Privacy-Preserving Alibi Systems

Benjamin Davis
University of California, Davis
One Shields Ave.
Davis, CA 95616
bendavis@ucdavis.edu

Hao Chen
University of California, Davis
One Shields Ave.
Davis, CA 95616
hchen@cs.ucdavis.edu

Matthew Franklin
University of California, Davis
One Shields Ave.
Davis, CA 95616
franklin@ucdavis.edu

ABSTRACT

An alibi provides evidence of a person's past location and can be critical in proving her innocence. An alibi must be bound to a person's identity to prevent from being transferred to another person; however, requiring a person to reveal her identity during alibi creation would compromise the person's privacy. We propose a privacy-preserving alibi system, where a user conceals her identity during alibi creation. The user's identity is revealed only when she chooses to present her alibi to a judge. We design two privacy-preserving alibi schemes. In the first scheme, the alibi corroborator is a public entity and therefore needs no privacy protection. Our second scheme protects the privacy of the corroborator as well, where the identity of the corroborator is revealed only when he chooses to help the alibi owner to present her alibi to the judge. We discuss the properties of our schemes and demonstrate their advantages over current alibis. As ubiquitous mobile computing presents an attractive platform for deploying our schemes, we have implemented our schemes on an Android device and shown its satisfactory performance.

Categories and Subject Descriptors

K.4.1 [**Management of Computing and Information Systems**]: Public Policy Issues—*Privacy*

General Terms

Security

Keywords

alibi, privacy, mobile, location

1. INTRODUCTION

Black's Law Dictionary defines an alibi as *a defense based on the physical impossibility of a defendant's guilt by placing the defendant in a location other than the scene of the crime at the relevant time.* [5] The ability to provide evidence of one's past locations can be extremely important. For example, in 2008 murder charges were dropped against a Bronx man and his brother after they used their New York City Transit MetroCard to support their claim that they were miles from the scene of the crime at the time it occurred. [19]

Some alibis are based on witness testimony. Such an alibi relies on the memory of the corroborator; however, the corroborator may forget about the encounter or may misremember key details, such as the identity of the other party, or the date and location of the encounter.

Other alibis are based on physical evidence. As mobile devices become ubiquitous and accompany us on our daily activities, they have the ability to determine where we are and what we are doing. Location-based services like Google Latitude can track our every move, so they could provide physical evidence as our alibis.

If a physical evidence is not bound to a person, it can be used by other people to claim their fake alibi. On the other hand, if a person has to reveal her identity when creating a physical evidence, then her privacy is at risk. Privacy advocates are becoming increasingly concerned [1] that third-party services have so much access to information about our lives. These services generally require the user to decide whether they want to be tracked at the time the tracking occurs. Imagine a user who temporarily disables their location-tracking service to prevent their employer from learning of a long lunch break. While this may seem a reasonable decision at the time, the user has no way to "go back" and show their location later if they need to prove their innocence when they are incorrectly accused of a crime.

When a person claims an alibi, she must reveal her identity because the judge must verify that the identity in the alibi matches the identity of that person. However, our key insight is that we could design an alibi system where a person does not reveal her identity when creating her alibi. In this system, a person remains anonymous until she chooses to use her alibi in front of a judge. This system allows a user to create alibis whenever she can without compromising her privacy.

An alibi involves two parties: the *owner*, who benefits from the alibi, and the *corroborator*, who testifies for the owner. Our goal is to allow an owner to create alibis with corroborators without revealing her identity to the corroborators. To prevent the transfer of an alibi from one owner to another, the alibi must be bound to the owner's identity, although this binding is hidden at alibi creation time. To prevent the owner from lying about the context, such as

ASIACCS '12, May 2–4, 2012, Seoul, Korea.
Copyright 2012 ACM 978-1-4503-1303-2/12/05 ...$10.00.

time and location, the alibi must also include the context certified by the corroborator.

The advent of ubiquitous mobile computing provides an attractive platform for implementing this privacy-preserving alibi scheme. The user's mobile device can act as the user's delegate in alibi creation. The corroborator can be a public entity, such as a subway station, or a private entity, such as another mobile device. When public entities corroborate alibis they need not protect their privacy, but private entities may wish to protect their own privacy. Therefore, we have designed two privacy-preserving alibi schemes, one with public corroborators (Section 2) and one with private corroborators (Section 5).

1.1 Contributions

- We propose a privacy-preserving alibi system where the identity of the alibi owner is concealed at the time of alibi creation. The owner reveals her identity only when she chooses to present her alibi to a judge.
- We design two privacy-preserving alibi schemes. The public corroborator scheme (Section 2) always reveals the identity of the corroborator. By contrast, the private corroborator scheme (Section 5) conceals the identity of the corroborator, and the corroborator reveals his identity only when he agrees to help the alibi owner to present her alibi to the judge.
- We discuss the properties of our schemes and demonstrate their advantages over current alibis. (Section 7)
- We have implemented the scheme on a mobile device and evaluated its performance. (Section 8)

2. PUBLIC CORROBORATOR SCHEME

In our "public corroborator" scheme we assume that all corroborators are publicly known, and the corroborators' identities and locations are not considered private. These schemes are appropriate for settings where corroborators are public entities, such as subway stations or other infrastructure without privacy concerns. We discuss an alternative scheme in Section 5 that allows the corroborator to control the disclosure of his identity, which is more appropriate for corroborators who also have privacy concerns, such as other mobile device users.

In contrast to alternative schemes (such as VeriPlace [12]), we do not require that corroborators have fixed locations, nor do we require a central database mapping all corroborators to their locations.

2.1 Overview

Figure 1 illustrates the two phases in our public corroborator alibi scheme.

