3rd Hot Topics in Privacy Enhancing Technologies
HotPETs 2010

Berlin, Germany

July 23, 2010

Welcome to HotPETs 2010!

HotPETs has been conceived as the "little brother" of the Privacy Enhancing Technologies Symposium – the venue to discuss the new hot ideas in Privacy Enhancing Technologies, including technical, philosophical, policy and practical issues. Earlier this year we were extremely excited to receive 15 submissions on all of the above topics. Fitting all of them into what was originally planned an afternoon session was clearly not possible, after some tough decisions, 11 remain for presentation in Berlin.

This is the third year that HotPETS is running alongside the PET Symposium. In the first year, in Leuven, we were trying out the concept of a small informal workshop alongside the conference presenting the less mature, early stage ideas. This HotPETs is memorable for an excellent panel debating the legal aspects of privacy and anonymity. At the second workshop in Seattle, we had 12 papers and an excellent invited talk by Alessandro Acquisti. We hope to keep up the standard of the previous two meetings and live up to the standard of the early PET workshops – informal, lively, inspiring, sometimes controversial and definitely thought provoking.

We thank everyone who submitted the papers, the presenters, the PETS program chairs Nick Hopper and Mikhail Atallah, and Hannes Federrath for patiently putting up with our disorganization.

We hope that everyone will have an enjoyable, inspiring, productive and thought provoking day. Our job is now done, the rest is up to the presenters and the audience.

Andrei Serjantov
Carmela Troncoso
Berlin, July 23, 2010

Program: Friday, August 7, 2010

9:30 Opening Remarks

9:35 HotPETs Session 1: Anonymity Systems

On the Optimal Path Length for Tor. *Kevin Bauer (University of Colorado at Boulder), Joshua Juen (University of Illinois at Urbana-Champaign), Nikita Borisov (University of Illinois at Urbana-Champaign), Dirk Grunwald (University of Colorado at Boulder), Douglas Sicker (University of Colorado at Boulder), Damon McCoy (University of California at San Diego)*

New Directions in Scalable Anonymous Communication. *Prateek Mittal, Nikita Borisov (University of Illinois at Urbana-Champaign)*

Compromising Tor Anonymity Exploiting P2P Information Leakage. *Pere Manils, Abdelberi Chaabane, Stevens Le Blond, Mohamed Ali Kaafar, Claude Castelluccia, Arnaud Legout, Walid Dabbous (INRIA)*

10:50 Break

11:15 Invited Speaker: Dr. Alexander Dix (Berlin Commissioner for Data Protection)

12:30 Lunch

2:00 HotPETs Session 3: Privacy in the Real World

Examining Privacy and Surveillance in Urban Areas: A Transportation Context. *Caitlin Cottrill (UIC)*

A Study on the Re-Identifiability of Dutch Citizens. *Matthijs R. Koot, Guido van 't Noordende, Cees de Laat (University of Amsterdam)*

Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL. *Christopher Soghoian (Indiana University), Sid Stamm*

3:15 Break

3:30 HotPETs Session 4: HotPETs Session 4: Location Privacy

Private Sharing of User Location over Online Social Networks. *Julien Freudiger, Raoul Neu, Jean-Pierre Hubaux (EPFL)*

A Unified Framework for Location Privacy. *Reza Shokri, Julien Freudiger, Jean-Pierre Hubaux (EPFL)*

4:20 Break

4:40 HotPETs Session 5: Identity Management

PseudoID: Enhancing Privacy for Federated Login. *Arkajit Dey (MIT), Stephen A. Weis (Google)*

Preliminary Thoughts on Privacy Supporting Binding of Biometrics to Credentials. *Jan Camenisch (IBM Research Zurich), Thomas Gross (IBM Research Zurich), Thomas S. Heydt-Benjamin (The Free Haven Project)*

5:30 Closing Remarks

On the Optimal Path Length for Tor

Kevin Bauer[1], Joshua Juen[2], Nikita Borisov[2],
Dirk Grunwald[1], Douglas Sicker[1], and Damon McCoy[3]

[1] University of Colorado at Boulder
{bauerk, grunwald, sicker}@colorado.edu
[2] University of Illinois at Urbana-Champaign
{juen1, nikita}@illinois.edu
[3] University of California at San Diego
dlmccoy@cs.ucsd.edu

Abstract. Choosing a path length for low latency anonymous networks that optimally balances security and performance is an open problem. Tor's design decision to build paths with precisely three routers is thought to strike the correct balance. In this paper, we investigate this design decision by experimentally evaluating several of the key benefits and drawbacks of two-hop and three-hop paths. We find that (1) a three-hop design is slightly more vulnerable to endpoint compromise than a two-hop design in the presence of attackers who employ simple denial-of-service tactics; (2) two-hop paths trivially reveal entry guards to exit routers, but even with three-hop paths the exit can learn entry guards by deploying inexpensive middle-only routers; and (3) three-hop paths incur a performance penalty relative to two-hop paths. Looking forward, we identify and discuss a number of open issues related to path length.

1 Introduction

Design decisions made by low latency anonymizing networks frequently involve achieving a correct balance between security and performance. For example, Tor does not employ cover traffic or add intentional delays in order to ensure performance that is sufficient to support interactive applications such as web browsing. However, this decision has increased Tor's vulnerability to end-to-end traffic correlation. Another key design decision is path length. Tor employs a decentralized architecture of precisely three routers to mitigate any single router's ability to link a source and destination. However, three-hop paths have a performance cost. In this paper, we seek to better understand the security and performance trade-offs related to path length design decisions.

Tor's design — like most low latency anonymizing networks — is vulnerable to end-to-end traffic correlation attacks. If the endpoints are compromised, an adversary can apply any one of many known traffic analysis attacks [1–8] to correlate the source and destination. Conventional wisdom indicates that three-hop paths achieve an appropriate balance between security and performance. However, two-hop paths may be attractive to users seeking improved performance, though it is unclear what security trade-offs two-hop paths may incur.

Through analysis, simulation, and experiments performed on the live Tor network, we critically evaluate the advantages and disadvantages of two-hop and three-hop paths from security and performance perspectives. In addition, we identify and discuss a variety of open issues related to the security and performance of different path length choices.

Path length and security. We consider an adversary who uses selective disruption tactics (as in [4, 9–11]) to force circuits to be rebuilt in the event that a malicious router participates in a circuit that is not compromised. Through simulation of Tor's router selection algorithm fueled by real router data obtained from Tor's trusted directory servers, we show that three-hop paths are up to 7% more vulnerable to path compromise than two-hop paths under the same attack.

One potential disadvantage of a two-hop design is that exit routers can trivially discover clients' entry guards, since they communicate directly. We empirically demonstrate that malicious exit routers can identify clients' entry guards even with three-hop paths by deploying middle-only routers that employ selective disruption. Our results show that an adversary with only ten malicious exit routers and 50 middle-only routers can learn the entry guards for nearly 80% of all circuits constructed. We also analyze the potential to identify clients uniquely through knowledge of their entry guards.

We lastly perform experiments on the real deployed Tor network to show that low cost timing-based traffic analysis techniques that link circuits by their circuit building messages can be highly successful in practice. On the live Tor network with a workload of real user traffic, we show that timing analysis can successfully link 97% of the traffic from clients that we control even before any data traffic is sent.

Path length and performance. In addition to an analysis of path length from a security perspective, we show that shorter paths offer better performance as perceived by end-users in terms of download time. We perform an analysis of typical web browsing behavior and demonstrate that users will see fewer circuit failures with two-hop paths, which results in faster web page loading and an improved user experience.

2 Tor: The Second Generation Onion Routing Design

Tor is the second generation onion routing design providing a low latency anonymizing overlay network for TCP-based applications [12]. One of Tor's primary design goals is to ensure low enough latency to facilitate interactive applications such as web browsing and instant messaging. Tor's system architecture consists of *Tor routers*, which are volunteer-operated servers, *directory servers* that organize information about the Tor routers, and *Tor proxies* (or clients). Tor routers may be configured by their operators to allow connections only to other Tor routers, or to allow exit connections to arbitrary hosts on the Internet. Tor clients query one of the authoritative directory servers to obtain a signed list of the available Tor routers, their public keys, bandwidth advertisements, exit policies, uptime, and other flags indicating their entry guard status and other information.

To establish an anonymous virtual connection through the Tor network to a desired destination, the client must first choose a path (or circuit[1]) of precisely three Tor routers and establish a shared symmetric key with each, using authenticated Diffie-Hellman and a telescoping key agreement procedure. Once the circuit has been created, the client encrypts their data in 512 byte units called *cells* with each key in a layered manner and forwards these cells to the first router in the circuit. Upon receiving a cell, each router removes its layer of encryption using its symmetric key shared with the client and forwards the cell to the next router in the circuit. Finally, after the exit router removes the final layer of encryption, it establishes a TCP connection with the destination and sends the client's data. More details can be found in the Tor Protocol Specification [13].

2.1 Tor's Router Selection Algorithm

The manner in which Tor clients select their routers has serious implications for the network's security properties. For example, if a client chooses malicious routers, then they may experience lost anonymity. At Tor's inception, it was composed of only a few high-bandwidth routers and had few users, so it was sufficient to select routers uniformly at random. As the network grew in popularity and router bandwidth diversity, it became necessary to balance the traffic load over the available bandwidth resources, which can be achieved by selecting routers according to their bandwidth capacities. However, Tor routers self-advertise their bandwidth capacities. It has been shown that an adversary can falsely advertise high bandwidth claims to attract traffic and increase their ability to compromise circuits [4, 14].

Recent work has proposed methods to securely verify these self-reported bandwidth claims [15]. Active measurements have been integrated into the Tor network's directory servers to verify routers' bandwidth claims [16]. However, the security of these active measurements has yet to be evaluated.

Tor's router selection algorithm [17] chooses routers with the following constraints:

- A router may only be chosen once per path.
- To prevent an adversary who controls a small network from deploying a large number of routers, each router on a path must be from a distinct /16 subnet (in CIDR notation).[2]
- Each router must be marked as `Valid` and `Running` by the authoritative directory servers.

[1] The terms "path" and "circuit" are used interchangeably throughout this paper.

[2] Tor also allows an operator of many relays to set an advisory `Family` flag that will ensure that their nodes are not chosen twice per path.

– For non-hidden service circuits, each router must be marked as Fast, indicating that the router has at least 100 KB/s of bandwidth or is within the top 7/8 of all routers ranked by bandwidth.
– The first router on the path must be marked as a Guard by the authoritative directory servers. Clients select precisely three entry guards to use on their circuits, and choose new guards periodically.
– The last router on the path must allow connections to the destination host and port.

For general purpose circuits, Tor's path selection algorithm weighs router selection by each router's perceived bandwidth capacity. In order to ensure that there is sufficient exit bandwidth available, the bandwidth of Exit routers is weighted differently depending on the fraction of bandwidth that is available from non-Exit routers. Suppose that the total exit bandwidth is E and the total bandwidth available is T. If $E < T/3$, then Exit routers are not considered for non-exit positions. Otherwise, their bandwidth is weighted by $(E - (T/3))/E$ [17].

Entry guards were introduced to Tor's design to mitigate the threat of profiling and the predecessor attack [14]. Entry guard nodes have special uptime and bandwidth properties. A router is marked as a Guard by the authoritative directory servers only if its mean time between failures is above the median of all "familiar"[3] routers and its bandwidth is greater than or equal to 250 KB/s [18]. By default, clients choose precisely three entry guards to use for their circuits. To ensure that there is sufficient guard bandwidth available, guard node selection is weighted by $(G - (T/3))/G$, where G is the amount of available guard bandwidth. If $G < T/3$, then guard nodes are not considered for non-guard positions [17].

3 Security Analysis

In this section, we study the security implications of Tor's path length. First, we evaluate how an adversary's ability to compromise circuits varies between two-hop and three-hop paths. Second, we explore how two-hop paths reveal circuits' entry guards and discuss the potential for adaptive surveillance attacks. We also describe an attack where an adversary with few exit routers and comparatively many middle-only routers can identify the entry guards on a large fraction of circuits. Third, the amount of information about clients that is revealed by entry guard knowledge is analyzed. Finally, we evaluate a low cost traffic analysis technique that links circuits using only circuit building messages on the live Tor network. This attack's success re-iterates the fact that three-hop paths provide no protection whatsoever against these attacks.

