A Historic Name-Trail Service

Petros Maniatis
Computer Science Department
Stanford University
Stanford, CA 94305-9040
maniatis@cs.stanford.edu

Mary Baker
Computer Science Department
Stanford University
Stanford, CA 94305-9040
mgbaker@cs.stanford.edu

Abstract

[1] *In this work we consider the mobility of personal on-line identifiers. People change the identifiers through which they are reachable on-line as they change jobs or residences or Internet service providers. This kind of personal mobility makes reaching people on-line error-prone. As people move, they do not always know who or what has cached their now obsolete identifiers so as to inform them of the move. Use of these old identifiers can cause delivery failure of important messages, or worse, may cause delivery of messages to unintended recipients. For example, a sensitive email message sent to my now obsolete work address at a former place of employment may reach my unfriendly former boss instead of me.*

In this paper we describe HINTS, a historic name-trail service. This service provides a persistent way to name willing participants on-line using today's transient on-line identifiers. HINTS accomplishes this by connecting together the names a person uses along with the times during which those names were valid for that person. A correspondent who wishes to reach a mobile person can use an obsolete on-line name for that person, qualified with a time at which the on-line name was successfully used; HINTS resolves this historic name to a current valid on-line identifier for the intended recipient, if that recipient has chosen to leave a name trail in HINTS.

1. Introduction

The on-line world is inhabited by nomads. People change their on-line names as they switch Internet service providers (ISPs), either because they change jobs and they use the ISP of their employer, or because they switch to a better, cheaper, or more convenient service for their personal ISP. As a result, people accumulate a legacy of on-line identifiers that are, in most cases, dangling pointers into obscurity, making it hard to reach people. This problem is exacerbated by the trend for mobile people to garner a plethora of on-line identifiers pointing to a variety of accounts, applications and communication devices [23].

Unfortunately, sometimes an obsolete on-line identifier does point to something, and this can be much more dangerous than a dangling identifier. If the obsolete identifier belongs to my previous personal ISP, then some unrelated, unfortunate subscriber of that same ISP is bothered by my legacy of spam. If the obsolete identifier belongs to my previous place of employment, it might allow my sensitive communications to reach a potentially disgruntled former colleague or boss.

The problem stems from the fact that on-line identifiers *do not* belong to the people that they name; they belong to the organization that manages the associated name space. An employee of Clump Inc. does not own her Clump Inc. email address; that address belongs to the company itself, which loans it out to its employee for the duration of her employment there. Similarly, my yahoo.com address only names me as long as Yahoo! allows me to use it and maintains its service. As a result, a mobile person moves slowly from identifier to identifier in a potentially changing landscape of on-line names over which he has little control or authority.

This *personal-identity mobility* problem is analogous in many ways to physical mobility problems, and some of the solutions in that space may at first appear similar, including the application of persistent unique identifiers to both physical mobility and personal identity mobility. In the physical space, for example, Mobile IP [20] attempts to solve the problem of retaining reachability to a physical host that moves from one Internet connection to another, allowing that host to be addressed through a persistent "home" IP address, despite the mobile host's network interface attaching to networks that require other IP addresses. In the personal identity space, the promise of unique persistent personal

---

identifiers as a panacea has been made many times, but it is unlikely to succeed. In some cases, the proposed identification scheme requires a retrofit of the entire communications infrastructure of the Internet to work [12]. In other cases, the "unique personal identifier" is just another identifier assigned from a proprietary name space [6, 18, 19, 21].

The difference between these mobility spaces is the time scale over which the identifier must persist. Mobile IP is not intended to provide reachability to a single host at a persistent IP address over a decade despite its "home" network going out of business, yet this is essentially the problem we must solve for personal identity mobility. We believe that any practical solution to the personal identity mobility problem will need to take into account the passage of time itself.

In this paper we propose the **HI**storic **N**ame-**T**rail **S**ervice (HINTS). In our naming scheme, current on-line identifiers, email addresses and instant messaging account names remain the primary identifiers that people use in everyday communications. However, HINTS enables a person voluntarily to build a trail that connects all the identifiers that person uses over time into a *name history*. A correspondent can then safely name the person by using a potentially obsolete identifier qualified with a time at which that identifier was successfully used in the past. With this information the appropriate name history for a unique person can be found and followed to its latest entries, which provide a more recent way to reach the sought person.

HINTS offers mobile people control over forward pointers from their old to their current on-line identifiers. Even if an old name belongs in a name space maintained by a now-defunct organization, that name can still be resolved within HINTS to a current identifier for its past owner. This independence of a historical name from the provider of its associated name space promotes a robust, persistent way to refer to people, even when the practical names we use in everyday life are neither robust, unique nor persistent.

This is very much a work in progress: the goal of this paper is to describe the concept of historic naming as a means to address identity mobility, and to introduce a simple version of HINTS that can be deployed using today's technology. However, our current interests focus on designing a decentralized, more fault tolerant and secure version of HINTS, which we also briefly describe.

We proceed by providing an abstract model of the current personal on-line identification landscape, in Section 2. In Section 3 we introduce the basic concepts of HINTS, including the structure of the name space, the functionality of the service, and the implementation considerations involved. We evaluate the usability of this service in Section 4. In Section 5, we delineate what the service can and cannot do, given how personal on-line identification works today and elaborate on the next step for HINTS,

namely security and fault-tolerance. We discuss what additional requirements these properties place on on-line service providers, and we sketch the resulting enhanced design before concluding.

2. A Model For Personal On-line Identification

This section outlines the naming model on which the historic name-trail service is based. This model abstracts the way in which names are *currently* associated with people on-line. We also describe the components that provide naming services and explain the division of control and naming responsibilities between these services and their clients.