2.1.1 Alibi Creation

The owner can opportunistically participate in "alibi creation" whenever corroborators are available. The owner creates an OwnerStatement for each alibi she creates. This OwnerStatement is tied to the Owner who created it, but by itself can't reveal the identity of the Owner unless the Owner reveals that link.

The corroborator sends back CorroboratingEvidence, which is some way for the corroborator to state "I have received the OwnerStatement a in the current context c."

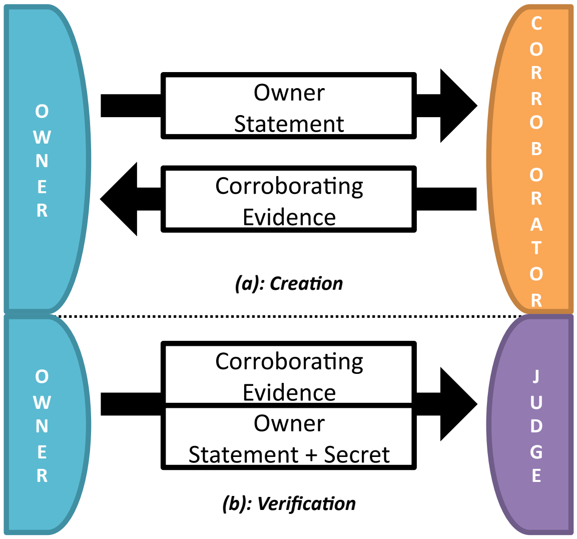

Figure 1: The public corroborator scheme

Participating in the creation phase does not reveal the identity of the owner, but does give the owner the opportunity to claim an alibi for their current context at a later time.

2.1.2 Alibi Verification

When the owner wishes to claim an alibi they have created, they participate in the "alibi verification" stage with a judge. In this stage, the owner reveals her identity associated with only the OwnerStatement used to create the single alibi she is claiming. This doesn't reveal her identity in any other unclaimed alibis, or allow anyone else to create more alibis on her behalf.

The owner must demonstrate two things to the judge. First, the owner must show that the corroborator certifies that they received a specific OwnerStatement in the context where the owner claims to have an alibi. Second, the owner must demonstrate the link between the OwnerStatement and the Owner. The judge checks to make sure that the OwnerStatement corresponds to the Owner (that is, her identity), and that it is the same OwnerStatement that the corroborator claims to have received at the context in question.

We note that just as in the "traditional" alibi setting, the "strength" of an alibi (e.g. as considered by a jury) depends heavily on the perceived trustworthiness of the corroborator. We compare our alibis to traditional alibis in Section 7.

2.2 Design

2.2.1 Initialization

Each alibi owner and alibi corroborator has their own public/private key pair (pk_o, sk_o) and (pk_c, sk_c), respectively. Both parties must have the necessary sensors to determine the current context (location, date, time), and represent this information in the same way. Also, we assume that the alibi owner has access to a collision-free hash function $MD(\cdot)$ (for Message-Digest).

The judge has access to both the alibi owner's and alibi corroborators's public keys (with the ability to verify signatures made by these parties). The judge can also compute the message digest function $MD(\cdot)$ used by the alibi owner.

2.2.2 Alibi Creation

We assume that the owner has a fixed, unique identity called OwnerID, and agrees with the corroborator on a value describing the current context.

The alibi owner creates the tuple i which includes the owner's identity and the current context as determined by the owner.

$$i = (\text{OwnerID}, \text{Context}_o)$$

The alibi owner signs i, and a tuple containing i and the alibi owner's signature over i becomes what we call the *OwnerFeatures*.

$$\text{OwnerFeatures} = (i, \text{Sign}_{sk_o}(i))$$

Sending the OwnerFeatures directly to the corroborator would reveal the owner's identity. We prevent this by using highly efficient cryptographic primitives to create a representation of the OwnerFeatures that doesn't reveal the OwnerFeatures values without extra information from the Owner.

We use a cryptographic commitment scheme to create the OwnerStatement from the OwnerFeatures. Specifically, we use the scheme presented by Halevi and Micali [8], which is a non-interactive string commitment scheme based on collision-free hashing. We note that we are not tied to this particular scheme. Other string commitment schemes (such as [3]) could be used in place without changing the security our of scheme (except of course for differences in the hardness assumptions underlying the commitment schemes).

To form a commitment to the OwnerFeatures, the alibi owner uses the message digest function $MD(\cdot)$ to compute s_o where

$$s_o = MD(\text{OwnerFeatures})$$

The alibi owner chooses a random value x_o which we call the *verification secret*, which she keeps secret until she wishes to claim her alibi. Next, the alibi owner randomly selects[1] a universal hash function $h_o(\cdot)$ where

$$h_o(x_o) = s_o$$

Finally, the alibi owner computes

$$y_o = MD(x_o)$$

This gives the owner the OwnerStatement, which is the commitment to the OwnerFeatures.

$$\text{OwnerStatement} = (h_o, y_o)$$

The owner sends the completed OwnerStatement to the alibi corroborator. The corroborator must certify that the OwnerStatement was received in the current context. The alibi corroborator combines the OwnerStatement tuple with the current context (as determined by the corroborator) to create the tuple

$$a = (h_o, y_o, \text{Context}_c)$$

and creates the CorroboratingEvidence

$$\text{CorroboratingEvidence} = (a, \text{Sign}_{sk_c}(a))$$

The alibi corroborator sends the CorroboratingEvidence back to the alibi owner. The alibi owner confirms that the values in a_o are correct (including the context provided by the corroborator), and that the signature from the alibi corroborator is valid. At this point, the alibi owner has the CorroboratingEvidence it needs to claim their alibi later.

We note that the alibi owner must store the context, h_o, x_o, and the corroborator's signature. This information allows the owner to recompute the rest of the values needed to claim the alibi. The owner can claim the alibi they just created without requiring the corroborator to maintain any information about this exchange.

