Enabling New Mobile Applications with Location Proofs

Stefan Saroiu, Alec Wolman
Microsoft Research

ABSTRACT

Location is rapidly becoming the next "killer application" as location-enabled mobile handheld devices proliferate. One class of applications that has yet-to-emerge are those in which users have an incentive to lie about their location. These applications cannot rely solely on the users' devices to discover and transmit location information because users have an incentive to cheat. Instead, such applications require their users to prove their locations. Unfortunately, today's mobile users lack a mechanism to prove their current or past locations. Consequently, these applications have yet to take off despite their potential.

This paper presents *location proofs* – a simple mechanism that enables the emergence of mobile applications that require "proof" of a user's location. A location proof is a piece of data that certifies a receiver to a geographical location. Location proofs are handed out by the wireless infrastructure (e.g., a Wi-Fi access point or a cell tower) to mobile devices. The relatively short range of the wireless radios ensures that these devices are in physical proximity to the wireless transmitter. As a result, these devices are capable of proving their current or past locations to mobile applications. In this paper, we start by describing a mechanism to implement location proofs. We then present a set of six future applications that require location proofs to enable their core functionality.

1. INTRODUCTION

Location is rapidly becoming the next "killer application" as location-enabled mobile handheld devices proliferate. Many applications and services today enable mobile devices to discover and communicate their location to a server "in the cloud"; in turn, the server uses this information to perform computation and return data relevant to the device's location. For example, in a mapping application (e.g., Google Maps), a device sends its GPS coordinates to a server which returns the relevant map information back to the client. In a 911 scenario, the device communicates its location (either through GPS or through some sort of cell tower triangulation) to a server which then dispatches assistance to the user.

One class of future location-aware applications are those in which users have an incentive to lie about their locations. These applications are unable to rely solely on the device and its software to transmit the correct location, because users have an incentive to cheat. Instead, these applications require their users to be able to *prove* their locations thereby eliminating, or at least vastly reducing, the possibility of users lying. For example, suppose a store wants to offer discounts to frequent customers; in this context, making devices aware of their location is not sufficient; instead, users must be able to show evidence of their repeated visits to the store. In another application, a content delivery server in the cloud wants to restrict what content is delivered to a particular device, depending on where users are located. These restrictions are often due to copyright laws.

While many of today's mobile users have devices capable of discovering their location, they lack a mechanism to prove their current or past locations to applications and services. The unavailability of such a mechanism has made this class of applications absent from the current landscape of mobile applications. The goal of this paper is modest – we take a step forward in facilitating the implementation and deployment of such applications. We do this by describing one possible implementation of an infrastructure that can provide location proofs, and we describe six potential applications that utilize location proofs.

This paper presents "location proofs" – a simple primitive that allows mobile devices to prove their locations to mobile applications and services. At a high-level, a location proof is a small piece of meta-data issued by a component of the wireless infrastructure (e.g., a Wi-Fi access point or a cell tower) in coordination with a mobile device. Any device can request a location proof from the infrastructure when it is within communication range; the recipient device can then transmit the proof obtained from the infrastructure to any application that wishes to verify the device's location. Location proofs are also timestamped allowing the recipient device to store them and use them later in the case when an application wants to verify a device's location at some point in the past. Finally, location proofs are signed by the infrastructure. To make use of a location proof, an application must trust the infrastructure in order to verify the location proof's signature.

To illustrate how location proofs work, let's consider the example of the content delivery server (e.g., a movie server) that wants to restrict what content it delivers to users depending on their locations. Before starting a download, the server asks the device to obtain a location proof from the cellular network. The device contacts a nearby cell tower and requests a location proof, which it then transmits to the movie server. The movie server can then verify the device's current location and then decide whether or not access to the content should be granted.