People need names on-line for a variety of purposes, such as authentication, communication, and authorization. In most cases, these names are simple: e.g., plain email addresses [22]. In other cases, on-line names may be two-tiered, starting with a primary application-unspecific name, which is used to look up application-specific addresses in a directory via an access protocol such as LDAP [24]. For simplicity, we take the former approach in this paper. Specifically, we use email addresses as the *primary* names of people on-line in our examples. However, the techniques we describe in later sections apply directly and without change to names in other applications, such as instant messaging, or to two-tiered naming environments.

The names people use are drawn from a multitude of independent *name spaces*. A name space is a set of all possible names that are or can be assigned by a single administrative entity, called a *name-space provider*. For example, Yahoo! Inc. is the name-space provider operating the `yahoo.com` name space: this is the set of all possible account names for Yahoo! Inc.'s services, such as web mail, instant messaging, calendaring, etc. In most cases, the name-space provider responsible for a name is easy to determine from the name itself. For example, `name@yahoo.com` is a name in Yahoo!'s name space. The computers responsible for maintaining a name space are the *name servers* of the provider. To find an authoritative name server for a name space, we can access the Domain Name Service [16], either through a distinct DNS resource record type, or through a straightforward name mapping (e.g., `names.thatschool.edu` for the `thatschool.edu` name space).

A person on-line may be addressable via multiple names drawn from different name spaces. For example, one may be addressed using names assigned by a school, by a personal Internet service provider, by web services such as Yahoo! and Hotmail, and by professional associations such as the ACM. These names can be used for distinct or overlapping purposes: professional, personal, and commercial.

The association of a name with a person follows the regulations defined by the corresponding name-space provider.

These regulations may dictate whether a name association is temporary or permanent, whether a name may be reassigned after its previous association is discontinued and, if so, the minimum amount of time between successive assignments of a single name. For example, Stanford University's Computer Science Department assigns names to its graduates for life, whereas ISPs assign names to clients for the duration of the pertinent service agreement. It is important to note, however, that the name-space provider is the ultimate controller of the name space, regardless of the "promises" it makes to those using its names. It is not uncommon, for example, that a claimed permanent name assignment has had to change for legal, social or political reasons. As an example experienced by one of the authors at Stanford University, the arrival of a new faculty member who desired an email address already assigned to a graduate resulted in a reassignment of that email address to the faculty member, in spite of the institution's guarantee of lifelong identifier assignments to its graduates.

In most cases today, a person exercises implicitly his "stewardship" of a name assigned to him, by being able to access the application for which the name is assigned. For example, in today's Internet, if I can read and send email as `name@yahoo.com`, then I am assumed to be the person to whom that name has been assigned. Though not bulletproof, this simplistic form of authentication is widely used for signing up in mailing lists, for signing up for web services or even for conducting business on-line. The name-space provider can revoke an association between a person and his name by stopping that person from accessing the named service; for example, the provider can cancel the associated web email account, or change the requisite password. Aside from intra-enterprise settings, not many name-space providers today offer rigorous security in how they assign names or how their clients assert ownership of their names. Section 5 describes a more secure naming model, very similar to an attribute certification model, that allows us to address the problem more comprehensively.

In the remainder of this paper, we use *mobile person* to mean someone who wishes to remain reachable despite identifier changes. A *correspondent* is someone who wishes to reach a mobile person. In the examples we use, Jane Mobile is a mobile person, and Dan Friend is a correspondent of Jane's.

3. A Historic Name-Trail Service

The primary goal of this work is to provide mobile people with a forwarding service that is available to help them remain reachable despite their identity mobility. We accomplish this by allowing a mobile person to determine to what his historic names—names qualified with a time when they validly named that person—point. We approach the problem first by taking the simplest path possible: a centralized, trusted service that operates similarly to most web services currently deployed and works with no cooperation from current infrastructure. We explore more sophisticated, secure and fault-tolerant possibilities in Section 5.

Reachability in the face of identity mobility requires the transfer of some of the naming "power" over a name from the associated name-space provider to the person to whom the provider assigns the name, within closely guarded *temporal* confines. Specifically, even though a provider can do anything it wishes to a name, its actions should not be applied retroactively: if Yahoo! assigns `jmobile@yahoo.com` to Jane Mobile from August 1999 to May 2000, it should not later be able to "change history" so as to strip Jane of her control over `jmobile@yahoo.com` for the indicated time period; Jane's authority over `jmobile@yahoo.com` from August 1999 to May 2000 must be *persistent*.

By imposing this persistence of authority, HINTS splits responsibilities between mobile people and name-space providers. Name-space providers are responsible for creating and destroying associations between names and people (in fact, between a name and the person who can access the service state for that name). People are responsible for acting on behalf of, and being reachable as, a particular name during the period that they have been assigned that name.

This separation of control enables the definition of a "virtual" global persistent name space, with most of the ambitious properties suggested in IdentiScape [12], such as persistence, controllability and human-centricity, but without requiring the creation and maintenance of a centralized name service for a global, flat name space implied by that system.

3.1. The Name Space

We define the HINTS name space by extending names with a continuous time designation. For example, to refer to the person named by the identifier `jmobile@yahoo.com` in March of 2000 we construct the identifier Historic[`jmobile@yahoo.com`, 03/2000]. More generally, the HINTS name space contains identifiers of the form Historic[`name@namespace`, time]. A HINTS name corresponds to a time-specific primary name, and is meaningful both while the associated primary name is valid (i.e., assigned), and after that primary name has been reassigned or obsoleted by its name-space provider.