2.2.3 Alibi Verification

When the alibi owner wishes to claim their alibi, she present her evidence to the judge. The owner must demonstrate two things. First, the owner must provide the CorroboratingEvidence, which shows that the corroborator certifies that they were presented with a specific OwnerStatement in a specific context. Second, the owner must demonstrate that the OwnerStatement the corroborator received is valid and linked to the owner's identity.

The owner sends h_o, y_o, the context value provided by the corroborator and the corroborator's signature over these values, as well as the owner's verification secret x_o. Note that the if the owner did not store y_o then she may recompute it with $y_o = MD(x_o)$. The judge checks that the signature is valid.

Then, the owner decommits their commitment in the OwnerStatement by providing x_o and the OwnerFeatures values. The judge inspects the contents of OwnerFeatures to make sure the OwnerID belongs to the owner, that the context in the OwnerFeatures matches the context signed by the corroborator, and that the owner's signature is valid. The judge then computes $s_o = MD(\text{OwnerFeatures})$, and checks that $h_o(x_o) = s_o$ and that $MD(x_o) = y_o$.

Sybil-alibi attacks.

Normally alibis work only in the favor of the owner, because the owner chooses to verify an alibi only when she would benefit. However, in certain circumstances, the owner might be coerced to verify an alibi in her disfavor. In this case, we need to prevent the Sybil-alibi attack, where malicious corroborators create new alibis for the owner based on her verified alibi. In this attack, a malicious corroborator takes the OwnerStatement from the verified CorroboratingEvidence to create a new CorroboratingEvidence. The same x_o that the owner used to verify her original CorroboratingEvidence could be used to verify this forged CorroboratingEvidence.

To prevent this Sybil-alibi attack, when multiple CorroboratingEvidences are provided for the same owner, our scheme verifies that the OwnerStatement in each CorroboratingEvidence is unique.

3. THREAT MODEL

To distinguish them from the alibis in our systems, we call traditional alibis (as thought of in the current legal system) "physical alibis." A physical alibi has three components: the identity of the owner, the identity of the corroborator, and the context (date, time, and location information).

[1] We choose a random x_o, and select $h()$ of the form $h(r) = Ar + b$ by choosing A randomly and computing $b = s_o - A(x_o)$, where $h()$ is in linear space over GF(2).

Identity.

We require a public key infrastructure that binds keys to legal identities. This allows us to represent witness statements in the physical world as signed messages in our scheme. Since a private key represents a legal identity, we assume that no one except the owner has her private key. Determining whether the identity associated with a private key matches the identity of the human using the key is outside of the scope of this paper.

Context.

In a physical alibi, the corroborator believes that both he and the alibi owner were in the same context based on certain facts, which determines the reliability of the alibi. For example, if the corroborator saw the alibi owner, the alibi is highly reliable; however, if the corroborator overheard the alibi owner's voice in another room but never saw her, the alibi is less reliable, because the corroborator could have heard a recording of her voice. We distinguish between the reliability of the alibi evidence and the trustworthiness of the corroborator, and we will discuss the latter next.

In our scheme, we require that the corroborator can correctly measure his context, which should also includes the means by which he interacted with the alibi owner. The judge takes the means of the interaction into consideration when determining the reliability of the alibi evidence. For example, an alibi created through an interaction via near field communication (NFC) might be considered stronger than one created over WiFi, as the corroborator is likely to be more certain of his proximity to the alibi owner. We can improve the reliability of alibis evidence by secure location verification techniques [18], which are orthogonal to this paper.

Our scheme also requires the owner to include his view of the context in the alibi (in OwnerFeatures). The purpose is to prevent the attack where a malicious corroborator creates a new CorroboratingEvidence from an old OwnerStatement without the alibi owner's participation. If the corroborator uses a different context in the CorroboratingEvidence than the one in OwnerFeatures, the judge will detect the discrepancy when verifying the CorroboratingEvidence. On the other hand, if the attacker must use the same context value, then the attacker can only create additional CorroboratingEvidence for the original alibi. During the verification stage a judge can detect when two CorroboratingEvidence values correspond to a single OwnerStatement, revealing this misbehavior.

Privacy.

We assume that an attacker may try to learn the identity of any party in our system only via messages in our protocols. Therefore, we do not consider privacy attacks using out of band channels. For example, the cellular network provider of the alibi owner may learn her identity; the corroborator may use recording devices, such as cameras, to determine the identity of the alibi owner. These are out of the scope of this paper.

Trust.

We require no trusted third party. Moreover, we require no trust between alibi owners and corroborators. If a corroborator is malicious, he can refuse to provide the valid CorroboratingEvidence the owner needs for the alibi. This is a form of denial of service attack. We do not attempt to prevent this attack, because a solution would force witnesses to provide valid alibis, which we do not believe to be desirable. On the other hand, our scheme prevents a malicious corroborator from discovering the identity of the alibi owner (Section 4.1.3).

If the corroborator collaborates with the alibi owner, they can create false but valid alibis, which is perjury. Just as we cannot prevent the creation of perjury (even though we may expose it by other means) in real life, our scheme does not try to prevent perjury.

If the corroborator unilaterally intends to create false alibi to benefit the owner without the collaboration from the owner, this is another form of perjury. In the physical world, we cannot prevent the creation of such perjury (even though we may expose them through other means). By contrast, our scheme can detect such attacks (Section 4.1.1).

As with physical alibis, the value of alibis produced by our scheme depends on the trustworthiness of the corroborators. We leave the problem of determining the trustworthiness of the corroborator to the judge.

4. PROPERTIES OF THE PUBLIC CORROBORATOR SCHEME

Now that we've defined an alibi scheme, we describe all of the security properties we desire in our setting and how our implementation satisfies them.