Location proofs use public keys to represent the identities of mobile devices and the infrastructure components. This allows applications to use an identity system of their choice as long as there is a method to map these identities to the associated public keys. Based on this, location proofs have several attractive security properties – they are not forgeable and they are not transferable from one device to another. In addition, location proofs have an additional privacy property: users can decide when to request them and whether to present them to applications and services. The infrastructure does not need to manage or monitor any of these mobile devices, thereby drastically reducing management costs and privacy concerns. An alternate way of implementing location proofs is a "big-brother" scheme in which the infrastructure continuously monitors the locations of mobile users. Such a design has significant privacy implications which we will discuss in-depth in Section 4.1. Although location proofs are non-transferable, one problem that stems from the nature of mobile devices is that they can easily be passed from one user to another. This means that malicious users can impersonate others just by carrying their mobile devices. In Section 5,

HotMobile 2009, February 23-24, 2009, Santa Cruz, CA, USA.

we present a high-level description of a scheme that makes such attacks much harder to mount in practice.

Location proofs are incrementally deployable – any cell tower or Wi-Fi access point can start to support them with very limited coordination with other parts of the infrastructure. This coordination is limited to the proof verifier needing a trust relationship with the proof provider (i.e., the public key). Many applications only require a small-scale deployment of infrastructure capable of handing out locations proofs. For example, a coffee store can start running a promotion promising a free drink to any customers that visited their store daily in the past week. A Wi-Fi access point that issues location proofs is a simple and cheap way of implementing such a promotion. Similarly, a teacher can offer rewards to those students who never miss a class during the semester. With location proofs, students can collect them and submit them at the end of the semester to receive their reward. Section 2 will present several such applications of location proofs and expand on their implementation.

Any wireless infrastructure component can distribute location proofs to nearby mobile devices. To accomplish this, the infrastructure component must implement a simple two-way protocol that issues location proofs. Once issued to a device, a location proof demonstrates that the device was within radio range of the infrastructure. These ranges differ depending on the type of the infrastructure, from a few hundred meters for Wi-Fi to a few kilometers for cell towers. This paper presents a design of location proofs only for Wi-Fi. We chose Wi-Fi because the standard is open and well-understood, making it easy for anyone to implement our design and use it in their mobile applications.

2. APPLICATIONS

In this section, we describe several potential applications where location proofs play a central role in enabling them. The common theme across all these applications is that they offer a reward or benefit to users located in a certain geographical location. Thus, users have an incentive to lie about their locations.

2.1 Store Discounts for Loyal Customers

Retaining customers offers many benefits to a store [7]. Loyal customers are more likely to recommend the store to others, they are more willing to try new products and to spend more money, and their feedback is often more helpful. Thus, many stores are actively looking for new ways to retain their loyal customers by providing them with discounts, coupons, or with other rewards.

One way to build a loyal customer base is to offer discounts to the customers who visit the store repeatedly or who spend a longer time in the store. With location proofs, customers' devices can gather the location proofs from inside the store; when a discount is available, each customer can prove their loyalty to the store by presenting their set of location proofs collected over time. Similarly, restaurants could offer priority seating for frequent customers. The key benefit of location proofs is that it vastly simplifies the process of keeping track of customers on behalf of the business owner.

2.2 Green Commuting

Carbon emissions are believed to be a significant cause of global warming. One of the main factors contributing to carbon emissions is car travel. In this context, companies and other organizations have started to seek ways to reduce car travel by providing incentives for employees to find alternative commuting options. For example, Microsoft has initiated a program that rewards employees who leave their cars at home and instead walk, bike, or commute by bus to work. This program currently has no checks in place – the rewards offered are not significant enough to cause rampant cheating among Microsoft employees. There is discussion to expand this program to all employers in Redmond WA. However, in our discussions with the people who run this program, we learned that the city of Redmond is skeptical about the success of deploying such a program citywide without stronger checks. To make it successful, they believe that employers must be able to verify the commuting options chosen by their employees.

We believe location proofs can provide an efficient and inexpensive implementation of such checks for green commuting. We could deploy Wi-Fi access points capable of handing out location proofs every half-a-mile along the roads of our city; our back-of-the-envelope calculation suggests that 200 such access points should be sufficient to cover most of the major roads and city buses. Commuters could collect timestamped location proofs on their way to work. Once at work, these timestamps together can provide an accurate indication of the commuter's mode of transportation. For example, a commuter presenting two location proofs collected from two electricity poles half-a-mile a part is likely to have walked if the timestamps are more than 7.5 minutes apart[1]. Note that people can still cheat in our system; for example, a person could have commuted by car instead and just take a stop in between the two poles to have a coffee. However, we believe that such a system is viable as long as most people would not regard the reward worth the inconvenience and dishonesty of cheating.