The time component of the identifier defines a version of the chosen name space at a particular, coarse-grained time. The time component may designate an entire year, a year and a month, or a full date. The coarser the time component, the more likely it is that the HINTS name denotes multiple people to whom that identifier belonged during that time.
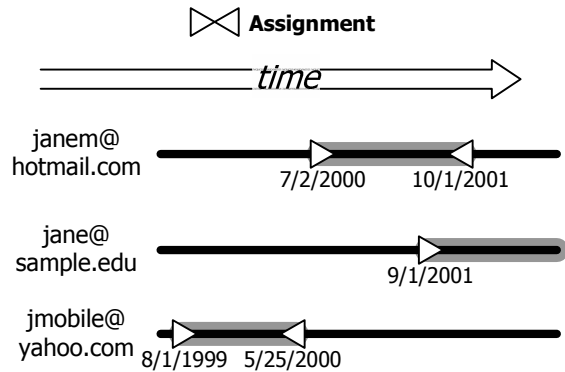
Figure 1. Jane's name history. Each of the three names shown has been held by Jane at some point in time. For each name, the thick gray line represents the time period during which Jane was assigned that name. For example, Jane held `janem@hotmail.com` **between July 2000 and October 2001.**

However, since identifiers change at "human" time scales, refining the time component to anything shorter than a day may be unnecessary. We assume that clocks of clients or providers are at least coarsely synchronized (e.g., they all agree on what day it is), but we address a more secure timing environment in Section 5.

3.2. A Personal Naming History

HINTS relies on a *name history* to resolve historic names. A name history links together HINTS names that refer to the same person. The objective behind maintaining a name history is to be able to reach a currently valid name-to-person association, starting with a now obsolete one. In the Jane Mobile example, the goal is to be able to obtain name `jane@sample.edu`, which is currently valid and is held by Jane, starting with Historic[`jmobile@yahoo.com`, 03/2000], even though `jmobile@yahoo.com` is no longer held by Jane. Figure 1 shows an example name history for Jane Mobile.

A name history is maintained as a sequence of historic records pertaining to a single name within a single name space, such as `jmobile@yahoo.com`. An on-line entity called a *name historian* maintains these records. The history of a name is modified due to either *assignment changes* or *linking changes*.

Changes in assignment affect to whom a particular name points. These are effected in provider-specific ways, such as changes in the password needed to receive service under a particular name, etc. Assignment changes can be detected by the name historian either directly, through an explicit notification from the name-space provider, or indirectly, through failure of a mobile person to respond to a challenge issued to his formerly assigned name. Since in this design we assume no cooperation from name-space providers, only indirect detection of assignment changes is possible. However, the historian can initiate indirect detections of assignment changes reactively, at the request of a mobile person.

Changes in linking affect how a single person assumes different on-line names over time. The historian emulates the concept of a "person" on-line by maintaining a trail of "personal manifestations," such as knowledge of the same secret password or of the secret portion of the same cryptographic key pair. Linking represents the intention of such an on-line person to be or not to be named by particular on-line names. The mobile person requests explicitly to be linked to or unlinked from a name by contacting the name historian with the appropriate request.

Assignments and links must both support the association of an on-line name with the historian's on-line representation of a mobile person. Specifically, to accept such an association, the historian must establish two facts: first that the mobile person wishes to assume that name, as communicated directly to the historian via a linking request; *and*, second, that the on-line person has access to that name, which the historian can detect indirectly by sending a challenge to that name via email (or any other applicable protocol) and expecting an appropriate response.

The historian periodically reestablishes both assignment and linking for an association. Since the name-space providers are not aware of the service, the historian has no recourse but to poll the name space periodically or in response to user requests, so as to establish whether the assumed mobile person still has control of his former name. Similarly, to avoid cases where a mobile person who becomes unavailable is assumed to be asserting control over a name by default for extended time periods, the historian expects periodically that person to reassert his links actively. Both of these periodic activities can be performed in an automated fashion between the historian and a user agent, and do not require the attention of the mobile person, except when he actually changes identities.

Figure 2 illustrates an *association record*, the basic building block of a historic name database. The specific record identifies Jane Mobile using her first and last names, although in practice a person is identified by the historian using an internal, private, implementation-specific name space for mobile person identification that no correspondent ever sees. An association record is active if it covers the present time (that is, its expiration time is in the future). An expired association record is archived and becomes immutable.

```
AssociationRecord{
  Name            : jmobile@yahoo.com
  Person          : Jane Mobile
  Start Time      : March 2, 2000
  End Time        : May 1, 2000
  Expiration Time : July 1, 2000
  Next Link       : June 29, 2000
  Next Assign     : June 29, 2000
}
```

Figure 2. An association record representing the association between Jane Mobile and the name `jmobile@yahoo.com` **from March 2 to May 1 of 2000. The historian considers the association valid until July 1 and expects a reconfirmation thereof on June 29. If that confirmation does not arrive, the association record is archived as ending on May 1st. Otherwise, the association record remains active, pushing its end time to July 1st and its expiration time 2 months later. The duration of the time-to-live period of 2 months is arbitrarily chosen here and can be modified per name space, per person, or per name historian.**

Figure 3 illustrates some of the historic associations that the name historian maintains for Jane. As shown in Figure 1, the name-space provider for `jmobile@yahoo.com` unassigns the name from Jane on May 25, 2000. As a result, even though Jane wishes to retain the name, as evidenced by the link extension to July 1st she requested, the historian stores an archived association record that only extends up to the last time both the name-space provider and Jane agreed on the association, May 1, 2000.

3.3. The Name Historian

In this section, we delineate the functionality available to users of HINTS. In particular, we describe the interface to the name historian itself, its design and some implementation details.

The primary objective of any name service is to support name resolution; in the HINTS context, this means that HINTS names must be resolved to primary names such as email addresses. If Dan Sender wishes to send email to Jane Mobile, he or his application must first resolve the HINTS name with which he previously reached her successfully (Historic[`jmobile@yahoo.com`, 03/2000]) to a currently valid primary name (`jane@sample.edu`).