4.1 Security Properties

4.1.1 Non-forgeability

A CorroboratingEvidence binds the owner's identity, the corroborator's identity, and the context. An alibi is valid if it can be successfully verified (Section 2.2.3). We claim that no valid alibi can be created without the collaboration of both the owner and the corroborator.

First, we consider how someone, including the corroborator, could forge an alibi without the owner's cooperation.

- The forger could try to forge a fresh alibi for an alibi owner, but this would fail because he does not have the owner's private key needed to create a valid OwnerStatement.

- The forger could try to use an existing unclaimed alibi to create another CorroboratingEvidence. Even if the forger creates new CorroboratingEvidence for an existing alibi, no one could verify the new CorroboratingEvidence without the verification secret chosen by the owner during the creation of the original OwnerStatement.

- The forger could attempt to use an existing OwnerStatement to create fake CorroboratingEvidence with a context different from the one in which the owner created the OwnerStatement. However, this forgery would be detected in the verification stage because the OwnerFeatures linked to the OwnerStatement includes the owner's context value, which will not match the context in the forged CorroboratingEvidence.

- The forger could use an existing OwnerStatement to create fake CorroboratingEvidence for the same context in which the owner created the OwnerStatement.

However, at worst this attack can only result in adding a false corroboration to an existing, valid alibi. As the owner already has a valid alibi this forgery can only give the owner an additional (malicious) corroborator of the existing alibi, and cannot result in an alibi that places the owner in a different context. If the judge inspects both alibis then this misbehavior is easily detected, as both CorroboratingEvidence values will correspond to the same OwnerStatement, which would not occur under normal circumstances.

Next, we consider how someone, including the alibi owner, could forge an alibi without the corroborator's collaboration. This is infeasible because the forger doesn't have the private key of the intended corroborator so therefore cannot create the signature in the corroborating evidence.

4.1.2 *Non-transferability*

A corroborating evidence is non-transferable because it has the signature of both the owner and the corroborator.

4.1.3 *Privacy*

Our scheme preserves the privacy of the owner in the following properties:

- No one, including a malicious corroborator, can uncover the identity of the alibi owner at any stage before the owner verifies her CorroboratedEvidence in the protocol.

- No one, including any number of collaborating malicious corroborators, can link multiple unclaimed alibis created by the same alibi owner (including unclaimed OwnerStatements and corresponding CorroboratingEvidence values).

- When an owner claims her alibi by entering into the verification stage, she reveals her identity. However, no one, including any number of collaborating malicious corroborators, can link her to any of her unclaimed alibis.

The above properties are guaranteed by the string commitment scheme in our protocol.

4.2 Other Properties

4.2.1 *Storage*

Our scheme requires the owner to store all the data necessary for verifying an alibi. By contrast, no corroborator needs to store any data about the alibis that he has helped create (except their private keys, which our threat model assumes). The advantage of this design is that it aligns with the incentive of the owner to safe guard his alibis.

4.2.2 *Efficiency*

Our scheme is efficient both in time and space. In Section 8 we evaluate the performance of our scheme on an Android device.

5. PRIVATE CORROBORATOR SCHEME

5.1 Motivation

In our public corroborator scheme, the corroborator's identity is always revealed during the creation phase, but the owner's identity isn't revealed until the owner wishes to claim that alibi. In settings where corroborators are public entities (e.g. subway stations), it is acceptable for someone to learn the identities of every corroborator with whom she creates an alibi. However, in other settings a corroborator may not want to reveal his identity every time an alibi owner wants to create an alibi with him. Particularly, if the corroborator allows his mobile device to create abilis for anyone within proximity, the previous scheme would allow an attacker to identity and track the corroborator. We wish to design a private corroborator scheme where the corroborator's identity is not revealed at alibi creation, analogous to the property that the owner's identity is not revealed at alibi creation in the public corroborator scheme.

5.1.1 *Rejected Designs*

One might simply apply our public corroborator scheme but allow the corroborator to decide to whether to create CorroboratingEvidence for the alibi owner during each encounter. However, this would require the corroborator to decide at alibi creation time whether he wants to reveal his identity, while the owner can wait until she verifies her alibi to reveal her identity. This deficiency would create big privacy and usability headache for the corroborator: for each alibi creation request, the corroborator would have to decide whether to help create the alibi either manually or using some policies, which could be complex and error prone.

One might require the corroborator to store each OwnerStatement (sent by the alibi owner) along with the associated context, and only return the CorroboratingEvidence when the owner requests to verify the alibi rather than during alibi creation. However, this would require the corroborator to bear the burden of storing the alibi, when the owner has much higher incentive to store her alibis safely. In this setting, an honest and willing corroborator may be unable to corroborate an important alibi because he deleted the alibi when he ran out of disk space. We would like a scheme that is completely stateless for the corroborator. We want to allow the owner to retain all of the information necessary for her and the corroborator to corroborate her alibi.

One might imagine a scheme where the corroborator sends his created alibi to a trusted third party instead of the alibi owner. However, this violates the requirement for no trusted third party in our threat model.

One might suggest that we use zero knowledge schemes to allow the provider to prove that there exists some corroborator without revealing the identity of the corroborator. However, we believe that alibis are of little value if the corroborator's identity is not revealed, because the value depends, in part, on the trustworthiness of the corroborator.