[3] A router is "familiar" if one-eighth of all active routers have appeared more recently than it [18].

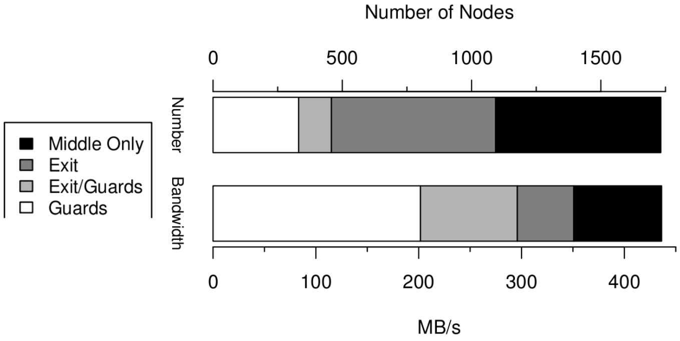


Fig. 1. Distribution of routers used in simulations, as gathered from a directory server

3.1 Selective Disruption and Path Length

To understand the relationship between path length and circuit compromise, we simulate Tor's current router selection algorithm (described in Section 2.1) using router data collected from an authoritative directory server.

Simulation setup. We adopt a simulation methodology similar to Murdoch and Watson [19] in which malicious routers are added to the network and circuit compromise statistics are computed. In particular, we simulate 1 000 clients who choose precisely three entry guards and each construct 100 circuits of length two and three that are suitable for transporting HTTP traffic (port 80).[4] Next, a variable number of malicious routers between 5 and 50 are injected into the network. Each malicious router has 10 MB/s of bandwidth,[5] is marked as a `Guard`,[6] allows port 80 to exit, and is operated on a distinct /16 subnet.

A snapshot of all Tor routers was obtained from an authoritative directory server on January 6, 2010. This snapshot (summarized in Figure 1) consists of 1 735 total routers marked as `Valid` and `Running`. Note that the snapshot has sufficient entry guard and exit bandwidth such that both entry guards and exit routers may by used for any position of the circuit, provided that they have the appropriate flags.

Results. Figure 2 shows the fraction of circuits that are compromised as the number malicious routers and amount of adversary-controlled bandwidth increases. First, note that for attackers that do not apply selective disruption, the circuit compromise rate is directly proportional to the adversary's resource in-

[4] We simulate HTTP exit traffic because prior work found it to be the most common type of traffic by connection on the real Tor network [20, 21].

[5] Currently the largest believable bandwidth value.

[6] Obtaining the `Guard` flag only requires that the router demonstrate stability for a relatively short period of time. We anecdotally found that a new router on a high bandwidth link can obtain the `Guard` flag after running for roughly seven days.

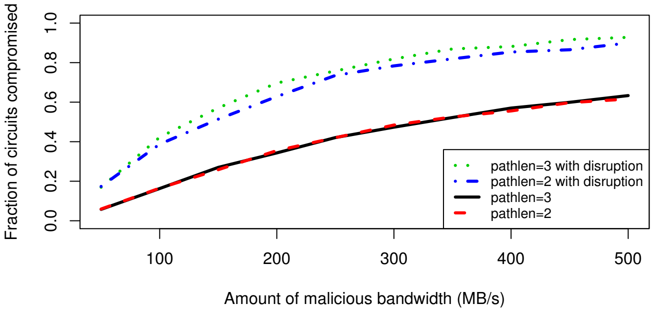


Fig. 2. Fraction of HTTP circuits compromised

vestment: 50 attackers with 10 MB/s of bandwidth each control over half of the network's bandwidth, but are able to compromise just over half of all circuits. Also, the compromise rate is the same regardless of whether three- or two-hop paths are used for each malicious router configuration since the attacker gains no advantage from participating in circuits that are not compromised.

Now consider an adversary whose routers selectively disrupt circuits on which they're chosen that they cannot compromise. Regardless of path length, if the client has the misfortune of choosing three malicious entry guards, then due to selective disruption, their circuits are always compromised. If a client chooses no malicious entry guards, then their circuits are never compromised. Clients that chose one or two malicious entry guards experience circuit compromise with a certain probability. For example, when there are 10 malicious routers and clients use three-hop paths, 38% of clients choose no malicious guards, 47% choose one malicious guard, 13% choose two malicious guards, and 1% choose three malicious guards. Of the clients that choose one or two malicious guards, their circuits are compromised 63% and 85% of the time, respectively. Note that entry guards offer some degree of protection against circuit compromise, since clients that choose no entry guards are safe and the threat increases with the selection of additional malicious entry guards.

As shown in Figure 2, across all malicious router configurations the fraction of circuits compromised is up to 7% higher for three-hop paths relative to two-hop paths. With three-hop paths, when the client selects one or two malicious guards, their circuits are disrupted when they use a non-malicious guard and a malicious middle or exit router (or both). In this case, the circuit is rebuilt and the client may use a malicious guard. With two-hop paths, if the client does not use a malicious guard, then only the exit position can disrupt non-compromised circuits. Since three-hop paths have one additional position from which to disrupt circuits, they exhibit a slightly higher compromise rate relative to two-hops. However, it is unclear if this small increase in the risk of endpoint compromise is a sufficient danger to justify a change in Tor's default path length.

3.2 Adaptive Surveillance Attacks and Path Length

In addition to the threat posed by compromised routers, Tor's three-hop design is ostensibly vulnerable to attacks whereby a powerful ISP or government adversary can monitor a targeted circuit's endpoints' networks to identify the traffic's source and destination. This attack is believed to be difficult with three-hop paths because it relies on a circuit having the misfortune of choosing an entry and exit router that reside within monitored networks. Since Tor achieves network diversity in its route selection in practice [22], this attack would require collusion by many network operators.

However, with two-hop paths exit routers can directly observe the entry guards. Suppose that a client builds a circuit through an adversary-controlled exit router, but uses a non-malicious entry guard. Since the exit router knows the client's entry guard, they could adaptively demand network logs from the entry guard's network through legal channels or other forms of coercion. While this attack requires a powerful adversary and consequently may be unlikely, two-hop paths make the attack technically feasible which may encourage malicious exit routers (or their network operators) to implement it.

While two-hop paths enable adaptive surveillance attacks by leaking entry guards to the exit router, adaptive surveillance is possible even with Tor's current three-hop design. If an adversary deploys both malicious exit routers and malicious middle-only routers, they can collude to identify the entry guards used for every circuit on which they are used for the middle and exit positions. We next show that an adversary who controls few exit routers and comparatively many malicious middle-only routers can identify the entry guard used for a large fraction of circuits.

Simulation setup. Experiments are conducted where an adversary controls only ten exit routers configured to exit HTTP (port 80) traffic and injects 50 and 75 middle-only routers to the Tor network summarized in Figure 1. All malicious routers have 10 MB/s of bandwidth and now disrupt circuits when they do not control both the middle *and* exit positions. We simulate 1 000 clients who each build 100 circuits.

This attack strategy has a low cost for the adversary, since they do not need to demonstrate router stability (as is necessary to obtain the guard flag). In addition, all malicious middle-only nodes could be deployed on the same /16 network and all malicious exit routers could be deployed on a second /16 network. Thus, the resources required to launch this attack are modest.

Results. For an attacker with 10 malicious exit routers and 50 middle-only routers, the adversary can identify the entry guard for 79% of all circuits constructed. When the attacker deploys 75 middle-only routers, they discover the client's entry guard for 85% of all circuits. For these circuits, the adversary could apply pressure and potentially coerce the entry guard (or its network operator) to reveal the identity of the client.

Table 1. Daily statistics for clients per entry guard and entropy estimates

No. of Samples	**Minimum**	**Maximum**	**Median**	**95% Confidence Interval**
$n = 737$	680	164 000	8416	(24 104, 27 176)
Entropy	8.20	0.29	4.57	(3.05, 2.88)

Perhaps the most compelling argument in favor of three-hop paths for Tor is that the middle router hides the entry guards from exit routers. By using a middle router, a malicious exit typically knows only information about the client that is leaked by their applications. However, if malicious exits collude with middle routers who can observe the entire circuit, it becomes feasible for the exit to learn a large fraction of the total client population's entry guards.

To make matters worse, deploying a relatively large number of middle-only routers causes a global change in Tor's router selection process. In these experiments, when 50 middle-only routers are introduced, the aggregate entry guard bandwidth G and aggregate exit router bandwidth E no longer satisfies $G \geq T/3$ and $E \geq T/3$, respectively, where T is the total bandwidth. In this network configuration, exit routers may only be used for the exit position and entry guards may only be used for the guard position. This enables the adversary to focus their few exit routers toward occupying the exit position and maximize their ability to conduct adaptive surveillance.

3.3 Entry Guard Linkability

With a two-hop design, we know that malicious exit routers can discover clients' entry guards. It is possible that clients' entry guards may be uniquely identifying or place clients into small anonymity sets. To understand the extent to which knowledge of clients' entry guards may be identifying, we next analyze publicly available data on entry guard usage from the Tor Metrics Project [23]. From this data, eleven entry guards provide information about the number of clients that they observe over time.[7] Table 1 presents a statistical summary of the number of clients observed by each entry guard on a daily basis. With this data, we can estimate how much identifying information is leaked through knowledge of a client's entry guard.

We apply the standard entropy metric from information theory [24] to measure how much information is revealed about a user by their entry guard selections. The total number of unique Tor users per day is currently estimated to be between 100 000 and 300 000 [25]. Thus, without any additional knowledge, 17.61 bits of information are necessary to uniquely identify a Tor user.[8] Now suppose that a malicious exit router knows a particular client's entry guard. On average, roughly 25 000 clients use the same entry guard, so this knowledge leaks only

[7] To preserve users' privacy, this data is aggregated by country of origin, quantized by multiples of eight, and compiled daily.

[8] This analysis assumes that 200 000 unique clients use Tor each day.

2.96 bits of information about a user's identity. Even in the worst case when a client shares a guard with as few as 680 other clients, only 8.20 bits are revealed (the full entropy results are shown in Table 1).

If an attacker knows all three of a particular client's entry guards, the client may be more identifiable since a choice of three guards may be significantly more unique than a single guard. While it is usually difficult to link a client across multiple entry guards, if a client inadvertently identifies herself — perhaps by logging-in to a website or using an application that does not support SSL/TLS — over time her full set of entry guards could be leaked to a malicious exit router. Tor clients do, however, expire their entry guard selections periodically, which may help to protect users from this type of profiling.

We should also point out that, even with three-hop paths, linkability pitfalls still exist in Tor. First, a Tor circuit can be used by several connections, which can be trivially linked by the exit router. Second, the predecessor attack shows that the entry guards used by a client can be learned after $O(1/(f_m f_e))$ circuit constructions on average, where f_m and f_e are the probabilities that a malicious router will be chosen as the middle and exit router, respectively [26]. Selective disruption and other techniques [27] can be used to increase the speed of such attacks.

3.4 Low Resource Traffic Analysis on the Live Tor Network

Prior work has shown that end-to-end traffic correlation attacks launched against low latency anonymous networks can achieve near perfect accuracy [2]. To support interactive or delay-sensitive applications, Tor does not explicitly delay or batch messages to help defend against end-to-end traffic correlation attacks. Consequently, Tor's design assumes that these attacks can achieve high accuracy in practice. In fact, such an attack has been proven effective against the live Tor network in 2006 [14]. Since then, a low resource traffic analysis technique has been proposed that uses only circuit construction messages to link a source and destination before any data is sent [4]. This approach allows low bandwidth attackers to maximize the number of circuits compromised, but this low cost attack has yet to be validated on the live Tor network. We next evaluate this traffic analysis approach on the live Tor network.

Experimental setup. We deploy two Tor routers[9] hosted on a 100 Mb/s network link onto the live Tor network. Each router has a distinct configuration: (1) One Tor router is configured as a non-exit and after roughly ten days of uninterrupted operation, it obtained the `Guard` flag from the authoritative directory servers. (2) A second Tor router is configured with the default exit policy.[10] During their operation, both routers sustained roughly 3 MB/s of traffic.