To provide such name resolution, the name historian runs a centralized, trusted service maintaining name histories.
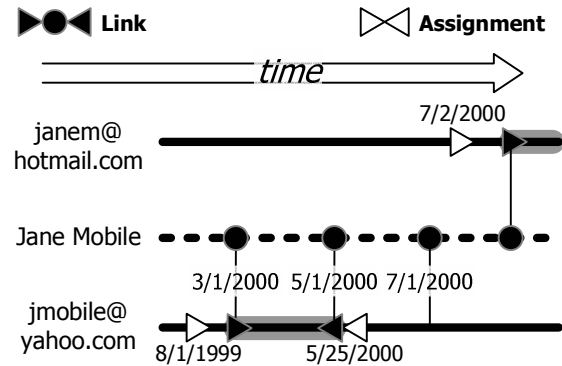
Figure 3. The names `jmobile@yahoo.com` **and** `janem@hotmail.com` **are linked to the historian's account for Jane Mobile (dashed line in the middle). The gray line segment covers associations that the historian has accepted.**

The historian is trusted to check the validity of the name histories it stores and to report on those histories when asked. Given the naming model we assume in this design (see Section 2), the historian is a powerful component in the network. We examine the shortcomings of having such a centralized design, and we propose alternatives in Section 5, based on stronger assumptions about name-space providers.

The historian operates a name space itself, consisting of the account names of its users. This name space, however, is only used to authenticate mobile people to the historian and need not be visible externally to the service. The historian manages its own name space similarly to how most web services manage their account name spaces: a client (mobile person in this case) signs up on-line and receives an account and a means of authentication, such as a password or an asymmetric key pair, for future exchanges with the historian.

To accomplish the task of resolution, the historian must also be able to receive link requests from mobile people as described in the previous section, and send and verify challenges to the holders of primary names to detect name assignment changes.

The resulting abstract interface to the history service is as follows:

- Create mobile person accounts.

- Request link changes (linking and unlinking) between a mobile person account and a primary name.

- Resolve a historic name to a currently assigned primary name.

The historian maintains a database of association records, as described in the previous section. The database contains two sorted indices, one on [*name*, *start time*] and another on [*person*, *start time*].

When a mobile person requests to be linked to a name, the historian contacts that name with a random number N as a challenge, by email or any other applicable protocol. If the email is returned with the appropriate response to the challenge (e.g., a repetition of N), then the historian considers the name assigned to the mobile person and creates a new association record. Algorithm 1 details this process. We justify the need for a challenge-response protocol with the assumption that, although spoofing the source address of an email message is technically trivial, it is much harder for an adversary to eavesdrop on email addressed to a particular address outside his control.

Algorithm 1 Link mobile person $\mathcal{A}$ to name a@A. H is the historian, M is the mobile person, now is the current time (on the historian's clock) and l is the maximum validity period of a link. M is assumed to have logged in as $\mathcal{A}$. $\Rightarrow$ denotes a network message. $\leftarrow$ denotes assignment. The process executes on H.

1: $H \Leftarrow M$: Link $\mathcal{A}$ to a@A
2: $N \leftarrow$ random nonce
3: $H \Rightarrow$ a@A: Request confirmation of assignment to $\mathcal{A}$ with nonce N
4: $H \Leftarrow M$: Assignment confirmation of a@A to $\mathcal{A}$ with nonce N
5: **if** confirmation is negative or none arrives **then**
6:    Do nothing and exit
7: $L \leftarrow$ the association record where name is a@A and person is $\mathcal{A}$ with the greatest start time
8: **if** none is found **then**
9:    Store into database [Association a@A, $\mathcal{A}$, now, now, $now + l$, $now + l$, $now + l$] {format is name, person, start, end, expiration, next link, next assignment}
10: **else**
11:    **if** $ExpirationTime(L) < now$ **then**
12:       Store into database [Association a@A, $\mathcal{A}$, now, now, $now + l$, $now + l$, $now + l$]
13:    **else**
14:       Update into database record L to [Association a@A, $\mathcal{A}$, $StartTime(L)$, now, $now + l$, $now + l$, $now + l$]

Links are severed in a similar manner, although no confirmation from the associated primary name is necessary; the reason for severing a link is exactly that the associated primary name is no longer under the control of the mobile person and, as a result, a confirmation from the primary name might not even be possible. However, a notification is sent to the primary name, to make it harder for an unauthorized user of the historian account to make changes unobtrusively. Algorithm 2 describes the process in more detail.

Correspondents query the historian for resolved historic

Algorithm 2 Sever the link from the historian account $\mathcal{A}$ to name a@A. H is the historian and M is the mobile person, logged in as $\mathcal{A}$.

1: $H \Leftarrow M$: Sever a@A from $\mathcal{A}$
2: $H \Rightarrow$ a@A: Notification of link severance from $\mathcal{A}$
3: $L \leftarrow$ the association record where name is a@A, person is $\mathcal{A}$ with the greatest start time
4: **if** $ExpirationTime(L) > now$ **then**
5:    Update into database record L to [Association a@A, $\mathcal{A}$, $StartTime(L)$, now, now, now, now]

---

names in a simple request-response protocol. The historian resolves a historic name by first mapping it to an internal account and then finding the latest still-valid association from that account to a primary name. Algorithm 3 has the details.

Algorithm 3 Resolve historic name Historic[a@A, t] to primary a name, if possible, or return failure otherwise. This is run on the historian H.

1: $L \leftarrow$ the association record where name is a@A and the start time s is the highest such that $s \leq t$
2: **if** no record is found **then**
3:    Return no results
4: $L' \leftarrow$ the association record where person is $Person(L)$ and the end time is greater than now
5: **if** no record is found **then**
6:    Return no results
7: Return $Name(L')$

Algorithm 3 assumes, for clarity and simplicity, that a single association record is active for a given name. This need not be the case in practice; the historian can return to correspondents as many currently active identifiers as the mobile person has registered.

4. Usability Concerns

From the point of view of customers of the name historian, several usability questions remain. We address some of these in this section but defer others until the description of the more secure decentralized history service in Section 5.

The first concern is how customers can choose which historic time to associate with an address for presentation to the name historian. Keeping track of the last time an email or other address was successfully used is arguably functionality that current email clients can easily offer, and is very similar in complexity to keeping track, for example, of the last time a web URL was used, which most web browsers do transparently by default. Many other communication applications, such as instant messaging or chat programs, can similarly track this information, or it might be preferable for

the applications to share and refer to a per-user database of this information.