5.2 Overview

Under the above considerations, we have designed a private corroborator alibi scheme where the alibi owner, when wishing to verify her alibi, simply contacts the alibi corroborator to "ask" if he is willing to corroborate her alibi. The owner can use an anonymous messaging system such as [13] to contact the corroborator. Our scheme gives the corrob-

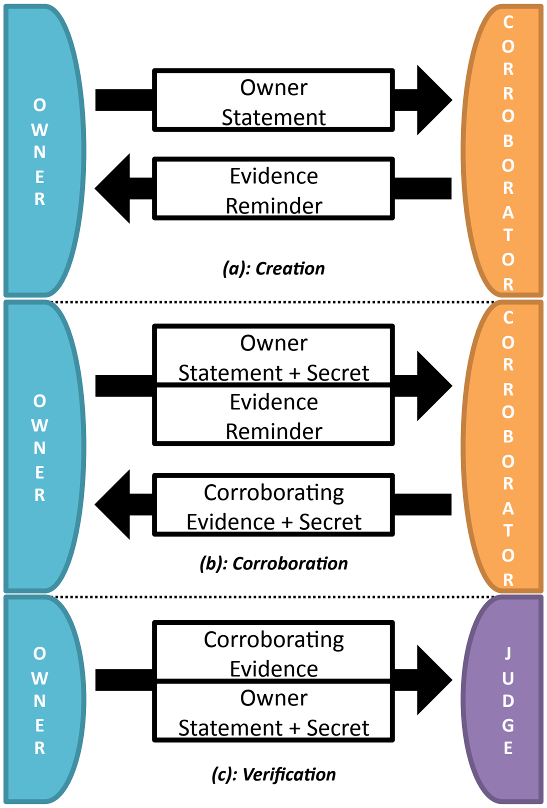

Figure 2: The private corroborator scheme

orator as much control over his privacy as the owner over hers. Just as the alibi owner can freely ask corroborators to create alibis without revealing her identity, the corroborator can freely help owners to create their alibis without revealing his identity. The alibi owner reveals her identity only when she wishes to verify her alibi, and the corroborator reveals his identity only when he helps the owner to verify her alibi.

We show the three phases of our private corroborator scheme in Figure 2. Note that we have added a "corroboration" phase, in which the owner asks the corroborator to identify himself and corroborate the alibi. This means that the owner must be able to contact the corroborator without knowing his identity. In our private corroborator scheme we assume that the participants also have access to some anonymous messaging system that provides this functionality, such as SMILE [13]. SMILE requires a trusted third party, but only for message delivery. While misbehavior of a third party in SMILE might prevent an alibi owner from communicating with a corroborator (denial-of-service), this third party can not compromise the privacy of the alibis in our scheme.

If the corroborator agrees to corroborate the alibi, then he returns the CorroboratingEvidence the owner needs to claim her alibi. The owner claims her alibi by presenting her OwnerStatement, the link to her OwnerFeatures, and the CorroboratingEvidence to the judge.

5.3 Initialization

Just as in our public corroborator scheme, each alibi owner and alibi corroborator has their own public/private key pair (pk_o, sk_o) and (pk_c, sk_c), respectively. In our private corroborator scheme, each corroborator also has her own pseudorandom function $\text{prf}_c(\cdot)$ with secret key. Both parties have the rest of the capabilities as in the public corroborator scheme.

5.4 Alibi Creation

In this phase, the owner begins the exchange just as in the public corroborator scheme. The owner creates her OwnerStatement in the way described in Section 2, and sends the OwnerStatement to the corroborator.

Upon receiving the OwnerStatement, the corroborator computes his signature over the OwnerStatement and the context in which it was received by the corroborator. However, instead of sending this signature back to the alibi owner, the corroborator commits to this data using the cryptographic string commitment scheme.

To commit to the information needed to corroborate the owner's identity the corroborator selects a random value r_c and computes

$$\begin{aligned} j &= (\text{OwnerStatement}, \text{Context}_c) \\ s_c &= MD(j, \text{Sign}_{sk_c}(j)) \end{aligned}$$

The corroborator commits to s_c by computing

$$\begin{aligned} x_c &= \text{prf}_c(r_c) \\ y_c &= MD(x_c) \end{aligned}$$

and choosing a universal hash function $h_c(\cdot)$ where

$$h_c(x_c) = s_c$$

Combining these values gives the corroborator the EvidenceReminder, which the corroborator sends to the alibi owner.

$$\text{EvidenceReminder} = (h_c, y_c, r_c, \text{Context}_c)$$

In order to be able to claim their alibi later, the alibi owner only needs to store the context used in the OwnerFeatures, h_o, x_o, and the EvidenceReminder $(h_c, y_c, r_c, \text{Context}_c)$. The alibi corroborator does not need to store any values (except for their own private keys).

Note that the corroborator does not send his verification secret x_c to the owner at this time. This value does not need to be stored because the corroborator can use r_c and her pseudorandom function (keyed with her secret key) to recompute x_c in the corroboration phase.

Also in this phase the provider and corroborator must exchange whatever information necessary to allow the provider to send a message to the corroborator anonymously if the provider wishes to claim their alibi.

5.5 Alibi Corroboration

To claim their alibi, the owner contacts the corroborator (via some anonymous messaging system such as [13]) to ask if they are willing to reveal their identity in association with a specified context. If so, then the owner first reveals his identity to the corroborator by sending her verification secret x_o and OwnerFeatures to the corroborator. The corroborator checks the owner's decommitment, and proceeds if the decommitment is valid.

The owner sends the EvidenceReminder the corroborator created in the creation phase back to the corroborator. Because the EvidenceReminder contains r_c, the corroborator can recompute x_c needed to decommit h_c and y_c. The corroborator takes the OwnerStatement and context provided by the owner and recomputes

$$\begin{aligned} j &= (\text{OwnerStatement}, \text{Context}_c) \\ s_c &= MD(j, \text{Sign}_{sk_c}(j)) \end{aligned}$$

The corroborator checks to see if this s_c value is the value they committed to in the creation phase. That is, the corroborator checks to see that the $h_c(x_c) = s_c$ and $MD(x_c) = y_c$.