To evaluate the expected success of traffic analysis, we operate our own Tor clients and attempt to link their circuits to their destinations. Upon building

[9] These routers ran software version `Tor 0.2.1.20`.

[10] Ports often associated with outgoing e-mail, peer-to-peer file sharing applications, and high security risk services are blocked.

a circuit, each client downloads www.google.com, tears down the circuit, and repeats this procedure. To preserve users' privacy, we ignore traffic at the entry guard that is not produced by one of our clients.[11] Note that we do not retain any linkable data nor do we attempt to deanonymize any other clients but our own.

Traffic analysis methodology. We apply a traffic analysis technique in which circuits are linked by their circuit building messages before the clients send any data cells. This approach leverages the fact that Tor's circuit establishment procedure sends a fixed number of circuit building messages in an identifiable pattern.

Briefly, circuit linking via circuit building messages works as follows. First, our entry guard ensures that the circuit building request is from a client and not a Tor router. Next, it is necessary to ensure that the next router for our entry guard is the same as the previous router for our exit router (with a tight time difference). Finally, the circuit building messages for the entry, middle, and exit routers should occur in increasing chronological order. More details about our linking procedure can be found in [4].

Results. On the live Tor network, our clients build a total of 1 696 circuits that always use our entry guard. Of these 821 circuits use our exit router and 875 circuits use a different exit router.[12] The middle routers are chosen according to Tor's default selection algorithm. Through traffic analysis, we link their circuits with 97% accuracy, 0.6% false negatives (6 false negatives in total), and 6% false positives (52 false positives in total). We regard these results as a lower bound on attainable traffic analysis success, as it should be possible to increase the accuracy by also using data cells to link circuits. Also, we observe that circuits that use a popular (*i.e.,* high bandwidth) middle router tend to be more prone to false positives. Thus, an attacker who sees a positive result with a low bandwidth middle router can be more confident in the result. Given the high accuracy and the relatively easy manner in which the traffic analysis was conducted, we confirm that three-hop paths offer no protection against low cost timing attacks.

4 Performance Analysis

We have already studied Tor's path length from a security perspective. We next examine its performance implications. Since the vast majority of Tor traffic is interactive web browsing [20, 21], we investigate the performance benefits of a two-hop design from a web browsing end-user's perspective.

Experimental setup. In order to understand Tor's performance in a manner that reflects the quality of a user's experience, we simulate real clients accessing the 15 most popular websites[13] over Tor version 0.2.1.24 with Polipo version 1.0.4 and measure the download times. Experiments are conducted in February

[11] This data collection procedure was approved by the University of Colorado's Institutional Review Board.

[12] This setup allows us to count the number of false positives that occur during linking.

[13] We consider the 15 most popular websites according to http://www.alexa.com.

2010 over the course of four days.[14] Circuits are constructed according to Tor's default router selection algorithm and the Firefox browser downloads one of the web pages.

In the event of a circuit failure, Firefox's default behavior is to time-out after two minutes. However, real users may be impatient and explicitly force the browser to reload the page by pressing the "refresh" button. Prior work has found that users of low latency anonymous networks tend to tolerate no more than four seconds of latency [28]. Thus, in the event of a circuit failure, we assume that users wait not the full two minutes for their browsers to time-out, but precisely four seconds before explicitly reloading the page.

Results. A CDF of download times for two- and three-hop paths is shown in Figure 3. For three-hop paths, half of all web page downloads take longer than 12 seconds, while for two-hop paths, half complete in over 8 seconds. The mean download time for three-hop circuits is over 28 seconds, which is twice the expected download time for traffic over two-hop circuits (14 seconds).

We observe that circuit failures tend to be a significant cause of the additional expected download times with three-hop circuits.[15] 21% of circuits fail with three-hop paths, but only 15% of circuits fail with two-hop paths. The observed unreliability of three-hop circuits may contribute to high download times, as some users may wait unnecessarily for their browser to time-out. In these experiments, we assume that the user can quickly identify that their session has stalled (*i.e.,* by observing that no web content has loaded) and refresh the page after waiting four seconds for content to appear. However, some users may take significantly longer to launch another web request.

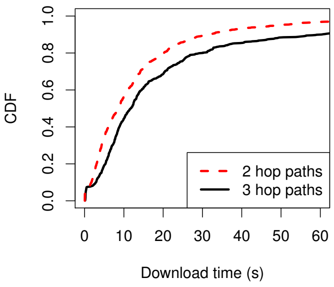


Fig. 3. Time to download popular web pages

5 Discussion

Having discussed Tor's path length from security and performance perspectives, we next discuss a variety of open issues related to path length.

[14] While prior studies have found that Tor's performance varies by time of day [28,29], a more recent study did not identify diurnal patterns in Tor's traffic load [20]. Thus, we do not believe the time of day to be a significant factor that effects performance.

[15] A circuit failure occurs when a circuit fails after the circuit has been established and at least one data stream has been attached. This is different than a circuit building failure, where a chosen circuit cannot be built.

5.1 User-Configurable Path Lengths

Since two-hop paths offer better performance, it may be tempting to allow users who value performance over security to use two-hop paths while users who need stronger security may use three-hop paths. Suppose that most users value performance and consequently, Tor chose a default path length of two hops. Security-conscious users could optionally use three hops to take advantage of the additional security that three-hop paths offer against adaptive surveillance. However, clients who choose to use longer paths may be identified as desiring additional security, which alone could draw an adversary's attention. Furthermore, it has been argued that most users tend to keep default options, even when the defaults may not be optimally suited to their needs [30]. Allowing users to configure their own path lengths assumes that users understand the full security implications of their choice, which is unlikely, particularly for novice users. Thus, all users should be encouraged to use the same path length.

5.2 Potential Liabilities for Exit Routers

Beyond the potential risks of identifying users who desire stronger security by their path length choice, two-hop paths could be a liability for exit router operators. With three-hop paths, exit routers know nothing about clients other than what may be revealed by their traffic. However, with two-hop paths, exit routers are exposed to clients' entry guards; thus, they are no longer agnostic with regard to the clients whose traffic they transport. Exit routers could be presented with subpoenas to reveal entry guard information to governments or law enforcement agents, which increases the risks associated with operating an exit router. Since Tor's exit bandwidth is relatively scarce yet essential to the network's ability to function properly, liabilities for exit router operators should be minimized to attract additional exit routers.

5.3 Secure Bandwidth Estimation

The attacks that we describe in Sections 3.1 and 3.2 are particularly dangerous in the absence of secure bandwidth verification, since malicious routers could otherwise inflate their perceived bandwidth to attract traffic. With secure bandwidth estimates in place, it will no longer be possible to carry out these attacks with few resources. However, it is important to remember that such attacks are still within reach of medium-to-large organizations, or even determined individuals: at current hosting rates, running a 10 MB/s node for one week (long enough for a node to be declared a guard) can cost less than $1 000;[16] thus, the financial resources required to attack the network successfully are moderate at best. Additionally, attackers may be able to insert their own high-bandwidth nodes into the Tor network by compromising computers at well-provisioned institutions.

[16] See, for example, `http://aws.amazon.com/s3/`.

5.4 Does a Two-Hop Design Discard Many Routers?

Many Tor routers are not configured to allow exit traffic and are not fast and/or stable enough to be an entry guard. These routers are only used for the middle position. We next consider whether a two-hop design would discard a significant number of middle-only routers and their collective bandwidth.

From the directory server snapshot analyzed in Section 3, we find that 639 routers may only be used for the middle position. These routers collectively contribute about 85 MB/s of bandwidth. To understand how bandwidth is distributed among non-exit and non-guard routers, Figure 4 shows a CDF of these routers' bandwidth contributions. Half contribute less than 50.3 KB/s each and only 11% offer the 250 KB/s necessary to meet the bandwidth criterion for the guard flag. These higher bandwidth routers collectively contribute 54.3 of the 85 MB/s of middle-only bandwidth. If stable enough, they could eventually obtain the guard flag and be used for the entry position.

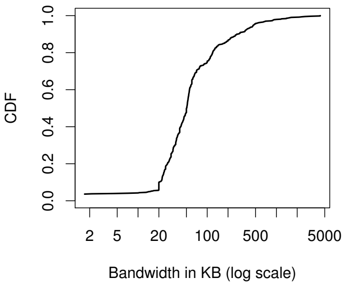


Fig. 4. Bandwidth contributions from middle-only routers

6 Related Work

Security in low latency anonymous networks. An early security analysis of low-latency anonymous networks suggests that an anonymous path is compromised if its endpoints are controlled by an adversary; the expected success of such an attack is roughly $(c/n)^2$, where there are c malicious routers, n total routers, and clients choose routers uniformly at random [31]. As networks such as Tor evolved, it became necessary to balance the traffic load over a diverse set of volunteer routers, making the task of analytically modeling path compromise more challenging. To meet this challenge, Murdoch and Watson propose that path compromise be analyzed empirically through faithful simulation of the underlying routing mechanism in the presence of different threat models [19]. We adopt a similar empirical approach to reasoning about Tor's security properties.

Selective disruption attacks. Selective disruption attacks are a form of denial-of-service (DoS) that allow an adversary to increase the number of circuits compromised. These attacks work as follows: a malicious router who uses selective disruption should refuse to forward traffic in the event that they participate in a circuit that is not compromised. This causes the circuit to fail and be rebuilt, providing an opportunity for malicious routers to compromise another circuit.

Bauer *et al.* show that an attacker with only six malicious Tor routers who utilizes the selective disruption strategy can compromise over 46% of all clients' circuits in an experimental Tor network with 66 total routers [4]. Similarly, Borisov *et al.* demonstrate that an adversary who uses selective disruption experiences a significantly greater path compromise rate. Without selective disruption, an attacker who controls 50% of the network's bandwidth can compromise 25% of all circuits, but with selective disruption they can compromise up to 66% of circuits [9]. However, their analysis was performed using a highly simplified model of Tor path selection. They also found that, for mix networks, increased path length results in greater susceptibility to selective disruption attacks, but did not analyze the effects of path length in Tor. We examine how selective disruption attacks are less effective with two-hop paths than three hops.

Given the danger of selective disruption, Danner *et al.* describe an algorithm for detecting selective disruption attacks that requires a number of probes that scales linearly with the network size [10]. However, such an active probing approach may introduce high load into the network. Also, active probing could be gamed by an intelligent attacker who can recognize the probes, or the adversary could disrupt circuits probabilistically to blend in with expected background circuit failures. Ultimately, the DoS strategy allows attackers to perform traffic analysis on a far greater number of circuits than would otherwise be possible.

End-to-end traffic correlation attacks. Prior work has shown that end-to-end traffic correlation attacks are highly effective against low-latency anonymizing networks. Levine *et al.* demonstrate through simulation that the performance of timing-based traffic correlation attacks is dependent on network conditions, but they show that an adversary can correlate traffic with perfect accuracy when the packet drop rate is very low [2]. For circuit linking experiments carried out on a small experimental Tor network, Bauer *et al.* report only 12 false positives out of over ten thousand successful correlations [4]. However, their traffic load was light and uniform, which may have contributed to the extremely low false positive rate. Also, Syverson and Øverlier report a negligible false positive rate for a traffic correlation attack on a Tor hidden service [14]. In this paper, we verify that similarly high traffic correlation accuracies can be expected for low-resource traffic analysis attacks launched on the real Tor network.

Alternate router selection strategies. Given the threat of malicious routers positioning themselves at circuit endpoints for a large number of circuits, Snader and Borisov propose that clients have the ability to "tune" the router selection process between security and performance [32]. Choosing routers more uniformly at random reduces the end-user's risk of choosing malicious routers who inflate their bandwidth claims to attract traffic, however, at the potential cost of choosing low bandwidth routers and experiencing poor performance. In addition, Sherr *et al.* propose that link-based attributes (such as latency or jitter) be used to select routers rather than node-based attributes (like bandwidth) [33]. However, these proposed routing techniques have yet to be adopted in practice. Consequently, we only consider Tor's current router selection algorithm in our subsequent analysis.