2.3 Location-Restricted Content Delivery

A recently emerging class of Web content delivery applications are those that deliver TV shows, such as Joost or Hulu. TV content is subject to complicated copyright laws that restrict their broadcast to certain countries only. To conform with these copyright laws, these websites use IP-to-Geo schemes [10] to discover the location of each viewer and to restrict their content delivery accordingly. Unfortunately, these schemes are often inaccurate and can mistakenly restrict some viewers from watching content that should be permissible under the copyright laws. With location proofs, clients can provide proofs of their locations to these websites. Additionally, these websites can save the location proofs to provide evidence about their compliance with the copyright laws to any enforcement agency.

Using location proofs also provides additional benefits over using IP-to-Geo schemes. Because the location information provided by the location proofs is much more fine grained, websites can tailor their content delivery to the respective region or geographical area of the viewer. For example, a major league sports game could provide two audio tracks, each with the commentary that is biased favoring one team over the other. The website could determine which audio track to deliver based on the viewer's location. Currently, the coarse geographical information in IP-to-geo schemes is inadequate for the needs of such an application.

2.4 Reducing Fraud on Auction Websites

A common security problem on auction websites such as eBay is account theft – attackers break into legitimate accounts and use their established reputations to commit fraud. Most often these attackers are from remote places. When a transaction occurs, buyers currently lack a way to establish that the seller is in fact present in the geographical region indicated in their profile. Such a check could increase the confidence that the seller's account has not been broken into.

Location proofs could provide one such mechanism. For example, for eBay, once the bidding is complete, the seller would be required to present a location proof that validates his geographical location to the buyer. The buyer can independently check that the location encapsulated by the proof matches the location in the buyer's profile. This can serve as additional evidence that the seller's ac-

[1]We assume that most people do not walk faster than 4 miles per hour.

count has not been compromised by a remote attacker.

2.5 Police Investigations

Many police investigations are quickly resolved by examining the alibis of the persons involved in an incident. If examining these alibis does not lead to an obvious suspect, police investigations become more lengthy and more costly. Therefore, police forces are interested in findings ways for people to be able to produce alibis quickly and cheaply.

With location proofs, people can use their mobile cell-phones to produce such alibis. On a police investigation, a person could decide whether the location proofs collected by their cell-phone could be used as an alibi. Note that this is different than the big-brother scenario in which the wireless infrastructure continuously monitors the whereabouts of their users. Location proofs let the users decide whether they want to collect the proofs in the first place and whether they want to present them as evidence.

2.6 Voter Registration

During an election, voters are often asked to provide proof of their presence in particular region, state, or country for a predetermined period of time. In the US, this is often called the "physical presence requirement". This is not only inconvenient to prove, but it is sometimes impossible for some people. To resolve these situations, there are some cases where people are allowed to take an oath in the presence of a public notary in case they lack the necessary evidence for this requirement. In other cases, the law may simply exclude such people from their right to vote. A similar presence requirement is often also needed for citizenship requirements.

Once again, location proofs can provide a simple mechanism for demonstrating the physical presence requirement. People can submit a collection of location proofs that match the geographical location requirement and the duration requirements of the physical presence test.

3. WHAT IS A LOCATION PROOF?

A location proof is a piece of data that certifies a geographical location. Access points (APs) embed their geographical location in location proofs, which are then transmitted to designated recipient devices. A location proof has five fields: an issuer, a recipient, a timestamp, a geographical location, and a digital signature. We use latitude and longitude coordinates to specify a geographical location. We use public keys to represent the identities of the issuer and the recipient present in the proof. Later in this section, we describe how location proofs can work with a variety of identity schemes, including Windows Live IDs [11], OpenID [12] logins, and email addresses. The only requirement we place on an identity scheme is the ability to map users' identities to the keys present in the proof. Finally, the digital signature covers all the fields of a location proof except the AP's public key. The recipient uses the AP's public key to verify the integrity of the location proof. We use XML for the location proof's format (see Figure 1).