Alternatively, correspondents can remember an approximate time period when they contacted a mobile person and construct a historic name accordingly, perhaps by referring to saved correspondence with the mobile person and extracting dates from that correspondence.

By extending the abstract interface of the history service, HINTS can even help correspondents who must rely on their own memory to construct a resolution request by presenting them with a history of name associations for the same name. For example, Dan Friend remembers having contacted Jane in the late 90's, but that is as specific a time frame as he can recall. If he can find out that `jmobile@yahoo.com` belonged to someone from 1995 to 2000, and then to someone else from 2000 to 2002, then Dan can reasonably choose Historic[`jmobile@yahoo.com`, 1999] to locate Jane Mobile. Listing association periods for a name is a straightforward operation helped by the database indices maintained, but privacy concerns could potentially make this an unattractive addition to the interface.

Unfortunately, if I never knew Jane Mobile before, there is no historic name that I can use to resolve Jane's current contact information. HINTS is not intended as a search engine for finding contact information for people I have never attempted to reach before. We believe that web search engines and currently deployed lookup services such as WhoWhere [11] are more appropriate for that purpose. However, old contact information gathered from such a web search becomes more valuable with HINTS, since it can then potentially be resolved to current contact information.

A second usability question is the frequency with which addresses should be resolved to determine whether they have been linked to a more recent address. The more frequently a name is resolved, the less likely it is that a communication is sent to an outdated or incorrect destination, but the more load this places on the history service and applications. While our experiences so far suggest that checking a name every week or so should be more than satisfactory, we do not have any real evidence to support this notion.

A third question that affects the potential success of HINTS is concern for the privacy of individuals. Entering identifiers into a HINTS name trail is voluntary; if a person wants obsolete identifiers to remain so, he need not tell HINTS about those identifiers. However, mobile people need to be aware of the risk that messages sent by correspondents to those old identifiers may reach someone else who now has control over that old identifier. If those messages are potentially sensitive, then mobile people may desire to see a widespread use of HINTS. Furthermore, we believe that identifiers inevitably "leak out" to the outside world; for example, it is not always convenient for an employee to avoid sending email from his company account to

an external destination even though this results in the destination having a record of his company-internal email address. True protection from spam and other threats is better performed using other techniques, such as the use of a personal communications proxy through which all incoming communications may be filtered [23].

Fourth, it would be very convenient if users beginning to use the history service could install into the historian the previous history of their identifiers, rather than merely beginning their naming history with their currently valid identifiers. Unfortunately, we see no reasonably secure but simple way to do this, as it is hard to prove in the present a user's authority over a past name when that past name is no longer valid for them. For instance, Dan Sender could claim past authority over Historic[`jmobile@lotsanames.com`, 1999] when in fact he had no such identifier assigned to him in the past. Securely setting up these past histories in the present would require the cooperation of, and trust in, the name space providers, which may no longer even be in business. On the other hand, the service in this current form is incrementally deployable, so that it is not necessary for everyone to begin using the service before it may be of use to some people.

Finally, many mobile people maintain multiple identifiers for the same application at the same time (i.e., many email addresses). HINTS historians must return all such currently active identifiers found during resolution for a given historic name; this might result in giving too much information to a potential correspondent. Although we believe this to be a real problem, we find its solution outside the scope of HINTS. Other solutions, such as the Mobile People Architecture [23], address the problem of personal routing adequately — i.e., forwarding messages to the appropriate application-specific address according to a mobile person's wishes — and are complementary to HINTS.

5. A Secure Name Historian

The name historian we describe in Section 3 achieves the primary goal of this work, reachability in the face of identity mobility. HINTS accomplishes this without the need for brand new globally visible, unique, persistent identifiers for every mobile person. Furthermore, it requires no retrofitting of current name-space providers' practices, requires no cooperation whatsoever from those providers, and can be deployed incrementally.

Unfortunately, this simple centralized approach has some shortcomings that can become significant as an increasing fraction of the world's business and fraud are transacted on-line. First, it relies on the honesty of the organization that operates the name historian. A centralized, centrally operated historian is a single point of failure, for failures including corruption, malfunction, connectivity disrup-

tion or going out of business. Second, the scheme offers no tangible assurances for the correctness of the information it gives out. For example, when a correspondent receives a primary name in response to a resolution request, he has to take the answer on faith; he has to believe that the historian did not invent the association and that the historian checked the validity of the association (i.e., issued a challenge to the claimed primary name and received a satisfactory response). Third, the scheme is only useful to those correspondents and mobile people who trust the historian. If no single historian is globally trusted, then there is no straightforward way to allow any correspondent to reach any mobile person.

In this section, we sketch an enhanced name historian design that offers the same functionality as what we present in earlier sections, but also alleviates the security and fault-tolerance concerns addressed above. We are currently focusing our design and implementation efforts on this stronger version of HINTS.

We believe that the increasing need for on-line security, the rising cost of identity theft and spoofing, and the evolving standardization projects for secure directory access will soon change how name-space providers do naming. We expect that providers will act much more like cooperative but competing certification authorities in the future; in fact, most large enterprises and even some governments [4] do perform naming or digital certification within their own realms in this more secure fashion.

Furthermore, we believe that unconditional trust in any centralized service is going to be increasingly difficult to digest for security-conscious mobile people.

We enhance our earlier design for a historic name-trail service in two major ways: First we use cryptography to prevent illegitimate changes in name-to-person associations, as can be caused by IP and email address spoofing, and eavesdropping on unsecured email. Second, we relax the need for trusting the name historian unconditionally, by employing secure time stamping [9] and undeniable attestation techniques [2], to limit the amount of unobtrusive damage a corrupt historian can cause to its clients' name histories.

We start by describing the stronger and slightly more cooperative name-space providers we need for this enhanced design, and then outline how our earlier concerns can be addressed.