Prior performance analyses. Beyond the security properties of low latency anonymizing networks, recent work has investigated their performance characteristics. It has been shown that users are more likely to use anonymous communication services that offer better performance [34]. In a performance study of Tor and AN.ON [35] (a mix cascade) from the end-user's perspective, Wendolsky *et al.* find that Tor is subject to unpredictable performance and observe that users exhibit a four second tolerance to delay [28]. However, Tor users often experience significant delays beyond this user tolerance threshold [29].

To help explain Tor's poor observed performance, Reardon and Goldberg identify that because Tor multiplexes many streams over the same TCP connections, congestion control interference among different circuits is produced [36]. These unintended interactions often cause very high delays for end-users. TCP-over-DTLS, an alternate transport design, is proposed to improve performance. Our work is complementary to these prior studies. Since end-users are sensitive to excessive delays, we quantify the performance improvement that can be expected with a two-hop design and argue that such a design may offer even more improvement in combination with TCP-over-DTLS.

7 Conclusion

We critically evaluate Tor's path length and consider the advantages and disadvantages of a two-hop and three-hop design. We show that two-hop paths are slightly less vulnerable to circuit compromise attacks than three-hop paths, but two-hop paths are trivially vulnerable to adaptive surveillance and introduce potential liabilities for exit node operators. While performance is improved with shorter paths, we conclude that there is no strong argument for reducing Tor's path length. However, we identify a number of open issues that could effect this decision. Our hope is that this paper encourages further investigation into the security and performance trades-offs of various path lengths.

Acknowledgments

We thank Prateek Mittal, Mike Perry, and the anonymous reviewers for their helpful comments and suggestions.

References

1. Serjantov, A., Sewell, P.: Passive attack analysis for connection-based anonymity systems. In: Proceedings of ESORICS 2003. (October 2003)
2. Levine, B.N., Reiter, M.K., Wang, C., Wright, M.K.: Timing attacks in low-latency mix-based systems. In: Proceedings of Financial Cryptography (FC '04). (February 2004)
3. Shmatikov, V., Wang, M.H.: Timing analysis in low-latency mix networks: Attacks and defenses. In: Proceedings of ESORICS 2006. (September 2006)

4. Bauer, K., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.: Low-resource routing attacks against Tor. In: Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007), Washington, DC, USA (October 2007)
5. Houmansadr, A., Kiyavash, N., Borisov, N.: Rainbow: A robust and invisible non-blind watermark for network flows. In: NDSS, The Internet Society (2009)
6. Ling, Z., Luo, J., Yu, W., Fu, X., Xuan, D., Jia, W.: A new cell counter based attack against Tor. In: CCS '09: Proceedings of the 16th ACM conference on Computer and communications security, ACM (2009) 578–589
7. Murdoch, S.J., Zielinski, P.: Sampled traffic analysis by Internet-exchange-level adversaries. In Borisov, N., Golle, P., eds.: Privacy Enhancing Technologies. Volume 4776 of Lecture Notes in Computer Science., Springer (2007) 167–183
8. Back, A., Möller, U., Stiglic, A.: Traffic analysis attacks and trade-offs in anonymity providing systems. In Moskowitz, I.S., ed.: Proceedings of Information Hiding Workshop (IH 2001), Springer-Verlag, LNCS 2137 (April 2001) 245–257
9. Borisov, N., Danezis, G., Mittal, P., Tabriz, P.: Denial of service or denial of security? How attacks on reliability can compromise anonymity. In: Proceedings of CCS 2007. (October 2007)
10. Danner, N., Krizanc, D., Liberatore, M.: Detecting denial of service attacks in Tor. In: Financial Cryptography and Data Security, Berlin, Heidelberg, Springer-Verlag (2009) 273–284
11. Tran, A., Hopper, N., Kim, Y.: Hashing it out in public: Common failure modes of DHT-based anonymity schemes. In: ACM Workshop on Privacy in Electronic Society. (2009)
12. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (August 2004)
13. Dingledine, R., Mathewson, N.: Tor protocol specification. `https://www.torproject.org/svn/trunk/doc/spec/tor-spec.txt`
14. Øverlier, L., Syverson, P.: Locating hidden servers. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy, IEEE CS (May 2006)
15. Snader, R., Borisov, N.: EigenSpeed: Secure peer-to-peer bandwidth evaluation. In: Proceedings of the 8th International Workshop on Peer-to-Peer Systems (IPTPS). (2009)
16. Perry, M.: TorFlow: Tor network analysis. http://fscked.org/talks/TorFlow-HotPETS-final.pdf
17. Dingledine, R., Mathewson, N.: Tor path specification. `https://git.torproject.org/checkout/tor/master/doc/spec/path-spec.txt`
18. Dingledine, R., Mathewson, N.: Tor directory protocol, version 3. `https://www.torproject.org/svn/trunk/doc/spec/dir-spec.txt`
19. Murdoch, S.J., Watson, R.N.M.: Metrics for security and performance in low-latency anonymity systems. In: Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008), Leuven, Belgium (July 2008)
20. McCoy, D., Bauer, K., Grunwald, D., Kohno, T., Sicker, D.: Shining light in dark places: Understanding the Tor network. In: Proceedings of the 8th Privacy Enhancing Technologies Symposium. (July 2008)
21. Loesing, K., Murdoch, S., Dingledine, R.: A case study on measuring statistical data in the Tor anonymity network. In: Workshop on Ethics in Computer Security Research. (January 2010)
22. Edman, M., Syverson, P.F.: AS-awareness in Tor path selection. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security (CCS). (2009) 380–389

23. Tor metrics portal: Data. http://metrics.torproject.org/data.html#stats
24. Shannon, C.: A Mathematical Theory of Communication. In: Bell System Technical Journal. Volume 27. (1948) 379–656
25. Loesing, K.: Measuring the Tor network: Evaluation of client requests to the directories. Tor Project Technical Report (June 2009)
26. Wright, M.K., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: An analysis of a threat to anonymous communications systems. ACM Trans. Inf. Syst. Secur. **7**(4) (2004) 489–522
27. Abbott, T., Lai, K., Lieberman, M., Price, E.: Browser-based attacks on Tor. In: Privacy Enhancing Technologies Symposium. Volume 4776 of Lecture Notes in Computer Science., Springer (2007) 184
28. Wendolsky, R., Herrmann, D., Federrath, H.: Performance comparison of low-latency anonymisation services from a user perspective. In Borisov, N., Golle, P., eds.: Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007), Ottawa, Canada, Springer (June 2007)
29. McCoy, D., Bauer, K., Grunwald, D., Tabriz, P., Sicker, D.: Shining light in dark places: A study of anonymous network usage. University of Colorado Technical Report CU-CS-1032-07 (August 2007)
30. Dingledine, R., Mathewson, N.: Anonymity loves company: Usability and the network effect. In: Workshop on the Economics of Information Security. (June 2006)
31. Syverson, P., Tsudik, G., Reed, M., Landwehr, C.: Towards an analysis of onion routing security. In Federrath, H., ed.: Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, LNCS 2009 (July 2000) 96–114
32. Snader, R., Borisov, N.: A tune-up for Tor: Improving security and performance in the Tor network. In: Proceedings of the Network and Distributed Security Symposium (NDSS), IEEE (February 2008)
33. Sherr, M., Blaze, M., Loo, B.T.: Scalable link-based relay selection for anonymous routing. In: 9th Privacy Enhancing Technologies Symposium (PETS '09). (August 2009)
34. Köpsell, S.: Low latency anonymous communication - How long are users willing to wait? In: ETRICS. (2006) 221–237
35. JAP. http://anon.inf.tu-dresden.de
36. Reardon, J., Goldberg, I.: Improving Tor using a TCP-over-DTLS tunnel. In: Proceedings of the 18th USENIX Security Symposium. (August 2009)

Examining Privacy and Surveillance in Urban Areas: A Transportation Context

Caitlin D. Cottrill
Urban Transportation Center, UIC
412 S. Peoria St., Suite 340, CUPPA Hall
Chicago, IL 60607
Phone: +1-312-996-4820
Fax: 312-413-0006
ccottr2@uic.edu

Abstract:
In the post-9/11 world, trade-offs between safety, security and privacy have received an increasing amount of attention and discussion. Particularly within the realm of transportation, it is evident that no clear bright line exists as to the degree to which travelers are willing to exchange their "privacy in public" for an increased measure of safety and security. While certain invasions of privacy are generally accepted as being "critical" to ensuring safety and security (such as scanning of personal effects when flying or GPS-equipped cell phones) and are therefore submitted to with some degree of aacceptance, other forms of surveillance (such as red light cameras) are regarded as being invasive without providing concomitant benefits, and are thus argued against in public meetings and, eventually, the courts. It is interesting to ask, then, what forms of transportation surveillance currently being implemented or proposed are most likely to be accepted or rejected within the context of the urban environment, and to what extent will questions of equity and fairness impact these responses? It is critical, at this juncture in the development of advanced forms of intelligent transportation systems, to step back and evaluate the relevant impacts of surveillance, not only on the function and security of the transportation system, but also on the travelers within that system.

I. Introduction

Many definitions of privacy have been proposed, but most tend to have issues of control of information and its flow as their foundations. Alan Westin has defined privacy as, “the claim of an individual to determine what information about himself or herself should be known to others.”[i] This very broad definition contains within itself a wealth of further claims related to different states of privacy, and to the context of the person and his or her information. By approaching the privacy claim from the viewpoint of context, the emerging literature on the social, political, and economic variations inherent in the experience of privacy reveal a range of expectations dependent upon the person’s individual understanding. The concept of privacy as based upon a subjective or contextually-based understanding is also consistent with the legal understanding of the subject – for example, the Fourth Amendment, central to legal justifications for privacy protection, has been understood by the courts to be centered on “reasonable expectations of privacy”.[ii] Security, on the other hand, often results in a loss of control due, in part, to an exterior determination of context. One of the most commonly used definitions of security is “A condition that results from the establishment and maintenance of protective measures that ensure a state of inviolability from hostile acts or influences.”[iii] While security may be established at the personal level, such as with home security measures, in the context of transportation it is more often used to refer to the travelers on the network, whether surface or other. Because the context of privacy is often set at the personal level, while the security context is most often determined at a community or societal level, the potential for conflict between the two is great.

This paper will focus primarily on the conflict between privacy and security from the framework of surface transportation, particularly in regard to Intelligent Transportation Systems (ITS). ITS technologies have begun to change the transportation landscape. Whereas traditional transportation initiatives focused primarily on constructing the physical landscape (such as roads, bridges and traffic signals) to enhance mobility and safety, ITS technologies are primarily concerned with managing the flow of information to travelers, transportation providers, and vehicles themselves to enable efficient transportation choices that in turn increase mobility and safety. ITS is a critical component in enhancing the flow of information, but its use comes at a cost. According to Joe Palen of Caltrans New Technology and Research Program,

> An Intelligent Transportation System (ITS), by definition, involves the use of intelligence to enhance the operation of the transportation system. Intelligence, by definition, requires information. Information, by definition, is data formulated in a formation. Data is generated by surveillance. Therefore, surveillance forms the basis

> for the formation of information for an ITS. You can't have a usable ITS without surveillance. (Palen, 1997)

Surveillance within the realm of surface transportation has many components, such as red light cameras used for the purposes of ticketing offenders and encouraging law-abiding behavior; GPS units used as vehicle probes to track the flow of traffic; and Electronic Toll Collection (ETC) systems used both to improve traffic flow through toll collection points and to estimate traffic flows along toll corridors. At first glance, the privacy and security ramifications of such technologies may not appear great, but both the information gathered and the possibility for linking this information with other sources opens up the potential for a wide range of security applications and privacy violations. This paper will first examine these potential uses and applications in the context of security, followed by some of the conflicts that may emerge with privacy preferences. Two examples of technological applications will then be presented to identify how they may have impacts within the urban environment. Finally, needs for further research will be identified.