3.1 Identities

Location proofs are personal and non-transferable. Thus, the description of location proofs must start with a description of what constitutes a personal identity in our scheme. Many different identity schemes could be used for location proofs. The only requirement is that these schemes can verify that a public key embedded in a location proof is uniquely mapped to one single identity. Many identity schemes (e.g., PGP [14], OpenID [4]) already have provisions for such a feature. The choice of the identity system is largely independent of the rest of the design requirements for location proofs.

```
<locproof>
 <issuer>Issuer's public key</issuer>
 <recipient>Recipient's public key</recipient>
 <timestamp>Timestamp when issued</timestamp>
 <geolocation>
  <latitude>…</latitude>
  <longitude>…</longitude>
 </geolocation>
 <signature>Location proof's signature</signature>
</locproof>
```

Figure 1: ***The XML-based format of a location proof. A location proof has an issuer, a recipient, a timestamp, a geographical location, and a digital signature. The identities of the issuer and the recipient are represented with public keys. The issuer embeds its geographical location and signs the location proof before issuing it. The signature only covers the recipient, the timestamp, and the geographical location.***

Single sign-on provider: One possibility is to use a single identity provider, such as a Windows Live ID [11] or a Google Account [5]. In this case, whoever verifies the identities (whether the wireless infrastructure or the applications) must setup a key with the single sign-on server. Once the user authenticates to the single sign-on server, the server returns a token encrypted with this key. Correctly decrypting this token allows the verifier to check the user's identity.

OpenID: OpenID [12] is a decentralized single sign-on system. Users need to register with any OpenID "identity provider", and any website can be such a provider. An OpenID is simply a URL hosted by the identity provider. The verifier of the identity must contact the provider to verify the user's identity. Because of its decentralized nature and the user's freedom to choose any provider, OpenID has better privacy properties than a single identity provider scheme.

PGP: PGP [14] uses a vetting scheme in which people sign each other's public keys. Over time, PGP creates a "Web of trust" in which people accumulate each other's signatures after verification. To verify a person's signature in PGP, people must find a chain of trust linking the person to themselves. This verification step is typically done by contacting a PGP repository that stores the "Web of trust".

E-mail addresses: Another possibility is to use e-mail addresses as identities. Users must demonstrate that they own the e-mail address they claim as their identity; websites already perform this verification today by sending an e-mail containing a URL and asking the user to click on the URL. Users must own the e-mail address to be able to perform this task. If the e-mail service does not have the capability of associating a key pair with an individual email account, then we would need an additional online service to perform this function.

Online accountable pseudonyms: Another recently proposed identity scheme with desirable privacy properties is online accountable pseudonyms [3]. These pseudonyms are anonymous allowing users to maintain their privacy. Creating such pseudonyms requires the physical presence of the user in a large social gathering, such as a large party, to protect the user's privacy. As a result, users are restricted in the number of identities they can feasibly create, which limits the possibility of Sybil attacks.

3.2 Issuing a Location Proof

Wi-Fi access points broadcast beacon frames to announce their presence. Clients receive beacons sent from nearby APs when not connected to a Wi-Fi network. Even when connected to a specific AP, clients periodically scan all channels to receive beacons from other nearby APs; this is done so the client can keep track of

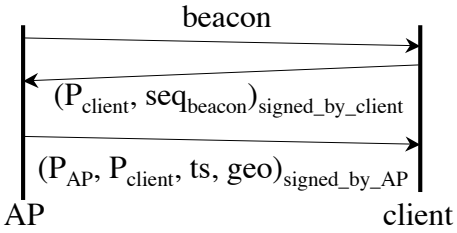

Figure 2: ***The protocol for issuing a location proof. APs send out beacons advertising their support for location proofs. A client requests a location proof by sending its public key and a signed sequence number. The AP checks the sequence number's signature and that the sequence number is current. If the request is valid, a location proof is sent back to the client.***

other available APs in case the primary AP becomes unreachable. A client does not have to transmit any data to receive a beacon; it merely needs to listen.