5.1. Certification Authorities As Name-Space Providers

A certification authority is not much more than a name-space provider that accompanies the assignment of a name to a person with the issuance of a signed statement called an *identity certificate*, such as the one shown in Figure 4.

```
IdentityCertificate{
  Issuer      : yahoo.com
  Subject     : jmobile
  Key         : AB34D9...
  Start Time  : August 1, 1999
  End Time    : July 31, 2001
  Nonce       : 2C08A3...
  Signature   : <Issuer, subject, key,
                 times and nonce signed
                 using yahoo.com's
                 private key>
}
```

Figure 4. The identity certificate that links the name jmobile@yahoo.com **to the public key** AB34D9....

```
RevocationCertificate{
  Issuer      : yahoo.com
  Subject     : jmobile
  Key         : AB34D9...
  Start Time  : May 25, 2000
  Nonce       : 69C802...
  Signature   : <Issuer, subject, key,
                 start time and nonce
                 signed using yahoo.com's
                 private key>
}
```

Figure 5. The revocation certificate that breaks the link between the name jmobile@ yahoo.com **and the public key** AB34D9....

Note that Jane Mobile is not mentioned explicitly in the certificate. Instead, the provider assigns the name to the person who knows the secret portion of the public key AB34D9..., after having established that Jane Mobile holds that key. The kind of identity verification performed before a name-space provider assigns a name to a person's public key is provider-specific and out of scope in this paper, but traditionally includes out-of-band exchanges between that person and the provider.

The name-space provider can revoke an assignment by publishing a *revocation certificate*, such as that shown in Figure 5, or by publishing another identity certificate for the same name but a different public key.

Finally, the name-space provider refreshes an assignment by issuing new identity certificates for the same name and key before the previous assignment expires.

Besides this basic certification functionality, in this environment of higher security awareness we assume that name-space providers are cooperative with the history service, in

```
LinkCertificate{
  Name      : jmobile@yahoo.com
  Person Key : E51BB2...
  Start Time : March 2, 2000
  End Time   : May 1, 2000
  Nonce      : 6FC3F0...
  Signature 1: <Name, Person Key,
               Times and Nonce
               signed by the current
               key of the name>
  Signature 2: <Name, Person Key,
               Times and Nonce
               signed by the current
               key of the historian
               account>
}
```

Figure 6. A link certificate associating name jmobile@yahoo.com **with the person currently represented by key** E51BB2... **on 3/2/2000.**

```
SeveranceCertificate{
  Name      : janem@hotmail.com
  Person Key : E51BB2...
  Time      : September 25, 2001
  Nonce     : EE3BF4...
  Signature : <Name, Person Key,
              Time and Nonce
              signed by the
              person key>
}
```

Figure 7. A severance certificate breaking the link from the person currently represented by the key E51BB2... **to** janem@hotmail.com **on September 25, 2001.**

the sense that they notify the historian of name assignment changes by conveying to the historian newly issued identity or revocation certificates. For replay protection, we also assume that each such certificate issued by a name-space provider contains a nonce value, which in the simplest case is the time of issuance, but can also be a random number picked as a freshness challenge.

5.2. Certified Historic Naming

One straightforward way to address this harder naming problem is to make all information that the historian maintains signed, so as to prevent illegitimate modifications. To make sure that a corrupt historian cannot divert a name

trail by changing to what historian user accounts point, mobile people are represented in name histories by their signing key pairs, called *person keys*, similarly to the approach taken in SPKI/SDSI [7]. In this manner, the on-line representation of a person becomes "the person who knows the secret signing key."

Name assignments can be naturally represented with the identity and revocation certificates that name-space providers issue. Linking can be represented with similar *link* and *severance certificates* (see Figures 6 and 7, respectively), signed by mobile people (i.e., the holders of historian accounts) who lay claim on different primary names. Instead of storing a single association record per primary-name-to-mobile-person association, the historian stores all signed certificates delineating the assignment and the linking time periods that make up an association.

The historian can perform the same tasks as those in Section 3.3 with only little more difficulty. For example, to resolve a historic name, the historian must find an association from the historic name to a mobile person and then back from the mobile person to a currently valid name association, as detailed in Algorithm 3. Instead of just locating an association record, the historian must find the appropriate certificates justifying the association from the given historic name to a person and back to another primary name; not only does the historian have to return the answer it found—a primary name or a negative answer—but it must also return a *proof*: a set of certificates that support its response. The correspondent must then check that all the statements are signed correctly, and that they in fact do support a valid resolution, regardless of whether the answer was a primary name or a name-not-found response.

Unfortunately, there are three major issues that face clients of such a name historian: the short lifetime of digital signatures, the need for temporal ordering of historic records, and the need for a "closed," append-only historic database. We elaborate on all three in turn.

First, digital signatures have a limited lifetime. How does one make sure that a statement signed a few years back was correctly signed at the time, even though the signing key may have by now expired? Some work has been done to make signed documents usable even after the pertinent signing key has expired [8, 13]. In any case, the client must either trust the name-space providers themselves to maintain historic records of when they used what public key pair to sign their issued statements, or trust an external authority, such as the KASTS Key Archival Service [13], to maintain these records. Briefly, this service maintains a securely time stamped archive of the keys that different name-space providers use during their lifetimes, along with the times when those keys were used. A client who wishes to verify a signed statement, including the certificates we describe above, can look up the name of the provider and the ap-

```
DelegationCertificate{
  Issuer     : E51BB2...
  Delegate   : D91452...
  Time       : September 1, 2001
  Nonce      : D8306A...
  Signature  : <Issuer, Delegate, Time
               and Nonce signed using
               both issuing and delegate
               keys>
}
```

Figure 8. A delegation certificate, making the delegate key a continuation of the key trail of the issuing key.

propriate time period. The returned key can be then used to verify the signature on the statement. However, it is important to also ascertain that the statement was signed *while* the key found was still valid (i.e., before it expired or was revoked).