II. Understanding security

For purposes of this paper, it is first necessary to clarify how security may be defined. Security may be approached from the two primary directions of deterrence and response. Under this approach, deterrence may be evaluated in terms of lessening the chance of a malicious attack, while response may be understood as the ability to use collected data for purposes of law enforcement, such as strengthening a case against a suspect with recorded footage of his or her involvement in an incident. Tinnefield describes the difference between the two by stating, "Preventive [or deterrence-oriented] surveillance is different from its traditional use for investigating a specific criminal offense, which requires proof that a crime has taken place or is likely to occur. Preventive surveillance lacks any such connection to a specific criminal act."[iv] While ITS applications may be used both for deterrence/preventive and response purposes, the timing and approach for each purpose is significantly different. For deterrence, the presence of ITS surveillance technology should be acknowledged for maximum impact at the point of potential conflict. For secondary, post-incident uses, however, covert methods of surveillance may be more useful, as avoidance of the surveillance device is likely if the location is known. Because many agencies intend that surveillance technologies will serve both purposes, it is possible that a combination of both overt and covert ITS surveillance applications may prove most effective.

III. Security in the transportation context

Following the events of September 11th, 2001, the U.S. Congress passed into law the Aviation and Transportation Security Act, which created the Transportation Security Administration (TSA). While much of the focus of the TSA has rested on aviation applications, the overall mission of the department is to, "protect…the Nation's transportation systems to ensure freedom of movement for people and commerce."[v] Under this guiding mission, part of the TSA's focus has been placed on security applications of ITS. According to *The National Intelligent Transportation Systems Program Plan: A Ten-Year Vision*, one of the overarching goals of ITS is, "…a transportation system that is well-protected against attacks and responds effectively to natural and manmade threats and disasters, enabling the continued movement of people and goods even in times of crisis."[vi] According to a 2002 report published by the NYU Wagner Rudin Center for Transportation Policy and Management, the Federal Highway Administration (FHWA) has recommended the following strategies to meet this goal:

1. Develop emergency plans, tools, and resources;
2. Perform vulnerability assessments;
3. Compile case studies on attacks;
4. Conduct freight technology security demonstrations;
5. Solicit ITS technology projects intended to improve security; and
6. Host workshops to discuss these and other related issues.[vii]

The fifth strategy mentioned above is perhaps most related to issues surrounding the conflicts between security and privacy. In accordance with this strategy, a number of technologies have been or are being implemented, including the following:

- Smart Cards
- Biometrics
- Automatic Vehicle Identification
- Map Databases
- Vehicle Classification Sensors
- Weigh-in-Motion Technology
- Spatial Geo-Location[viii]

In the realm of security, each of these technologies is dependent upon the ability to identify and track travelers and freight along the nation's transportation system. The guiding principle is that data collected and linked from the implementation of these technologies may be used, not only for managing the flow of travelers, but also to prevent or deter terrorist attacks. For example, Peyrebrune and Cerreño state that, "…technologies exist that enable security personnel to detect the contents of vehicles, including hazardous substances, explosives, and drugs, without opening the vehicles firsthand. Also available are technologies that match a specific vehicle with a specific operator and specific cargo, preventing travel in the absence of a match."[ix] Such ITS technologies, in addition to others listed above, may have very beneficial ramifications for the nation's security.

Particularly within the urban environment, the melding of the strategies outlined above become of great interest. Because of the diversity of modes available in areas of concentrated population (including mass transit, air transportation, personal vehicles, cycling and walking), along with static cameras used for other purposes, the potential to align and link different methods of ITS surveillance along a route increase. As in the example of the September 11th hijackers, where a network of images from ATMs, gasoline purchases, and airport security were combined to produce an activity path, it is possible to recreate detailed patterns of behavior from the network of surveillance methods becoming ever more pervasive in the urban environment.[x] In such a context, where law-breaking behavior has occurred, the response characteristics of ITS may be viewed as very beneficial by the general public. In question, however, is how the desire for these benefits compares to their potential privacy invasions.

IV. Understanding privacy

With the security benefits inherent in the applications noted above also comes the potential for violations of privacy. Peyrebrune and Cerreño, for example, note that, "…the issue of collecting information about people to prevent terrorist activities versus public privacy will be a public policy issue over the long term."[xi] Simson Garfinkel states the issue more clearly when he notes that, "If ITS systems are developed and deployed which do not respect the privacy of the American driver, there is a good chance that Americans will demand that the system be shut off. Without strong privacy provisions, ITS will not succeed."[xii] Some privacy concerns being voiced by opponents include the following:

- Because pervasive computing systems generally used in ITS may be embedded or invisible, users may not be aware that they are present and collecting data.[xiii]
- The ability to collect and connect data on users in their day-to-day activities may provide a more robust data set on actual travel patterns, origins and destinations – information that has not previously been readily available.
- Defining the secondary uses of collected data will greatly impact the user's level of comfort with pervasive ITS technology. Privacy considerations must guide the degree to which collected information may be shared and used.
- Users of ITS applications may not be aware of their potential privacy implications, making it difficult for them to accurately assess their desire for the potential benefits against potential privacy loss.

Though the issues named above do not cover the universe of concerns that have been raised by privacy advocates, they do indicate the diversity of issues that must be addressed by ITS technologies in the transportation context, and highlight some of the conflicts that are present in terms of security.

Given the difficulties identified above, it is necessary at this juncture to define more clearly the elements involved in expectations of privacy. The Federal Trade Commission (FTC) identifies five "core principles" relevant to privacy policy, namely: "(1) Notice/Awareness; (2) Choice/Consent; (3) Access/Participation; (4) Integrity/Security; and (5) Enforcement/ Redress."[xiv] In short, these principles form the government's definition of privacy as it may be reasonably expected by the consumer. The first principle, notice/awareness, may be considered as the most fundamental in regards to the privacy/security debate, as it sets the context for the remaining four. According to the FTC, "Consumers should be given notice of an entity's information practices before any personal information is collected from them. Without notice, a consumer cannot make an informed decision as to whether and to what extent to disclose personal information."[xv] The second principle, choice/consent, rests upon the belief that consumers should have options regarding if information will be collected and how that information will be used. This principle is particularly relevant in relation to secondary usage of data, in which information collected for one purpose may be used for a different and potentially unrelated purpose. The access/participation principle "refers to an individual's ability both to access data about him or herself…and to contest that data's accuracy and completeness."[xvi] Integrity/security is concerned with the responsibility of collectors to ensure the integrity of collected data via such means as cross-referencing against reputable data sets, as well as with ensuring that collected data is protected from loss and unauthorized access via both managerial and technical means. Finally, enforcement/redress is intended to ensure the efficacy of the preceding principles by providing a mechanism for enforcement. From a definitional standpoint, these principles expand the notion of "control" outlined above by introducing the associated concepts of notice, choice, access and security.

V. Privacy in the transportation context

In terms of the privacy/security debate, it is critical to note that in order to protect privacy in the context of personal expectations as required by the FTC, travelers must be aware of the potential that their personal information may be collected, and have some degree of control over what information they wish to share. For purposes of security from a deterrence standpoint, notice and awareness are also critical. For the same reasons that retail establishments often display a sign warning, "These premises are under CCTV surveillance", transportation departments and agencies may erect signs notifying travelers that their moves are being recorded. In this type of situation, the agency believes that by notifying the public that their actions are being monitored, they will be less inclined to participate in law-breaking behavior.

As noted above, however, notice and awareness may also have the effect of simply moving deviant behavior to a different location, as persons may avoid areas with overt surveillance. It is here that covert surveillance enters the picture, but this may be at odds with the principles of notice/ awareness and choice/consent.

If, as outlined in the introduction, the claim for privacy may be traced back in some part to issues of control, it is necessary that persons under ITS surveillance be aware of both the surveillance itself and of its potential ramifications in order that they may exercise control over what information is shared, to whom it is accessible, and for what purposes it is used. One issue of overarching concern with this requirement is that of equity and justice, as understanding the potential ramifications of ITS surveillance may highlight a knowledge gap between persons familiar with ITS technologies and those who are not. Without a concentrated effort to educate all travelers as to the actual benefits and costs evident in ITS surveillance, it is difficult to argue that travelers are able to make an accurate assessment of need. Two additional difficulties that emerge in the context of covert surveillance are, first, if covert surveillance is to be an effective means of security response, it must be allowed to gather any pertinent information outside of the control of the subject; and, second, information gathered under covert surveillance must be subject only to the control of the collectors for the purposes they deem necessary within the legal context. While these needs are at odds with the privacy principles outlined above, they are allowable within the legal framework that defines security measures. As was seen in the public outcry related to the National Security Agency's (NSA) warrantless wiretapping surveillance program, however, covert surveillance that applies to a broad spectrum of the population (such as would be necessary in a public travel environment) may be unpalatable to the general public without a full understanding of the benefits and costs of such a program and if viewed as taking place beyond the scope of applicable laws.

Within the context of ITS the uses of security and mobility management are mixed. Given this, the question arises as to which needs take precedence. If the overarching desire is for mobility improvements, then adoption of associated technologies such as GPS probes and ETC tags will depend in large part upon the acceptance of the traveling public, which in turn will depend upon identifying and addressing their privacy concerns. If, on the other hand, the desire for security is prevalent, it may be necessary to limit the amount of information provided to travelers in order to enhance the amount of information gathered. The following section will attempt to outline the potential ramifications of each approach in the examples of ETC and GPS.

VI. Privacy and security implications of Electronic Toll Collection (ETC)

Electronic Toll Collection (ETC) systems are gaining widespread adoption in the United States. Advocates of the systems tout their benefits in terms of convenience, improvement of traffic flow, decreases in fuel consumption and improvement in air quality.[xvii] According to Briggs and Walton, the systems generally work as follows: "ETC allows participating vehicles equipped with electronic transponders, or tags, to avoid stopping to pay tolls. Instead, the electronic transponder communicates via radio frequency or microwave to a roadside computer. The tagged vehicle is identified as an electronic toll collection system user, and the toll amount is debited from the user's account."[xviii] ETC systems require a fair amount of data from the user. According to the Briggs and Walton report, different ETC providers require a varying amount of information from travelers, including contact information (name, address, telephone number(s)), vehicle information (license plate number and state and make, model and year of the enrolled vehicle), and financial information (generally a credit card or bank account number).[xix] Additional information requested by some providers includes driver's license number, social security number, or mother's maiden name.[xx] While some systems debit transactions directly from the user's bank account, others use a pre-paid account that must be kept loaded with a minimum amount of funds. In the latter configuration, users are either notified directly when their balance falls or is about to fall below the minimum requirement, or have funds directly debited from a linked bank account.

A number of privacy concerns have been raised about ETC systems, particularly insofar as electronic records provide for the linkage of personal, vehicle, travel and financial records. In order to provide a record of transactions in case of charge dispute or other concern, ETC agencies typically keep logs of transactions, including locational information, on each customer. Patrick Riley notes one common concern in his case study of the FasTrak ETC system in the California Bay Area, stating that, "personal information can be used when required by law or ordered so by a court of 'competent jurisdiction.' This apparently includes civil court, as many divorce attorneys seek access to and admit as evidence FasTrak logs."[xxi] The potential for transaction logs and customer files to be accessed either for legal purposes or by malicious agents may create a disincentive for use, despite potential cost and time savings for the user. Riley notes that in a survey of Bay Area residents, the third and fourth most commonly noted reasons (out of 13 possible responses) cited by respondents for not taking part in the FasTrak program are "I have privacy concerns" and "I'm worried about how my private information may be used."[xxii] Such concerns have the potential to impact the deployment and benefits of ETC systems.

ETC systems are installed on designated toll roads only, which limits the applicability of their data collection at lower spatial resolutions. Additionally, transactions are only recorded at designated toll plazas, which may limit the degree of temporal resolution that may be attained from these records. One obvious benefit for purposes of data manipulation, however, is that transaction records are associated with a specifically identified transponder ID, and thus may be automatically linked to and stored with the user's profile. Such a system makes performing analysis on the database relatively easy. Additionally, the limited number of transaction records may make the computing overhead relatively low. Mills and Huber, in their 2002 article “How Technology Will Defeat Terrorism”, relate the security potential for ETC systems when they state, “Even systems as simple as these can be linked up to security networks, too, and can do much to enhance safety, because so much of security comes down to establishing identity and tracking patterns of conduct – just the sorts of things that the automatic toll collectors already do.”[xxiii] Given the evident concern with privacy implications of ETC, however, it is unclear as to whether the security uses of ETC will be efficient in an opt-in system as currently used. Mills and Huber state that, “The first step is to divide the civilian world into two, separating the trustworthy cooperators from the non-cooperators, so we don’t have to search *every* car, package, and pocket.”[xxiv] This approach, however, requires the clustering of all persons with high privacy concerns – thus conflating persons with legitimate privacy concerns and those who are law-breakers. While the potential for security uses of ETC-type systems is evident, balancing these against privacy concerns will require further examination of overt uses, such as in cases where the user must choose to participate, and covert uses, where ETC sensing equipment could be programmed to record identification information from Radio Frequency Identification (RFID) tags implanted in vehicles for tracking and other purposes.