If so, then the corroborator knows that they must have created a signature over the OwnerStatement and in Context$_c$, which they would only do if they received that OwnerStatement in an alibi creation exchange in that given context. So, because the corroborator has decided to support the owner's alibi claim for this context, the corroborator returns the CorroboratingEvidence including the signature in the same format as in the public corroborator scheme.

$$\begin{aligned} a &= (h_o, y_o, \text{Context}_c) \\ \text{CorroboratingEvidence} &= (a, \text{Sign}_{sk_c}(a)) \end{aligned}$$

The corroborator sends the CorraboratingEvidence to the owner. In our private corroborator scheme, the corroborator also sends the x_c value to the alibi owner along with the CorraboratingEvidence. The alibi owner uses this to verify the corroborator's decommitment to the EvidenceReminder. This allows the corroborator to demonstrate the link between the signature in the CorroboratingEvidence and the EvidenceReminder the corroborator gave to the owner in the creation phase. The owner can make sure that the signature received in the corroboration phase is the same as corroborator's signature made during the creation phase.

5.6 Alibi Verification

Alibi verification in the private corroborator scheme is done in exactly the same way as in the public corroborator scheme. Just as before, the owner sends h_o, y_o, and r_o to the judge, along with the context as specified by the corroborator, and the corroborator's signature over these values. The owner also decommits their OwnerStatement, and reveals the OwnerFeatures. The judge checks the corroborator's signature, and the owners decommitment and OwnerFeatures.

6. PROPERTIES OF THE PRIVATE CORROBORATOR SCHEME

The private corroborator scheme shares all the properties of the public corroborator scheme described in Section 4 (except that the corroborator learns the identity of the alibi owner at the corroboration stage in the private corroborator scheme, instead of at the verification stage in the public corroborator scheme). In this section we discuss additional properties of the private corroborator scheme.

6.1 Privacy

Our scheme preserves the privacy of the alibi corroborator in the following properties:

- No one, including a malicious alibi owner, can uncover the identity of the alibi corroborator before the corroborator creates the CorroboratingEvidence in the protocol.
- No one, including any number of collaborating malicious alibi owners, can link multiple EvidenceReminders created by the same corroborator.
- When a corroborator sends a CorroboratingEvidence in reply to an EvidenceReminder sent by an alibi owner, the corroborator reveals his identity. However, no one, including any number of collaborating malicious alibi owners, can link him to any of the evidence reminders that he has created (in the creation stage) but not yet used (in the corroboration stage).

These properties are guaranteed by the string commitment scheme in our protocol.

6.2 Reciprocity

The private corroborator scheme raises the question of reciprocity of privacy: is it possible for one party to learn the other party's identity without revealing his own? A fair exchange scheme (e.g. [15]) might allow us to achieve privacy reciprocity but it requires a trusted third party, which our threat model precludes.

We believe that privacy reciprocity is unnecessary for our scheme. First, before the parties enter the corroboration stage, neither party's identity is revealed. Second, after the parties enter corroboration, the owner reveals her identity before the corroborator does. Therefore, it is possible that the corroborator learns the owner's identity without revealing his identity to the owner, but only when the owner chooses to reveal her identity to get a corroborated alibi from the corroborator. Just as in the physical world, a defendant cannot remain anonymous while asking a witness to testify for her, and has to bear the risk that the witness may decline to come forward after she reveals her identity.

Note that a malicious corroborator cannot force an alibi owner to reveal herself, as the owner must initiate the corroboration stage.

7. COMPARISON TO PHYSICAL ALIBIS

We call alibis used in current legal systems *physical alibis*. A physical alibi includes the corroborator, the owner (a.k.a. the beneficiary), and the context, which includes the means by which the corroborator identifies the owner. For example, in the case a personal witness, the corroborator is a person and the means is via physical senses such as vision; in the case of a physical evidence, the corroborator is the entity that issues the physical evidence (e.g., a subway station), and the means is the physical evidence (e.g., a subway ticket).

We discuss some comparisons between our alibis and physical alibis. The unique properties of our alibis give participants several advantages over physical alibis, including:

- They better protect the privacy of the alibi owner (and corroborator in our private-corroborator scheme)
- They have non-forgeability properties beyond that of many physical alibis
- They embed the identities of the participants directly and unambiguously into the alibis
- They help prevent alibis from being forgotten or faded over time

7.1 Common Properties

Trustworthiness of Corroborator.

The strength of a physical alibi depends on the reliability of the evidence and the trustworthiness of the corroborator. The same applies to our alibis. For example, our scheme cannot prevent a collaborating owner and corroborator from creating a fake but valid alibi (a.k.a. perjury). Just like physical alibis, our alibis leave the determination of the trustworthiness of the alibis to the judges.

Privacy of Corroborator.

The corroborator of a physical alibi may wish to protect his privacy by remaining anonymous. In this case, the alibi becomes useless because no one can judge the trustworthiness of the corroborator.

In our public scheme (Section 2, the identity of the corroborator is public. However, in our private scheme (Section 5), the corroborator may remain anonymous by refusing to corroborate the CorroboratingEvidence that he created earlier.

7.2 Benefits

7.2.1 Privacy

Consent on Alibi Creation.

In physical alibi settings, alibis may be created for a person without her consent. For example, without a person's content, she may be remembered by a doorman, or her photos may be taken by a camera. By contrast, our scheme requires the owner to initiate alibi creation.

Consent on Alibi Verification.

Although alibis often benefit the owners, they may harm the owners as well, e.g., when they are used as evidence by the prosecutors. Therefore, the owner has to decide in advance whether she wants her physical alibi to be created (if she does not, then she may disguise herself or avoid contact with the corroborator). Since she may not know in advance whether her alibi may be beneficial or harmful, she faces a dilemma: if she chooses to have her alibi created, it may harm her in the future; however, if she chooses not to have her alibi created, she may lose important alibis that could prove her innocent in the future. The cause of her dilemma is that other people can verify her physical alibis without her consent.