Any AP capable of issuing location proofs adds its geographical location to its beacons. Upon receiving a beacon, a client can decide whether to explicitly request a location proof from the respective AP. To request a proof, the client extracts the beacon's sequence number to use it in the request for the location proof. Sending back the sequence number to the AP prevents replay attacks[2] The request for a location proof contains the client's public key and the signed AP's sequence number. The client signs the sequence number to protect their integrity and to make it hard for clients to impersonate other devices. We will present a more in-depth discussion of the security property of location proofs in Section 4.

Upon receiving the request, the AP checks whether the signature is valid and whether the sequence number is a current one. Our current design accepts requests whose sequence numbers were broadcasted by the APs within the last 100 milliseconds. Although 802.11 sequence numbers repeat themselves after 4096 frames, the 100 ms time interval is sufficiently small to prevent security attacks taking advantage of sequence number wrapping, such as replay attacks. If the request is invalid, the AP drops the request silently. In case of a valid request, the AP creates a location proof with a current timestamp and designates the client as the recipient. After creating the location proof, the AP broadcasts it. The AP does not check whether the client received the location proof. Figure 2 illustrates the protocol for issuing location proofs.

3.3 Verifying a Location Proof

To present a location proof, a client must sign it and prepend its public key before transmitting it. Upon receiving the proof, an application performs three steps. First, it checks the client's signature to make sure that the location proof has not been tampered with while being transmitted. Second, the application checks the AP's digital signature that is embedded in the proof itself. This step ensures that the client has not tampered with the location proof. Finally, the application verifies that the client is indeed the recipient of the location proof. If all these steps are successful, the location proof is deemed legitimate; it is now up to the application to use this location proof. Note that the application's semantics could reject the location proof even if legitimate. For example, a location proof could be invalid because its timestamp is incorrect according to the application's semantics.

3.4 Practical Considerations

[2]A replay attack is one in which the request for a location proof is maliciously repeated by an attacker.

An area of concern in practice is that clients can perform a denial-of-service (DoS) attack by sending many requests for location proofs to access points. Upon receiving requests, access points perform cryptographic operations to verify the legitimacy of the requests. A large number of such operations can overwhelm the computational resources of APs. We mitigate these attacks by rate limiting the number of requests for location proofs that are processed by APs. For example, a rate limit of two location proof requests per second is unlikely to affect any computational performance of today's APs. At the same time, we believe that a rate limit of two requests per second is sufficient for most scenarios in which devices need to request location proofs.

Another practical consideration is making sure that APs are configured with the correct location coordinates. While it is inexpensive to provision APs with GPS to automatically determine their geolocation, most APs are located in indoor environments where GPS does not work well. One way to overcome this difficulty is to equip the AP with an additional configuration interface for administrators. To install a location proof-enabled AP, the administrator first takes the AP outdoors and runs a setup program that uses GPS to determine the AP's location. After setup, the AP instructs the administrator that it is ready to be deployed indoors. While this approach can reduce the likelihood of misconfigured APs, it introduces two additional problems. First, it introduces error because the location where the GPS reading is performed is different than the *true* AP location. Second, APs are often relocated (e.g., an AP can be sold to another owner). To handle relocation, the AP location must be re-initialized in the new location. One way to automate this process is to provision the AP with an acceleremeter that can detect when the AP is being relocated, and then force the administrator to redo the setup before the AP will provide service.

4. SECURITY PROPERTIES

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

1. Integrity: A location proof is signed by the access point that issued it. Thus, a proof cannot be modified by anyone other than the piece of infrastructure where it originated from.

2. Non-transferability: Once a location proof is issued, it cannot be transferred from one user to another. When requesting a proof, the user incorporates in the request a signed version of the access point's sequence number. This ensures that the user making the request is the holder of the appropriate private key that corresponds to the public key that appears in the request. When the location proof is issued, it incorporates the client's public key signed by the access point, thereby designating this client as the recipient of the location proof.