VII. Privacy and security implications of Global Positioning Systems (GPS)

Global Positioning Systems (GPS), unlike ETC systems, tend to have the ability to be more universal in scope. The use of GPS in vehicle navigation and tracking systems has been growing in recent years, particularly as the selective availability feature, which introduced signal errors into non-military applications, has been disabled. Active vehicle-based GPS tracking architectures typically consist of an on-board unit, a base station and a communication link.[xxv] The on-board unit collects and periodically provides a message containing identification and travel environment data over the communications link to the base station, where the data are recorded and stored. Within the realm of tracking, GPS are increasingly being used for vehicle probe applications. According to Hoh and Gruteser, “Probe vehicles carry GPS receivers and communication infrastructure such as cellular links to periodically report records with the following parameters to traffic information systems: latitude, longitude, time, speed. From this information the system can estimate current mean vehicle speed, which can be fed into navigation systems or can be used to build a real-time congestion map.”[xxvi] Such probe applications are likely to grow in scale and deployment, and will be used here to examine potential benefits, drawbacks, and privacy concerns for GPS tracking applications. It should be noted that GPS-based location trackers do not have to be vehicle-based; in fact, one great benefit of GPS systems is that they may be included in devices that may be carried by a pedestrian or cyclist. This analysis, however, will focus on vehicle-based systems.

One of the primary benefits of GPS probe applications is that they are not bound to specific segments of the roadway. Unlike ETC systems, which tend to be concentrated on fairly large or heavily traveled roads (due in part to the overhead costs of implementing the systems, as well as to legal designations of toll roads), GPS probe systems are able to collect data at any given location as long as a connection may be established to the GPS satellite network. This aspect of GPS probe applications is particularly useful to transportation network administrators, as there has long been a paucity of data available on those roads with low functional classifications (particularly local roads) due to infrastructure costs and maintenance. Additionally, unlike ETC systems, which may only collect data at when the user's transponder interacts with the roadside computer, GPS tracking devices have a nearly unlimited ability to transmit information continually once the system has been installed. Additional benefits of a GPS-based probe system identified by FHWA are: relatively low operating cost after initial installation, automated data collection, and the increasing availability of GPS as a consumer product.[xxvii] Potential disadvantages noted are privacy issues, potential signal loss in urban areas, consistency between drivers, and a relatively high installation cost.[xxviii]

User benefits of GPS-based probe systems may be considered in two ways. Improved data and analysis abilities on the part of transportation planners and others may lead to improvements in the transportation network, saving the traveler time, fuel, and other resources. Additionally, real-time knowledge of events may improve the efficiency with which transportation network administrators are able to deploy information and personnel. A second way that these applications may be useful to the user is if they are linked with a vehicle navigation system, which could provide the user with efficient travel routes and updated information related to traffic incidents. For security purposes, GPS are particularly useful, as they may be small, difficult to detect, and transmit data that may be mined to look for patterns indicating the potential for deviant behavior.

Many of the benefits in terms of data collection, retention and analysis seen in ETC systems are also true of GPS probe applications. Assuming that sent messages include a unique vehicle identifier, the automated nature of the data collection system will make it easy for records to be linked to an individual vehicle and its travel path. Such information will be particularly useful for determining real-time traffic patterns, analyzing traffic patterns over time, and tracking individual travelers. Depending on the level of temporal resolution at which data are collected, great benefits are also possible for establishing a library of data and patterns on roadways of lower functional classification, which may allow better tracking of actual origin-destination routes. The level of temporal resolution, however, may have great impacts on the amount of computing overhead required for the use of these systems. If great amounts of data are collected from a great number of vehicles, the computing costs of storage and analysis may overwhelm the agency's system. Additionally, even if traveler data is collected anonymously, thus meeting in practice privacy requirements of the privacy concerned, it may still be able to use GPS to identify individuals. Hoh, et al. conducted a study of vehicle probes in which anonymous GPS traces of 239 vehicles in the Detroit, Michigan region were subjected to a clustering analysis to see if it would be possible to determine the likely home location.[xxix] Based on a sampling frequency rate of one record per driver per minute, the authors found that it was possible to identify a likely home location for approximately 85% of the vehicles.[xxx] Such a finding indicates that for purposes of privacy preservation, it will be necessary to lower the temporal resolution at which data are collected, while security advocates may desire the temporal resolution to remain high. For users to accept the full degree of GPS applications within security and transportation realms, it will be necessary to fully disclose both the potential for misuse and the protective methods by which such misuses may be halted.

Table 1 presents a general overview of the data characteristics and potential security and privacy impacts of the two ITS systems outlined above. As the above review has shown, each system has a variety of benefits and considerations for both users and agencies in terms of overhead and infrastructure costs, the level of data able to be collected, and the potential privacy and security impacts of their use. Additional considerations regarding use of archived data (including secondary use for marketers and law enforcement, among others) should also be considered, though not covered here in detail. The ability of both ETC and GPS systems to create electronic records of travel make them especially useful for security purposes, as this increases the efficiency with which collected data may be utilized for identification of potential security threats. However, this is also the characteristic that is perhaps of greatest concern from a privacy perspective, as it may allow a fairly detailed travel history to be constructed and used for purposes that may be resisted by system users. Generally, decisions regarding the resolution and speed of data collection must represent a balance between the data desired by transportation security advocates and the degree of privacy desired by the traveler. While there is great potential for transportation networks with the advent of ITS technologies such as ETC and GPS, their costs and potential risks, particularly from the point of view of the potential user should be taken into consideration when planning for system implementation.

System Type	Data Characteristics					Potential Security Applications	Potential Privacy Concerns
	Spatial Resolution	Temporal Resolution	Speed				
			Collection	Cataloging	Analysis		
Electronic Toll Collection Systems	Relatively limited; based on location of static collection points.	Relatively low due to limited number of collection points.	High	High to medium depending on number of records collected and computing capabilities.	High to medium depending on number of records collected and computing capabilities.	Analysis of travel patterns linked to individual financial records, allowing for potential identification of security threats; Potential to expand uses of ETC infrastructure to collect additional data on vehicles and travelers.	Linking of identifiable individual information, travel patterns, and financial records may subject records to malicious uses; Records may be subpoenad for use in legal proceedings.
Global Positioning Systems	High within the space covered by collection infrastructure.	Potential for a very high temporal resolution within the network; may need to be lowered due to privacy and computing overheads.	High	High to low depending on computing capabilities and number of records collected.	High to low depending on level of resolution.	Potential to collect and analyze data on individual travel patterns at high resolution with little knowledge by the person being surveilled.	High degree of information gathered may open concerns for malicious uses; Travelers may be unaware of amount of data being collected and secondary uses.

Table 1: Overview of ITS Data Collection Characteristics

VIII. Conclusion

The overview above has only begun to touch upon the issues prevalent in the security/privacy debate in relation to ITS. Because of the concentration of travelers and security threats in urban areas, the discussion is particularly relevant to urban dwellers and travelers. Armstrong and Ruggles, for example, note that, “Cameras are not (yet) everywhere, but camera proliferation has been accepted by many urban residents as a fact of everyday life.”[xxxi] While the presence of these static cameras may be accepted by some, further uses such as those outlined above may create more of an incentive to resist the installation and adoption of cameras that may be used to identify and track along a route in urban areas. Additional studies have also indicated that the socio-demographic characteristics of those who are willing to adopt such ITS technologies and those who resist them may differ, thus bringing additional questions of equity and fairness to light. As the transportation realm struggles to balance the need to ensure the security and protect the privacy of travelers along its network, it will be necessary to further address these issues, and examine in more detail the relevant benefits and concerns.

[i] Westin, A. “Social and Political Dimensions of Privacy.” *Journal of Social Issues,* Vol. 59, No. 2, 2003. Pp. 431-453.

[ii] Slobogin, C. “Public Privacy: Camera Surveillance of Public Places and the Right to Anonymity.” Mississippi Law Journal, 72: 213-299, 2002.

[iii] http://www.tsl.state.tx.us/ld/pubs/compsecurity/glossary.html.

[iv] Tinnefeld, M.T. “The Security Principle – A Challenge to the Right of Privacy and of Freedom of Information. Presented at the First Annual European Freedom Summit. Berlin, July 2, 2007.

[v] TSA Mission statement. http://www.tsa.gov/who_we_are/mission.shtm.

[vi] ITSA. “Delivering the Future of Transportation – The National Intelligent Transportation Systems Program Plan: A Ten-Year Visions.” Executive summary. January 2002.

[vii] Peyrebrune, P.E. and A.L.C. de Cerreño. “Security Applications of Intelligent Transportation Systems: Reflections on September 11 and Implications for New York State.” July 2002.

[viii] Ibid.

[ix] Ibid.

[x] Armstrong, M.P. and A.J. Ruggles. "Geographic Information Technologies and Personal Privacy." Cartographica, Vol. 40, Iss. 4. Winter 2005. Pp. 63-73.
[xi] Peyrebrune and Cerreño.
[xii] Garfinkel, S. "Why driver privacy must be a part of ITS." Converging Infrastructures: Intelligent Transportation and the National Information Infrastructure. MIT Press. 1996.
[xiii] Bhaskar, P. and S.I. Ahamed. "Privacy in Pervasive Computing and Open Issues." Second International Conference on Availability, Reliability and Security. 2007.
[xiv] Federal Trade Commission. "Fair Information Practice Principles." 25 June 2007. http://www.ftc.gov/reports/privacy3/fairinfo.shtm.
[xv] Ibid.
[xvi] Ibid.
[xvii] Holdener, D.J. "Electronic Toll Collection Information: Is Personal Privacy Protected?" Compendium: Graduate Student Papers on Advanced Surface Transportation Systems, Southwest Region, University Transportation Center Research Report # SWUTC/96/72840-00003-1, Texas Transportation Institute, Texas A&M University System. August 1996. Pp. D-3 – D-4.
[xviii] Briggs, V. and M. Walton. The Implications of Privacy Issues for Intelligent Transportation Systems (ITS) Data. Southwest Regional University Transportation Center. May 2000.
[xix] Ibid.
[xx] Ibid.
[xxi] Riley, P.F. "The Tolls of Privacy: An Underestimated Roadblock for Electronic Toll Collection Usage." Computer Law and Security Report, 24. 2008. Pp. 521-528.
[xxii] Ibid.
[xxiii] Mills, M.P. and P.W. Huber. "How Technology Will Defeat Terrorism: At home and abroad, digital wizardry will keep us safe." City Journal, Winter 2002.
[xxiv] Ibid.
[xxv] Padmanabhan, J. "GPS Based Vehicle Tracking System." http://www.gisdevelopment.net/technology/gps/techgp0044.htm.
[xxvi] Hoh, B. and M. Gruteser. "Enhancing Security and Privacy in GPS-Based Traffic Monitoring Systems." WINLAB Project Summary. Fall 2006.
[xxvii] Turner, S.M., W.L. Eisele, R.J. Benz and D.J. Holdener. Travel Time Data Collection Handbook. Report No. FHWA-PL-98-035. March 1998.
[xxviii] Ibid.
[xxix] Hoh, B., M. Gruteser, H. Xiong, and A. Alrabady. "Enhancing Security and Privacy in Traffic-Monitoring Systems." Pervasive Computing. IEEE, October – December 2006. Pp. 38-46.
[xxx] Ibid.
[xxxi] Armstrong, M.P. and A.J. Ruggles. "Geographic Information Technologies and Personal Privacy." Cartographica, Vol. 40, Iss. 4. Winter 2005. Pp. 63-73.