The ephemeral nature of digital signatures and signing keys also makes it hard to maintain the on-line representation of mobile people when the historian is not fully trusted, since mobile people must change the public keys that represent them regularly. People do this by issuing and submitting *delegation certificates* to the historian (see Figure 8), delegating their "personhood" from older keys to newer ones. Again, timing is of the essence, since a delegation must be performed before the previous key has to be abandoned, due to expiration or compromise. (Maintaining two independent sets of keys, one of which remains valid even if the other is compromised, can mitigate this problem.)

Second, any secure historic database must incorporate timing information on when its different records were archived. This is necessary for archived signatures, as described in the previous paragraphs, but also for authorization purposes, in key delegations or associations. For example, given a historic name, to establish the appropriate name association from assignment and linking certificates one must have knowledge of the relative temporal ordering of the issuance of those certificates. It must be shown that the historic name designates a time *after* an identity certificate and a linking statement created an association between the pertinent name and mobile person, but *before* those certificates expired or any potential revocation or unlinking statements came in effect. Relative temporal authentication of certificates in centralized [3] and decentralized [14] certificate archives can be invaluable here.

Briefly, relative temporal authentication uses one-way, collision-resistant hash functions, such as SHA-1 [17], to define the temporal ordering from earlier historic records to later ones. In a sense, a tamper-evident linked list is created

from all historic records, so that earlier records appear earlier in the list. Then regularly picked placeholder records from the list are published in a widely witnessed, secure, write-once publication medium, such as a high-circulation newspaper; a verifier who cares about when a particular record was appended into the historic database can trace the linked list backwards and forwards to find the previous and next, respectively, list links published in the newspaper, which in turn places the record in question in a rough time frame, that is, between the publication dates of two newspaper issues in the newspaper example. The collision-resistance and one-way properties of the hash function used to link records together guarantee that once a record has been placed in the historic database and the next link from the database has been "committed" on a newspaper, neither the maintainer of the database nor anyone else can tamper with history, changing when records appear to have been incorporated, adding or removing records, or modifying the contents of those records. This is the basic mechanism behind secure time stamping [9].

Third, it is important that the historian be unable to "forget" historic records that it has successfully accepted when they were submitted to it. For example, if a name-space provider submits a revocation certificate to the historian and the historian accepts it, it should be unable to deny the existence of such a revocation certificate convincingly when queried later. Undeniable attestation [2] is a cryptographic construct that allows clients to verify the historian's claimed existence or non-existence of certain records. The basic idea there is to construct a sorted data structure that allows *undeniable attestations* on its contents, that is, proofs that a particular element belongs or does not belong to the data structure. In the secure HINTS design, the database indices are, in fact, undeniable attesters.

An essential requirement is that attestation proofs are significantly shorter in size than the entire data structure itself. Consider, for example, a historic name database that holds a few billion certificates for many names and many mobile people. It would be extremely unrealistic to have to look through every single record in such a large database before being able to conclude no interesting revocation certificate has been archived there. The constructs from the work by Buldas et al. [2] and Maniatis and Baker [14] offer attestations with sizes logarithmic in the number of records stored, which, for extremely optimistic lifetime and popularity projections for a service like HINTS, never exceed roughly 20 KBytes; this is quite an acceptable size for records one expects to receive over the network once or twice a day.

Finally, while we describe an architecture for a secure name history that resolves old identifiers internally by following a trail of successively more recent identifiers, it is not necessary for the service to return any but the last iden-

tifier in the chain. In fact, privacy concerns make it preferable to return only this last identifier. Fortunately, the cryptographic proof of integrity for the name resolution results need not contain information about the identifiers between the historic name to be resolved and the resulting currently valid name.

6. Related Work

The literature regarding naming in distributed systems is vast, so in this section we confine ourselves to a sampling of systems and products that either combine dates with names or attempt to provide on-line names specifically for people.

Ours is not the only project to combine timestamps with names. Other examples of such work include the "tag" URI scheme [10], and the "duri" and "tdb" URN name spaces [15]. These schemes describe how to combine dates with names for several purposes, including uniqueness and persistence of identifiers, and the ability to mint identifiers without a stable authority assigning names. This work does not appear to address the need to resolve old names to currently valid names or the security and implementation issues we tackle, but the format of the identifiers themselves could very reasonably be used for HINTS names.

The new Internet domain names ending with `.name` are intended to provide a flat global name space for individuals that is not associated with particular employers and institutions. These names could also function as the primary names in our naming scheme. However, there are still many reasons why registrants for these names may end up changing their on-line identifiers over time. People who fail to pay their bills may lose access to their `.name` identifiers. People who change personal names, such as when getting married or taking stage names, may want to reflect this by changing their on-line identifiers as well. Moreover, these on-line identifiers may not be the only on-line identifiers registrants use. A user of a `.name` identity may also have email service through an employer, and it is hard to prevent identifiers from these other name spaces from "leaking out" in such a way that correspondents will not attempt to use them after they become obsolete.

IdentiScape [13] describes a flat global name space for people, in which identifiers do not necessarily reflect their personal names but instead can be a set of space-separated words in Unicode, thus permitting persistence of unique identifiers through simple lack of reuse. The hope is that this name space is large enough to accommodate many names per person, for everyone in the world, for the next one hundred years. Names resolved through IdentiScape point to personally controlled identity objects, which are repositories of access-controlled personal information, such as email addresses and phone numbers. The global IdentiScape names can thus point to a changing set of identifiers

associated with a person. Unfortunately, there is nothing to suggest that users will not change names within the IdentiScape name space or use identifiers from other name spaces as well, leaving identity mobility issues unsolved in IdentiScape.

PingID [21] is similar to IdentiScape in that an identity server, privately held by an owner of an identity, is responsible for authorizing (or not) the release of information according to who is asking. PingID's scope covers the usual applications: single sign-on, password management, and privacy management. PingID does not completely address identity mobility, since it does not allow reverse lookups from obsolete primary names to its own name space.