Once location proofs are issued, clients can transfer them to others only by sharing their private keys. While this is possible (e.g., collusion attacks), the feasibility and ease of such attacks are just a function of the identity scheme used by the location proofs. In some identity schemes, the cost for mounting a collusion attack is lower than others. For example, when using e-mail addresses as identities, a collusion attack requires two users to share the passwords of their e-mail accounts. Instead, when using PGP identities, a collusion attack requires the users to share their PGP identities; this sharing is likely to be detected by their circle of "friends" – others than have vetted their identities by signing them. There are other possible forms of mounting a collusion attack that do not require users to share their private keys; for example, users can collude when requesting location proofs from the infrastructure. We will discuss these relay attacks in Section 5.

3. Un-forgeability: Location proofs are signed by the infrastructure. Therefore, as long as the private keys of the access points are not compromised, it is impossible for an attacker to forge them.

4. Privacy: To reduce the privacy risks, any user can choose when to ask for a location proof and when to present their location proofs to any applications. An alternate implementation is one in which the infrastructure itself monitors the mobile devices and can vouch for the location of a device without any explicit participation. Such a design is often being proposed as a way to build surveillance and monitoring infrastructure. Next, we present this alternate design examining its privacy properties in-depth. The role of our examination is to identify precisely what privacy drawbacks such a big-brother design has.

4.1 The Privacy Implications of a Big-Brother Design

An alternate way of implementing location proofs is having the access points monitor all the clients continuously. In such a scheme, a client must request the APs to prove its geographical location. In turn, APs must record and preserve their clients' locations for future requests. The main benefit of such a design is that it requires no client support – the entire functionality of location proofs is infrastructure-based.

One important drawback of an AP-based design of location proofs is the loss of privacy. As mobile infrastructure is becoming ubiquitous, the continuous monitoring of clients raises the following three privacy concerns:

1. Privacy guarantees: What privacy guarantees does the infrastructure offer and who enforces them? Privacy watchdogs point out that the infrastructure is maintained by corporations whose incentives are often misaligned with people's expectations of privacy. Currently, there is no established set of guidelines of what information is acceptable to be recorded or stored, and what is not acceptable. Even if a privacy policy exists, enforcing it and verifying it is likely to be challenging because it requires cooperation from the infrastructure owners.

2. The implementation of the privacy policy: Most of today's privacy discussion and concerns are about an "all or nothing" privacy policy – either the infrastructure can monitor all people continuously or all people remain anonymous all the time. In practice, we believe most users want privacy in certain cases while in others they are willing to be monitored by an infrastructure. For example, employees might be willing to be monitored on their work premises while at work, whereas they would prefer to remain anonymous outside of working hours. While implementing such policies is relatively simple, making them intuitive and easy to use is likely to be challenging. For example, a privacy policy that requires people to opt-out from being monitored during certain times of the day while opting back in during other times will likely be error prone and too hard to use.

3. The granularity of private information: How does the infrastructure decide when to share the information collected with third-party applications and services? What is the granularity for controlling access or anonymizing the data? For example, users might be willing to allow the infrastructure to share aggregate statistics with third party applications (such as how crowded different city areas are), but they might not be willing to share personally identifiable information (such as the timeline of an individual).

At a high-level, these privacy concerns stem from two issues: first, users must rely on the infrastructure not to be malicious; and second, the infrastructure must provide access control and data sharing policies that are easy to use and satisfy the entire userbase. While both issues are challenging in practice, this paper explores a solution to the second problem – providing control and sharing policies that put users in control of their privacy policies.

Our design of location proofs puts the users in control. Users continuously collect location proofs about where their location is on their devices. The role of the infrastructure is restricted to just providing these location proofs to those users that are nearby and who request them. Users can then use the set of location proofs they have collected over time for a multitude of services. This puts users in control to decide how they want to use this information and who they want to share it with. However, our system cannot prevent the wireless infrastructure from monitoring users continuously if it chooses to do so.

4.2 Physical Attacks

Physical attacks pose a significant threat to location proofs. For example, an AP can be stolen and relocated, or it can be broken into to change its latitude and longitude coordinates. The use of tamper resistant hardware, such as a Trusted Platform Module (TPM), can increase the difficulty of mounting such attacks in practice.