A Study on the Re-Identifiability of Dutch Citizens

Matthijs R. Koot, Guido van 't Noordende, and Cees de Laat

University of Amsterdam, Informatics Institute,
Science Park 107, 1098 XG Amsterdam, Netherlands
{koot,noordende,delaat}@uva.nl

Abstract. This paper analyses the re-identifiability of Dutch citizens by various demographics. Our analysis is based on registry office data of 2.7 million Dutch citizens, ∼16% of the total population. We provide overall statistics on re-identifiability for a range of quasi-identifiers, and present an in-depth analysis of quasi-identifiers found in two de-identified data sets. We found that 67.0% of the sampled population is unambiguously identifiable by date of birth and four-digit postal code alone, and that 99.4% is unambiguously identifiable if date of birth, full postal code and gender are known. Furthermore, two quasi-identifiers we examined from real-life data sets turn out to unambiguously identify a small fraction of the sampled population. As far as we are aware, this is the first re-identifiability assessment of Dutch citizens that uses authoritative registry office data.

Key words: re-identification, data anonymity

1 Introduction

These days, large amounts of data about citizens are stored in various data sets, spread over databases managed by different organisations all around the world. Data about individual citizens drives policy research on all sorts of topics: finances, health and public administration, to name a few. Using personally identifiable information outside the purpose for which it was originally collected is prohibited in general by EU directive 95/46/EC on data protection. De-identification techniques are often used to remove identifying information from data sets, while attempting to retain as much useful information as possible, for example to still allow (statistical) analysis involving demographics.

Most data sets can therefore not be called completely anonymised, even if they are claimed to be; especially for *microdata*, i.e., data consisting of entries that map to single persons, but from which identifying parts are removed, a risk exists that entries can be de-anonymised when sufficient additional information is available. Our research deals with the question of which pieces of partially identifying information can, when combined, lead to re-identification. Such a combination of partially identifying information is called a quasi-identifier. This paper uses real registry office data of citizens in the Netherlands, to experimentally assess the re-identifiability of Dutch citizens using quasi-identifiers found in real-world data sets.

A seminal work on re-identification is due to Latanya Sweeney [14]. Using 1990 U.S. Census summary data, she established that 87% of the US population was uniquely identifiable by a quasi-identifier (QID) composed of three demographic variables [13, 14]:

Definition 1. $QID_{example}$ = { *Date-of-Birth + gender + 5-digit ZIP* }

In Massachusetts (U.S.) the Group Insurance Commission provides and administers health insurance to state employees. Sweeney legitimately obtained a de-identified data set containing medical information about Massachusetts’ employees from them, including details about ethnicity, medical diagnoses and medication [14]. The data set contained the variables described in $QID_{example}$. Sweeney also legitimately obtained the identified 1997 voter registration list from the city of Cambridge, Massachusetts, which contained the same variables. By linking both data sets, it turned out to be possible to re-identify medical records, including records related to Massachusetts’ governor of that time.

Sweeney proposed k-anonymity, a test asserting that for each value of a quasi-identifier in a data set, at least k records must exist with that same value and be indistinguishable from each other. This introduces a minimal level of uncertainty in re-identification: assuming no additional information is available, each record may belong to any of at least k individuals.

We analyze the (re-)identifiability of Dutch citizens by looking at demographic characteristics such as postal code and (part of the) date of birth. By ‘citizen’ we refer to a person who is registered as an inhabitant of the Netherlands. We examine the re-identifiability only in the context of linking the data sets that are described in this paper, and not using any additional outside information. For this paper, we limit ourselves to quasi-identifiers that we believe are most likely to be found in (identified) data sets elsewhere, based on commonly collected demographics. Regarding two real-life data sets, we only provide an assessment *of two specific quasi-identifiers*; other quasi-identifiers exist in those data sets, e.g. involving ethnicity and marital status, which are not discussed in this paper.

This paper is structured as follows: section 2 describes our approach; section 3 lists the results; section 4 describes related work and section 5 discusses the results.

2 Background

The Netherlands consists of 12 provinces and 441 municipalities of varying size [5]. A municipality is an administrative region that typically spans several villages or cities. Municipal registry offices are the official record-keepers of persons residing in the Netherlands, and maintain identified data about them. De-identified data about individual citizens is available in number of research databases. To illustrate our analysis we picked two, which we describe below. In section 3 we assess, amongst others, re-identifiability of entries in these data sets.

2.1 Example Data Sets

The Dutch *National Medical Registration* (LMR) is a data collection program established in 1963, in which hospitals in the Netherlands participate by periodically sending in copies of medical and administrative information about hospital admissions and day care treatment. Example purposes of the LMR are the analysis of the effects of treatment, performance comparison between hospitals, and epidemiological studies. The LMR is currently managed by the Dutch Hospital Data foundation[1]. Statistics Netherlands, the Dutch organisation for conducting statistical studies on behalf of the Dutch government, also receives annual copies of the LMR for research purposes [6]. External researchers can currently request access to the records collected during 2005 and 2007 [2, 4]. These data sets contain only records about Dutch citizens; records about other patients are omitted. Each record in the LMR describes the hospital admission or day care treatment of a single individual, and multiple records may be present per individual. The 2005 and 2007 data sets each contain approximately 2.5 million records.

The Dutch *Welfare Fraud Statistics* (BFS) data set at Statistics Netherlands contains records about investigations on suspected welfare fraud of Dutch citizens [3]. Each record in the data set relates to a single, completed investigation, and multiple records may be present per person. The information in the data set is provided by municipalities. Between 2002 and 2007, the average number of records (cases) per year was 38,161[2]. The BFS data set contains different information at a different granularity than the LMR data set, which is the reason we selected it as a second example. For example, the LMR data set contains information about postal code, whereas BFS does not.

Re-identified records from the BFS data set could be abused to embarrass or discriminate citizens that have been subject of fraud investigation. Similarly, re-identified records from the LMR data set could be abused to embarrass or discriminate people based on medical history or medical conditions, potentially negatively impacting job or insurance prospects. Such consequences are at the disposal of the person possessing the (re-)identified records.

2.2 Approach and Terminology

A data set containing information about persons is said to be *de-identified* if direct identifiers like social security number, phone numbers, names and house numbers are omitted. A *quasi-identifier* is a variable or combination of variables which, although perhaps not intended or expected to identify individuals, can in practice be used for that purpose.

[1] `http://www.dutchhospitaldata.nl`

[2] Source: http://statline.cbs.nl

A quasi-identifier may unambiguously identify a single individual, or reduce the number of possibilities to some small set of k individuals, the *anonymity set* [12]. A de-identified data set containing one or more quasi-identifiers can be *re-identified* by linking records to an *identified* data set containing the same quasi-identifying variable(s).

We assessed the (re-)identifiability of Dutch citizens by using quasi-identifiers composed of information about postal code, date of birth and gender information. We used registry office data of approximately 2.7 million persons, ∼16% of the total population, obtained from 15 of 441 Dutch municipalities. The 15 municipalities and number of citizens are shown in table 1. The sample contains small, mid-size and large municipalities. Although this selection is not random (selected by size) or necessarily representative for the whole population, we considered the selection appropriate for our analysis, since it enables us to assess whether differences in re-identifiability are observable for small municipalities compared to large municipalities that contain a city, for example. The municipalities chosen are spread over the country, such that there is no obvious bias due to geographical location of the municipalities in the countries - although the largest cities, Amsterdam, Rotterdam, and Den Haag, are located in the west of the Netherlands which is known as the most densely populated area of the Netherlands, called the “Randstad”.

We requested a (nameless) listing of gender, full postal code and full date of birth of all citizens of 30 municipalities, and eventually obtained records of 15 municipalities, totalling approximately 2.7 million citizens. The remainder of this paper is based on analysis of this data. We distinctly discuss data only at municipal level; i.e. ‘Amsterdam’ refers to the *municipality of Amsterdam* rather than the *city of Amsterdam*.

We primarily focus on quasi-identifiers that match the LMR and BFS examples in this paper. The results, however, apply to *any* data set that contains these quasi-identifiers. We did not attempt to obtain access to data from the example data sets, since for our purposes it suffices to know which possible quasi-identifying variables they contain, and this information is available from public documents [2–4].

2.3 Data Quality

Data from municipal registry offices is relied upon during transactions between the Dutch government and its citizens, including the process of passport issuance. Registry office data is not free of error: data may be inconsistent with reality due to e.g. failure of citizens to report changes timely and truthfully, typographical errors and software errors [10]. The registry offices are required to undergo a periodical audit, which includes an integrity check of a random sample of the electronic person records. Each record from that sample is matched against other official files associated with the person whom the record is about, such as birth certificates. Each variable containing an incorrect value is counted as a single error, and the maximum allowed rate for errors in ‘essential’ fields like DoB and postal code is 1% of the sample set size: to pass the test, a 100-record

Table 1. Municipalities included in our study (ordered by size)

Municipality	# of citizens
Amsterdam	766,656
Rotterdam	591,046
Den Haag	487,582
Utrecht	305,845
Nijmegen	161,882
Enschede	156,761
Arnhem	147,091
Overbetuwe	45,548
Geldermalsen	26,097
Diemen	24,679
Reimerswaal	21,457
Enkhuizen	18,158
Simpelveld	11,019
Millingen a/d Rijn	5,915
Terschelling	4,751
TOTAL:	2,774,476

sample cannot contain more than 1 error in essential fields. The sample size depends on the municipality size. During the 2002-2005 audit cycle, 339 of the 370 (92%) audited municipalities passed this test [10]. This suggests that Dutch registry offices are generally a reliable source of data. During our own data sanity checks we removed 11 records containing a postal code from outside the sampled municipalities, as those records would have caused false outliers[3]; the remainder of the records passed all sanity checks.

2.4 Postal Codes in the Netherlands

In the Netherlands, a postal code consist of a four-digit number and a two-character extension — e.g. "1098 XG", the postal code of our institution. The four-digit number is referred to as '4-Position PostalCode' (PC4), and corresponds to exactly one town (city, village). A town may be divided into multiple PC4-regions: for example, our data contains eighty different PC4-regions for the city of Amsterdam, "1098" being one of them.

The two-character extension indicates a street, but often also a specific odd or even range of house numbers *within* that street. The full postal code is referred to as '6-Position PostalCode' (PC6). A combination of full (PC6) postal code and house or P.O. box number uniquely indicates a postal delivery address in the Netherlands.

[3] These cases may be related to moving citizens, e.g. pending handover of data between municipalities.

3 Results

This section describes the results of our analysis. Section 3.1 describes an overall analysis of our input data. From the result data it becomes clear which combinations of variables can be used to single out individuals or small groups of citizens, and which combinations pose less of a privacy risk in that sense. Section 3.2 describes the potential re-identifiability of citizens in the LMR data set. Section 3.3 analyses the potential re-identifiability of citizens in the BFS data set. Throughout this paper, we use the following notations: *QID*=Quasi-IDentifier, *DoB*=Date of Birth, *YoB*=Year of Birth, *MoB*=Month of Birth.

By 'quasi-identifier' we refer to abstract variables, by 'quasi-identifier value' to a valuation of those variables. We use rounded values for the sake of readability. For each quasi-identifier, we counted the number of different (distinct) values in the data — this is the number of anonymity sets; the number of people sharing a specific quasi-identifier value represents the anonymity set size.

In addition to mean values, we provide quartiles and min-max values to give an indication of how a quasi-identifier maps citizens in anonymity sets of rather diverse or rather similar size[4]. We chose quartiles as a means to indicate the value distribution while maintaining some brevity and readability of tables. Another choice could have been made (e.g., for deciles or percentiles), however, none has a definite advantage over the other. By using quartiles we can state properties of the distribution of anonymity set sizes such as "at most 25% of the anonymity sets are smaller than <1st quartile>" and "at most 50% of the anonymity sets are smaller than <median>".

3.1 Analysis over Aggregated Data

This section describes the results of an analysis of the combined data of the citizens of all municipalities listed in table 1. By including both small and large municipalities, covering the smallest villages (the smallest having two inhabitants) and largest cities (the largest having 684,926 inhabitants) in the Netherlands, the minimum and maximum anonymity set sizes represent the worst and best cases we expect to be found *anywhere* in the Netherlands. Furthermore, the statistics over the combined data indicate how strongly identifiable a quasi-identifier is for the overall population.