CommonName [6] is one of many on-line redirection services (others are OneName [19] and Novell's DigitalMe [18]). CommonName allows the use of common names or phrases instead of URLs or email addresses. It also allows the owner of a common name to set up different redirection schedules, for example, the CommonName "Jane Mobile" is redirected to Jane's personal email address during the weekend but to her work email address on a weekday. CommonName provides plug-ins for popular email applications and web browsers. A CommonName is assigned to an account for the duration of the account or the service. However, once an account is discontinued, its CommonName can be reassigned [1].

Classmates.com [5] is a service that allows people to register the school or college they attended, the military base at which they served, or the company for which they worked. This allows people who, for example, attended the same class at the same school to reconnect in the future. HINTS has very similar goals to this venture, but we take a purely on-line and more general approach; a correspondent need not know anything more than the mobile person's identifier used in the past, rather than first and last names and a class year. Although this limits the amount of heuristic weeding one can do to likely candidate names, it makes it easier for email and other applications to run identifiers through HINTS automatically and also allows correspondents throughout a recipient's history of identifiers to contact the recipient, even if the correspondents did not know the recipient when he graduated from Stanford University in 1957.

7. Conclusion

The extremely volatile environment typical to on-line services and their users makes identifier changes and the commensurate management problems a painful fact of online life. In this paper, we address the problem of identity mobility, by extending the names people commonly use in today's applications with a designation of a time when those names were successfully used.

We present a simple design for HINTS, a historic name-trail service, that allows mobile people to record and make available their movements through the "identifier landscape." Correspondents can follow those name trails from where a person has been (i.e., a name she used to have) to where that person is (i.e., a name she uses now). In this way, a correspondent can resolve a temporally qualified name into a name that is valid now; he can use that name to reach a mobile person whom he has not contacted in a long time, avoiding names that point to no one and reassigned names that point to the wrong person.

Finally, we describe how security can enhance HINTS in the increasingly naughty Internet to prevent illegitimate use of historic names and name-history corruption by a malicious service itself. We outline a design for such an enhanced HINTS system, which we hope to evaluate and make available soon.

8. Acknowledgments

References

[1] I. Buchanan. CommonName Ltd. Personal Communication, May 2002.

[2] A. Buldas, P. Laud, and H. Lipmaa. Eliminating Counterevidence with Applications to Accountable Certificate Management. *Jounal of Computer Security*, 10(3):273–296, 2002.

[3] A. Buldas, P. Laud, H. Lipmaa, and J. Villemson. Timestamping with Binary Linking Schemes. In H. Krawczyk, editor, *Advances on Cryptology (CRYPTO 1998)*, volume 1462 of *Lecture Notes in Computer Science*, pages 486–501, Santa Barbara, USA, Aug. 1998. Springer.

[4] Canada Customs and Revenue Agency. Address Changes Online. http://www.ccra-adrc.gc.ca/eservices/tax/individuals/aco/menu-e.html, Sept. 2002. An application of the Canadian *Government On-Line* initiative, to provide every Canadian citizen with a digital ID.

[5] Classmates. The World's Best Place To Reunite. http://www.classmates.com/.

[6] CommonName Ltd. CommonName: Your internet identity. Available at http://www.commonname.com/.

[7] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. RFC 2693: SPKI certificate theory, Sept. 1999.

[8] S. Haber, B. Kaliski, and S. Stornetta. How do Digital Timestamps Support Digital Signatures? *CryptoBytes, RSA Laboratories*, 1(3):14–15, Autumn 1995.

[9] S. Haber and W. S. Stornetta. How to Time-stamp a Digital Document. *Journal of Cryptology: the Journal of the International Association for Cryptologic Research*, 3(2):99–111, 1991.

[10] T. Kindberg and S. Hawke. The "tag" URI scheme and URN namespace. http://www.ietf.org/internet-drafts/draft-kindberg-tag-uri-04.txt, Sept. 2002. Internet Draft, work in progress, Internet Engineering Task Force.

[11] Lycos, Inc. WhoWhere? http://www.whowhere.lycos.com/.

[12] P. Maniatis and M. Baker. IdentiScape: Tackling the Personal Online Identity Crisis. Technical Report CSL-TR-00-804, Computer Systems Laboratory, Stanford University, Stanford, CA, USA, June 2000.

[13] P. Maniatis and M. Baker. Enabling the Archival Storage of Signed Documents. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST 2002)*, pages 31–45, Monterey, CA, USA, Jan. 2002. USENIX Association.

[14] P. Maniatis and M. Baker. Secure History Preservation Through Timeline Entanglement. In *Proceedings of the 11th USENIX Security Symposium*, pages 297–312, San Francisco, CA, USA, Aug. 2002.

[15] L. Masinter. "duri" and "tdb" URN namespaces based on dated URIs. http://www.ietf.org/internet-drafts/draft-masinter-dated-uri-03.txt, Apr. 2002. Internet Draft, work in progress, Internet Engineering Task Force.

[16] P. V. Mockapetris. RFC 1034: Domain names — concepts and facilities, Nov. 1987.

[17] National Institute of Standards and Technology (NIST), Washington, D.C., USA. *Federal Information Processing Standard Publication 180-1: Secure Hash Standard*, Apr. 1995.

[18] Novell, Inc. DigitalMe. Available at http://www.digitalme.com/.

[19] OneName Corporation. OneName. Available at http://www.onename.com/.

[20] C. Perkins. RFC 3220: IP Mobility Support for IPv4, Jan. 2002.

[21] PingID. Solving The Business Of Identity. http://www.pingid.com/.

[22] J. B. Postel. RFC 821: Simple Mail Transfer Protocol, Aug. 1982.

[23] M. Roussopoulos, P. Maniatis, E. Swierk, K. Lai, G. Appenzeller, and M. Baker. Person-level Routing in the Mobile People Architecture. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems*, pages 165–176, Boulder, CO, USA, Oct. 1999. USENIX Association.

[24] M. Wahl, T. Howes, and S. Kille. RFC 2251: Lightweight Directory Access Protocol (v3), Dec. 1997.