5. STRONG IDENTITIES

Our discussion of location proofs so far has focused on certifying that a user's mobile device is in a certain location at a certain time. However, people do not always carry their devices. or even worse they may deliberately pass their devices to others with the intent of appearing to be somewhere else. Ideally, we would like to certify that a person rather than a device is in a particular place at a particular time. While not all the applications presented in Section 2 need this stronger verification, some applications might require it to be viable. For example, using location proofs for both police investigations and voter registrations would likely require an approach that makes it very difficult for people to lie about their whereabouts. In the remainder of this section, we present a high-level description of one approach to solving this problem.

One way to ensure the presence of the device's owner when issuing the location proof is to incorporate into the proof a piece of hard-to-forge information that identifies the owner. At first, we considered using a photo of the owner in the location proof issue protocol. The AP would ask the mobile device to take a photo of the owner and transmit it to the AP. The AP would then incorporate the photo inside the location proof together with the public keys, the timestamp, and the location information as described in Figure 1. The entire proof is signed by the AP to prevent anyone from replacing the photo.

However, the photo itself is not sufficient to thwart these attacks. A malicious user could pass his device to someone else together with his photo. This other user could still impersonate the device's owner by merely sending this old photo to the AP when requested. To prevent this possibility, we also added a challenge to this protocol inspired by the use of CAPTCHAS on the Web. When requesting a photo of the device's owner, the AP also sends a nonce (i.e., a randomly chosen number). Before taking the photo, the user must write this nonce on a piece of paper and hold the paper in a visible place in the photo. Upon receiving the photo, the AP incorporates the photo and the nonce into the location proof. Anybody can verify now whether the owner appears in the photo and whether the nonce in the photo matches the nonce in the location proof.

While the use of "paper nonces" makes it harder for someone to impersonate the device's owner, this approach is still not perfect. For example, a malicious user could take a photo of himself with a blank piece of paper and pass it to someone else. When requesting a location proof, this other user could use automated photo editing to insert the nonce (e.g, using Photoshop). If attacks of this nature are a concern, this scheme can be modified yet again to raise the bar. For example, instead of a paper nonce, the AP can challenge the user by sending in an entire English sentence. The user must now read the sentence and make an audio recording of it, and return the audio content back to the AP to incorporate in the location proof. Attacking such a scheme is much harder. One way is to have the impersonator fake the owner's voice. Another way is

to have the owner record each word in English and pass all these individual word recordings to the impersonator. The impersonator could then stitch the words together to form the requested sentence in the challenge. However, stitching words together to form a sentence and making the audio recording sound like natural speech is not an easy task.

Finally, all these challenge-response identity schemes suffer from an additional attack. Upon receiving the challenge, an impersonator could quickly send the challenge to the device owner. The owner would send back the response, which the impersonator could then relay to the AP. For example, the device owner could take the photo showing the nonce or record the English sentence and transmit this data to the impersonator. Such attacks are similar to one way in which CAPTCHAS are attacked today – relaying the CAPTCHAS to impersonators who are hired to solve them manually [16]. To increase the difficulty of mounting collusion attacks, our design presented in Section 3 restricts a user to requesting a location proof within only 100 milliseconds from when an AP beacon is heard. To successfully mount a collusion attack in which a user near the AP relays the beacon to another user who is far away, the entire round trip communication must be done within 100ms. However, note that in collusion attacks where two users share their private keys, there is no need to relay messages between the users to make the attack successful.

6. RELATED WORK

One closely related previous research effort proposes a trusted geotagging service that can enable several mobile applications [9]. This service is specific to tagging content with trusted location and time metadata – the protocol uses content hashes to make sure users cannot modify the content later. While the role of location proofs is to securely identify the location of end users, the role of the geotagging service is to add trusted location information to content. The secure geotagging service enables a suite of new applications that can take advantage of knowing where and when the content is generated. In contrast, our work focuses on providing a concrete protocol for implementing location proofs over Wi-Fi, that makes it hard for users to lie about their location.