By contrast, an alibi in our scheme is unverifiable unless its owner consents (by providing the verification secret). This removes the dilemma that the owners face when creating physical alibis. In our scheme, the owners can freely create alibis. Later, she can decide to verify only the alibis that are beneficial to her.

7.2.2 Reliability

Accuracy.

There are a number of problems with physical alibis where one person is the corroborator for another. The corroborator may misremember the identity of the alibi owner (e.g., Charlie thinks that he saw Bob when he actually saw Alice), the context (e.g., Charlie thinks that he saw Alice on Monday when he actually saw Alice on Tuesday), or the link between the alibi owner and context (e.g., Charlie thinks that he saw Alice on Monday and Bob on Tuesday when he actually saw Alice on Tuesday and Bob on Monday). An alibi in our scheme binds the identities of the alibi owner and the corroborator to the context, so it avoids all the above human inaccuracies.

Operation	Time (sec)
OwnerStatement Creation	0.279
Corroborator Creation (public scheme)	0.070
Corroborator Creation (private scheme)	0.279
Corroboration (private scheme)	0.277
Corroboration verification (private scheme)	0.205
Alibi Verification	0.216

Table 1: Average execution times for alibi operations on a Motorola Droid

Availability.

Physical alibis rely on the memory of the corroborators to recall the encounters. However, the corroborator may forget the encounter partially or completely. By contrast, in our scheme the alibi owner stores all the data necessary to corroborate and verify the alibi.[2] Since the alibi owner benefits from the alibi, she naturally has the incentive to store her alibis safely.

7.3 Weaknesses

Our scheme requires a trustworthy public key infrastructure where each private key represents a person. In our threat model, we assume that each private key can only be accessed by the owner of that private key. We note that if Mallory gains access to Alice's private key, then she can create alibis on Alice's behalf. Our scheme is not intended to determine whether the user of a private key is actually the owner of the private key.

8. PERFORMANCE EVALUATION

To evaluate the real-world feasibility of our scheme, we have implemented all of the computational steps required to create, corroborate, claim and verify alibis in our public corroborator and private corroborator schemes. Our implementation runs on the Android mobile platform, and we measured the performance of all of these operations on a Motorola Droid phone.

8.1 Benchmarks

In order for a participant to create or corroborate alibis in our systems, they must first create their public/private key pairs and perform other initialization operations. Secure key pair creation is relatively slow on mobile devices (averaged 6.91 seconds). However, once the participants complete this one-time initialization, they can create and corroborate as many alibis as they like. After creating 1,000 alibis we computed the average time required for the Motorola Droid to complete the major operations in our scheme. The results of our benchmarks are shown in Table 1.

[2] In the private scheme (Section 5), the corroborator needs to remember his private key in the corroboration stage, but this is guaranteed by the assumption of a public key infrastructure in our threat model (Section 3).

OwnerStatement Creation the time required for the provider to create their OwnerFeatures and commitment to this value

Corroborator Creation (public scheme) the time required for the corroborator to create the corroborating evidence in the creation phase of the public corroborator scheme

Corroborator Creation (private scheme) the time required for the corroborator to create the EvidenceReminder in the creation phase of the private corroborator scheme

Corroboration (private scheme) the time required for the corroborator to verify the EvidenceReminder and use it to create CorroboratingEvidence in the private corroborator scheme

Corroboration verification (private scheme) the time required for the owner to verify that the CorroboratingEvidence received from the corroborator matches the evidence created in the creation phase

Alibi Verification the time required for the judge to verify an alibi claim by examining the CorroboratingEvidence, OwnerStatement and associated verification secret

8.2 Storage

In our public corroborator scheme, the alibi owner must retain h_o (160 bytes), x_o (120 bytes), the corroborator's signature (256 bytes) and the owner's context value (variable size) to claim an alibi at a later time.

In our private scheme, the owner must retain h_o (160 bytes), x_o (120 bytes), and the owner's context value (variable size). In order to claim the alibi later, the owner must also store h_c (160 bytes), y_c (20 bytes), and r_c (120 bytes), as well as the corroborator's context (variable size).

When the owner receives CorroboratingEvidence from the corroborator, they must store the corroborator's signature (256 bytes) to claim the alibi.

9. RELATED WORK

As mobile devices are becoming more popular, researchers are becoming more and more interested in "location-based services" ([10, 7]). Existing work such as [17] describe how to create location proofs to show that you were in a particular place, but these systems lack user control over their privacy.

Some researchers (e.g. [1]) are concerned about the security and privacy implications of such systems. Studies (such as [11]) show that users are sometimes hesitant to share data about their current location with others, which motivates further research in privacy-based approaches.

There are a number of papers describing general frameworks for privacy in location services ([4, 6, 14]), though most of the approaches only seek to prevent disclosure of user identities entirely rather than leaving the user in control of this information. Systems like Nymbler [9] allow pseudonyms to be correlated after a certain point in time, but does not provide the facilities to allow users to identify themselves in only specific exchanges as required by our alibi system. The SMILE system [13] provides a "missed encounters" service in a system where mobile devices perform passive key exchanges opportunistically. In SMILE, the results of these exchanges later requires both parties to participate when entities are claiming to have participated in the exchange. Parties connected in the SMILE scheme only means that they may have shared an encounter (exchanged an ephemeral key), and this is not bound to specific identities or locations. The results of these exchanges can be transferred to other users or be claimed to have taken place in a different context and so are unsuitable for alibis. SmokeScreen [2] is a system that allows users with existing relationships in the same area to share presence information, but requires a central, trusted broker server to reveal identities of the participants.

vPriv [16] is system for location-based vehicular services that protect driver privacy. In vPriv the emphasis is on allowing a server to perform functions on the path of a car (e.g. time/location pairs) without learning the identity of the driver for all time/location pairs. Due to the nature of these vehicular services, vPriv is only concerned with preventing widespread spoofing/cheating. vPriv detects cheating by performing random "spot checks," which reveal the identity of the user (without their consent or participation). In our scheme, there is no such trusted party that can reveal the identity of a user associated with an alibi without the owner's consent. vPriv's random spot checks are likely to catch users who cheat frequently, but it is very likely that a single faked record will go undetected. While this is acceptable in their setting, a single forged alibi may have tremendous consequences. Our schemes prevent users from forging even a single alibi successfully.

