user-generated content: applications, challenges and implementations," in *Proc. of HotMobile*. ACM, Feb 2008, pp. 60–64.

[44] S. Saroiu and A. Wolman, "I am a sensor, and i approve this message," in *Proc. of HotMobile*, 2010, pp. 37–42.

[45] Z. Zhu and G. Cao, "Toward privacy preserving and collusion resistance in a location proof updating system," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 51–64, 2013.

[46] X. Wang, J. Zhu, A. Pande, A. Raghuramu, P. Mohapatra, T. Abdelzaher, and R. Ganti, "STAMP: Ad Hoc Spatial-Temporal Provenance Assurance for Mobile Users," in *Proc. of ICNP*, Gottingen, Germany, Oct 2013.

[47] K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in *Proc. of USENIX ATC*. USENIX Association, May 2006, pp. 43–56.

[48] D. Bhagwat, L. Chiticariu, W.-C. Tan, and G. Vijayvargiya, "An annotation management system for relational databases," *The VLDB Journal*, vol. 14, pp. 373–396, 2005.

[49] I. Souilah, A. Francalanza, and V. Sassone, "A formal model of provenance in distributed systems," in *Proc. of TaPP*. USENIX Association, Feb 2009, pp. 1–11.

[50] I. Martinovic, F. Zdarsky, A. Bachorek, C. Jung, and J. Schmitt, "Phishing in the wireless: Implementation and analysis," in *Proc. of IFIP SEC*, 2007, pp. 145–156.

[51] L. Richardson and S. Ruby, *RESTful web services*. O'Reilly Media, Inc., 2008.

[52] J. Douceur, "The Sybil attack," *Peer-to-peer Systems*, pp. 251–260, 2002.

AUTHOR BIOGRAPHY

Ragib Hasan: Ragib Hasan, Ph.D., is a tenure-track Assistant Professor at the Department of Computer and Information Sciences at the University of Alabama at Birmingham. Prior to joining UAB, He received his Ph.D. and M.S. in Computer Science from the University of Illinois at Urbana Champaign in October, 2009, and December, 2005, respectively, and was an NSF/CRA Computing Innovation Fellow post-doc at the Department of Computer Science, Johns Hopkins University. Hasan has received multiple awards in his career, including the 2014 NSF CAREER Award, 2013 Google RISE Award, and 2009 NSF Computing Innovation Fellowship.

Rasib Khan: Rasib Khan, M.Sc., is a member of SECRETLab and a Ph.D. graduate research assistant at the University of Alabama at Birmingham, USA. Khan was a NordSecMob EU Erasmus Mundus Scholar, and received his MS degrees from Royal Institute of Technology (KTH), Sweden, and Aalto University, Finland in 2011.

Shams Zawoad: Shams Zawoad is a graduate research assistant in SECuRE and Trustworthy Computing Lab (SECRETLab) and a Ph.D. student at the University of Alabama at Birmingham (UAB). He received his B.Sc. in Computer Science and Engineering from Bangladesh University of Engineering and Technology in 2008.

Md Munirul Haque: Md Munirul Haque, Ph.D., is a post-doctoral fellow in SECRETLab at the University of Alabama at Birmingham. Previously, he was a member of Ubicomp lab, Marquette University, USA, from where he received his Ph.D. in 2013. He has received his B.Sc degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET), Bangladesh in 2003.