In [2], the authors propose a location-based authentication mechanism that generates location signatures from the reception of the raw GPS signals from a large number of satellites. Based on the random variation of received signals, the authors claim that such signatures are very difficult to forge. The authors do not describe in detail the signature validation technique. Previous work [1, 17] also proposes using challenge-response schemes for verifying the positions of wireless nodes. In these schemes, a wireless node demonstrates that it is within range of a particular AP by responding to a nonce sent by the AP. The goal of these schemes is to use multiple receivers to accurately estimate a wireless node location using RF propagation characteristics.

In our own previous work [15], we described Lockr, an access control scheme based on social relationships. Lockr provides social attestations: metadata exchanged by users that certify their social relationships. Location proofs are inspired by social attestations; both are signed digital content that can prove a piece of information, whether that is a social relationship (as is the case with Lockr) or location information (as is the case with location proofs). Also, the security protocol described in Section 3 is inspired by the attestation mechanisms developed in this previous work.

Finally, there has been much previous work in the area of location privacy for wireless users and devices [6, 8, 13]. The goal of all this work is to allow users to limit the ways in which their information is exposed to applications and services in a way that offers them privacy. Our goal is different and much more modest – to allow users to certify their locations to mobile applications. However, some of these privacy management techniques might be applicable to location proofs to further enhance their privacy properties.

7. CONCLUSIONS

This paper introduces location proofs, a simple mechanism that allows mobile devices to securely prove their current and past locations. We present six potential applications that would be enabled by an infrastructure that provides location proofs. We present a concrete protocol, implementable over Wi-Fi, in which APs issue location proofs to mobile devices. We then characterize the security properties of our proposed design, and we discuss the difficulties that arise from collusion attacks, such as when users share their devices with one another. In the future, we plan to build a prototype infrastructure that issues location proofs, to gain experience with applications that can use this primitive.

8. REFERENCES

[1] S. Capkun and J.-P. Hubaux. Secure positioning of wireless devices with application to sensor networks. In *Proc. of IEEE INFOCOM*, 2005.

[2] D. E. Denning and P. F. MacDoran. Location-Based Authentication: Grounding Cyberspace for Better Security, Feb. 1996. Computer Fraud & Security.

[3] B. Ford and J. Strauss. An offline foundation for online accountable pseudonyms. In *Proc. of the 1st International Workshop on Social Network Systems (SocialNets)*, Glasgow, Scotland, 2008.

[4] G. Monroe and C. Howells and J. Rain. OpenID Service Key Discovery. `http://openid.net/specs/openid-service-key-discovery-1_0-01.html`.

[5] Google Accounts. `http://google.com/accounts`.

[6] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services through Spatial and Temporal Cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[7] J. L. Heskett, T. O. Jones, G. W. Loveman, W. E. Sasser Jr, and L. A. Schlesinger. Putting the service-profit chain to work. *Harvard Business Review*, pages 164–174, March–April 1994.

[8] T. Jiang, H. J. Wang, and Y.-C. Hu. Location privacy in wireless networks. In *Proc. of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2007.

[9] V. Lenders, E. Koukoumidis, P. Zhang, and M. Martonosi. Location-based Trust for Mobile User-generated Content: Applications, Challenges and Implementations. In *Proc. of the 9th Workshop on Mobile Computing Systems and Applications (HotMobile)*, 2008.

[10] MaxMind GeoIP Database. `http://www.maxmind.com/app/ip-location`.

[11] Microsoft. Windows Live ID. `http://accountservices.passport.net`.

[12] OpenID. `http://openid.net/`.

[13] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Proc. of MobiCom 2007*, Sept. 2007.

[14] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Wiley; 2nd edition, 1995.

[15] A. Tootoonchian, K. K. Gollu, S. Saroiu, Y. Ganjali, and A. Wolman. Lockr: Social Access Control for Web 2.0. In *Proc. of the 1st ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, Aug. 2008.

[16] ZDNet. Inside India's CAPTCHA solving economy, 2008. `http://blogs.zdnet.com/security/?p=1835`.

[17] Y. Zhang, Z. Li, and W. Trappe. Power-Modulated Challenge-Response Schemes for Verifying Location Claims. In *Proc. of IEEE Globecom*, Nov. 2007.