VeriPlace [12] is a different location proof architecture that has similar goals for protecting user privacy. They use wireless access points as the "corroborators" in their location proofs. The biggest difference between VeriPlace and our design is that VeriPlace requires each corroborator to have a permanently fixed, publicly-known location. Our scheme is much more flexible, as corroborators may move about and establish alibis whenever they encounter other users, allowing a much wider range of devices to be used as corroborators.

In APPLAUS [20], pseudoynms are used in location proofs to protect the privacy of the user. This scheme requires a trusted, third-party certificate authority to maintain a mapping of pseudonyms to real identities. To verify a location proof for a given user, a verifier asks the certificate authority to look up the pseudonyms associated with a user's identity. In contrast, our scheme does not require users to trust a third party to responsibly maintain the mapping between their real identity and the source identifier in our location proof (alibi). In our schemes, the identity associate with an alibi can *only* be revealed by the owner of the alibi.

Our scheme requires a string commitment scheme. While there are many different string commitment schemes, we feel that schemes based on hashing (such as those presented by Damgård et al. [3]) are both simple to implement and efficient in our setting. We used a string commitment scheme introduced by Halevi and Micali [8] as it is practical and provably secure in the unbounded receiver model, though our system could be tweaked to use other schemes.

10. CONCLUSION

We have introduced a privacy-preserving alibi system where the identity of the alibi owner is concealed at the time of alibi creation. The owner retains control over the disclosure of their identity, only revealing her identity when she chooses to present her alibi to a judge. We have designed two privacy-preserving alibi schemes: one for corroborators without personal privacy concerns, and another for corroborators who want to retain control over their the disclosure of their identities. These schemes provide several advantages over traditional alibis in the physical world. Finally, we have implemented both of our schemes on the Android mobile platform and demonstrated that our schemes are suitable for existing mobile devices.

11. ACKNOWLEDGMENTS

This paper is partially based upon work supported by the National Science Foundation under Grant No. 0831547, 0644450, and 1018964.

12. REFERENCES

[1] A. Blumberg and P. Eckersley. On locational privacy, and how to avoid losing it forever. *Electronic Frontier Foundation*, 2009. Technical Report.

[2] L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, pages 233–245, New York, NY, USA, 2007. ACM.

[3] I. Damgård, T. P. Pedersen, and B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '93, pages 250–265, London, UK, 1994. Springer-Verlag.

[4] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. *Pervasive Computing*, pages 152–170, 2005.

[5] B. Garner. *Black's Law Dictionary.* Thomson/West, Belmont, 2004.

[6] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132. ACM, 2008.

[7] K. Gratsias, E. Frentzos, V. Delis, and Y. Theodoridis. Towards a taxonomy of location based services. *Web and Wireless Geographical Information Systems*, pages 19–30, 2005.

[8] S. Halevi and S. Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '96, pages 201–215, London, UK, 1996. Springer-Verlag.

[9] R. Henry, K. Henry, and I. Goldberg. Making a Nymbler Nymble using VERBS. Technical report, Tech. Rep. CACR 2010-05, Centre for Applied Cryptographic Research, Waterloo, ON, Canada, 2010.

[10] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34:57–66, August 2001.

[11] L. Jedrzejczyk, B. Price, A. Bandara, and B. Nuseibeh. On the impact of real-time feedback on users' behaviour in mobile location-sharing applications. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, pages 1–12. ACM, 2010.

[12] W. Luo and U. Hengartner. Veriplace: a privacy-aware location proof architecture. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 23–32, New York, NY, USA, 2010. ACM.

[13] J. Manweiler, R. Scudellari, and L. Cox. SMILE: Encounter-Based Trust for Mobile Social Services. *CCS*, 2009.

[14] C. Mascetti, X. Wang, and S. Jajodia. Anonymity in Location-based Services: Towards a General Framework. In *MDM '07: Proceedings of the 2007 International Conference on Mobile Data Management*, pages 69–76, Washington, DC, USA, 2007. IEEE Computer Society.

[15] S. Micali. Simple and fast optimistic protocols for fair electronic exchange. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, PODC '03, pages 12–19, New York, NY, USA, 2003. ACM.

[16] R. A. Popa, H. Balakrishnan, and A. J. Blumberg. Vpriv: protecting privacy in location-based vehicular services. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 335–350, Berkeley, CA, USA, 2009. USENIX Association.

[17] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In *HotMobile '09: Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, pages 1–6, New York, NY, USA, 2009. ACM.

[18] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *Proceedings of the 2nd ACM workshop on Wireless security*, WiSe '03, pages 1–10, New York, NY, USA, 2003. ACM.

[19] B. Weiser. *After MetroCard Alibi, Murder Charges Are Dropped*, 2008 (accessed May 4, 2011).

[20] Z. Zhu and G. Cao. Applaus: A privacy-preserving location proof updating system for location-based services. In *INFOCOM, 2011 Proceedings IEEE*, pages 1889–1897, april 2011.