Throughout this paper, k denotes the anonymity set size; $k = 1$ means that some quasi-identifier value unambiguously identifies some individual, $k = 2$

[4] The lower (1st) quartile is the value separating the lower 25% of the values; the median value (2nd quartile) separates the higher half of the values from the lower half; the upper (3rd) quartile separates the higher 25% of the values. To illustrate: for both (100,100,100,100,100) and (1,1,1,1,496), the mean value is 100, while both sets are obviously very different. For the former set, all three quartiles are 100, as are both the minimum and maximum: this shows that the distribution is uniform. For the latter set of numbers, minimum value and all quartiles are 1, but the maximum value is 496: this shows that the distribution is skewed. Or, in our context, that the quasi-identifier maps citizens into anonymity sets of different sizes.

means that the value is shared by two individuals, and so on. Table 2 shows the statistical characteristics of anonymity set size k for various (potential) quasi-identifiers. The column *'# of sets'* contains the number of different values present in our data for a given quasi-identifier, i.e., the number of anonymity sets. Generally, the higher this number, the weaker privacy, because the anonymity sets will tend to be smaller in that case. The min/max values denote the size of the smallest and largest anonymity set.

Table 2. Anonymity set size k for various (potential) quasi-identifiers

Quasi-identifier:	# of sets	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
PC4	388	2	3,278	7,090	7,188	10,300	22,330
PC6	66,883	1	24	35	41	50	1,322
PC4+DoB	2,267,700	1	1	1	1	1	42
PC6+DoB	2,759,422	1	1	1	1	1	5
PC4+gender	776	1	1,652	3,536	3,594	5,151	11,730
PC6+gender	133,012	1	11	18	21	25	954
gender+YoB	221	1	5,219	14,570	12,550	19,740	25,580
gender+YoB+MoB	2,699	1	397	1,177	1,028	1,594	2,326
gender+YoB+MoB+PC4[a]	635,679	1	2	3	4	6	40
gender+YoB+MoB+municipality[b]	34,790	1	6	18	80	96	733
gender+DoB	71,318	1	21	40	39	54	571
gender+DoB+PC4	2,488,828	1	1	1	1	1	22
gender+DoB+PC6	2,766,475	1	1	1	1	1	4
town+gender	134	1	222	1116	20,700	3259	347,100
town+YoB	5,642	1	6	29	492	101	14,270
town+YoB+MoB	49,207	1	2	5	56	20	1,262
town+DoB	463,134	1	1	2	6	7	419
town+YoB+gender	10,492	1	4	17	264	60	7,515
town+YoB+MoB+gender	83,172	1	1	3	33	14	695
town+DoB+gender	697,875	1	1	2	4	5	226

[a] QID_A, see section 3.2.
[b] QID_B, see section 3.3.

As an example, the median anonymity set size of PC6 is 35, the minimum size is 1 and the maximum size is 1,322. This means that at most half of the values for PC6 have anonymity sets of sizes between 1 and 35, and that the sizes of the anonymity sets in the upper half are between 35 and 1,322.

Looking at the quartiles, it becomes clear that some quasi-identifiers are particularly strong, by which we mean that a large portion of the anonymity sets established by that quasi-identifier are of small size (e.g. $k = 1$ or $k \leq 5$). For example, for $\{PC4{+}DoB\}$, table 2 shows an anonymity set size of $k = 1$ for up to the 3rd quartile, meaning that 75% of the quasi-identifier values unambiguously identify a citizen. Looking at the lower quartiles, it also becomes clear that some quasi-identifiers are weaker identifiers: for $\{PC4\}$, only at most 25% of the sets are of size $k \leq 3,278$; for $\{gender + YoB\}$, at most 25% of the sets are of size $k \leq 5,219$. Overall, it turns out that quasi-identifiers containing both PC4 or PC6, as well as date of birth, are most identifying.

We were surprised to find that PC4 postal codes exist which are shared by only two citizens: we had expected that PC4 codes always map to relatively large numbers of citizens. Upon closer inspection, it appears that the data is accurate: it represents the inhabitants of a new construction area in the harbour of Rotterdam. These pioneering citizens turn out to be unambiguously identifiable nation-wide by only their $\{PC4 + gender\}$ or $\{town + gender\}$ — albeit only until other citizens officially move in.

Table 2 also clearly shows that the two-character extension to the PC4 postal code, making PC6, strongly increases identifiability: the median anonymity set size for $\{PC4\}$ is 7,090, for $\{PC6\}$ only 35.

Table 3. Number of Dutch citizens per anonymity set size, for various quasi-identifiers

Quasi-identifier:	$k = 1$	$k \leq 5$	$k \leq 10$	$k \leq 50$	$k \leq 100$
PC4	0	9	19	345	996
PC6	429	6,109	25,103	1,459,939	2,354,255
PC4+DoB	1,861,081	2,754,465	2,765,932	2,774,476	-
PC6+DoB	2,744,653	2,774,476	-	-	-
PC4+gender	4	27	103	889	2,555
PC6+gender	1,854	31,262	184,803	2,342,242	2,629,017
gender+YoB	5	14	53	250	516
gender+YoB+MoB	55	356	712	4,478	9,674
gender+YoB+MoB+PC4[a]	137,035	279,100	2,196,950	2,774,476	-
gender+YoB+MoB+municipality[b]	2,186	22,565	59,597	244,152	619,671
gender+DoB	2,014	14,506	40,322	1,392,622	2,725,472
gender+DoB+PC4	2,240,461	2,765,067	2,772,205	2,774,476	-
gender+DoB+PC6	2,758,578	2,774,476	-	-	-
town+gender	4	4	28	372	896
town+YoB	499	3,172	7,225	50,985	103,145
town+YoB+MoB	10,083	61,073	112,850	287,173	394,844
town+DoB	185,042	596,769	1,045,559	2,730,668	2,750,700
town+YoB+gender	1,153	7,195	16,333	102,018	150,135
town+YoB+MoB+gender	22,260	109,126	170,351	398,601	826,744
town+DoB+gender	288,409	1,029,601	1,813,559	2,750,669	2,764,050

[a] QID_A, see section 3.2.
[b] QID_B, see section 3.3.

Whereas table 2 focusses on the size distribution of the anonymity sets, table 3 shows the actual number of *citizens* found in those anonymity sets. The larger the value in columns '$k = 1$', '$k \leq 5$' and possibly '$k \leq 10$', the larger the portion of the population that is covered by anonymity sets of those (small) sizes and the stronger the quasi-identifier identifies citizens. The numbers confirm that $\{PC6 + DoB\}$ is a strong identifier, because here nearly all citizens have $k = 1$; $\{PC6\}$ alone is not a strong identifier, because only a very small portion of the citizens have $k \leq 10$ (compared to $k \leq 50$). We also included columns for a few larger set sizes ($k \leq 50$ and $k \leq 100$) for illustrative purposes. For example, only 896 out of 2.7 million citizens are identifiable to a group of ≤100 by $\{town + gender\}$, so by themselves, those variables do not pose a significant privacy risk for most citizens. For readability, we replaced numbers by '-' when the total population is reached at some k.

From the numbers for quasi-identifier $\{gender + DoB + PC6\}$ it follows that approximately 99.4% of the Dutch citizens in our data set (2,758,578 out of 2,774,476) can be unambiguously identified by $\{gender + DoB + PC6\}$; and it turns out that 67.0% (1,861,081 out of 2,774,476) can still be unambiguously identified by $\{PC4 + DoB\}$.

3.2 Case: National Medical Registration

The LMR contains a large amount of information about hospital admissions and day care treatment: amongst others, it contains fields describing the hospital, the patient's insurance type, diagnosis codes, the treatment that was provided and the medical specialisms and disciplines involved [2, 4]. This information could be privacy-sensitive and it is generally treated as such, even when de-identified. The LMR data set also contains demographic data about the patient. In particular, the LMR contains the following quasi-identifier:

Definition 2. $QID_A = \{\ PC4\ +\ gender\ +\ YoB\ +\ MoB\ \}$

Our data contains 635,679 different anonymity sets for QID_A. We use k_A to denote the anonymity set sizes for this quasi-identifier. 137,035 people, ∼4.8%, are unambiguously identifiable by QID_A, that is, they are the only person in the anonymity set, which thus has k_A=1. Furthermore, we found 212,536 citizens to have $k_A = 2$; $260,244$ to have $k_A = 3$ and 282,644 to have $k_A = 4$ (most common size). Table 4 lists the statistical properties of the size of the anonymity sets established by this quasi-identifier. The municipality size is included for quick reference.

The numbers show that there is no large difference in anonymity between citizens of different-sized municipalities: the range of the medians is 1–5. The highest median anonymity set size is found in Amsterdam, the lowest is found in Terschelling. The latter means that half of the QID_A values found in Terschelling unambiguously identify a citizen.

The municipality size (column *'# of citizens'*) and median anonymity set size (column *'Median'*) have a Pearson correlation coefficient of .60. The single largest anonymity set is found in Amsterdam and is of size 40. Based on the numbers shown in table 3, the total percentage of citizens identifiable to a group of 10 or less by this quasi-identifier is ∼79.1% (2,196,950 out of 2,774,476).

Figure 1 visually represents the numbers in table 4. Some large anonymity sets exist as outliers, especially for larger municipalities, but overall anonymity is approximately the same (poor) over all municipalities.

Note that there is a difference in constraints between registry office data and the hospital admission data set: whereas the year of birth is allowed to be zero by the Dutch registry offices — e.g. for immigrants about whom the date of birth is not fully known —, the LMR requires it to be non-zero and be estimated if unknown [1]. This means that LMR-records about a person who is officially registered with zero year of birth (in our data set we only found 3) will *not* be re-identified by quasi-identifiers involving the year of birth. On the other hand, the quality of data from the LMR and BFS depends on their sources (hospitals and municipalities); it is not asserted whether each record accurately represents reality [2–4] – note that any mismatch (error) prevents linkability, and thus improves privacy for the involved individual.

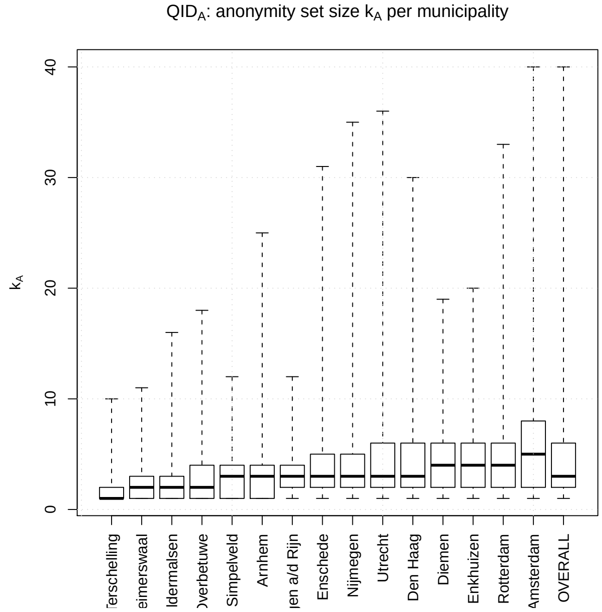


Fig. 1. Box-and-whisker plot showing anonymity set sizes k_A, per municipality. Whiskers denote the minimum and maximum values; the boxes are defined by lower and upper quartiles and the median value is shown.

3.3 Case: Welfare Fraud Statistics

In the BFS data set, we recognised the following as a potential quasi-identifier:

Definition 3. QID_B = { *municipality* + *gender* + *YoB* + *MoB* }

Table 4. Statistical summary of k_A, divided by municipality (ordered by median)

Municipality:	# of citizens	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Amsterdam	766,656	1	2	5	6	8	40
Rotterdam	591,046	1	2	4	5	6	33
Enkhuizen	18,158	1	2	4	4	6	20
Diemen	24,679	1	2	4	4	6	19
Den Haag	487,582	1	2	3	4	6	30
Utrecht	305,845	1	2	3	4	6	36
Enschede	156,761	1	2	3	4	5	31
Nijmegen	161,882	1	2	3	4	5	35
Arnhem	147,091	1	1	3	3	4	25
Millingen a/d Rijn	5,915	1	2	3	3	4	12
Simpelveld	11,019	1	1	3	3	4	12
Geldermalsen	26,097	1	1	2	2	3	16
Overbetuwe	45,548	1	1	2	3	4	18
Reimerswaal	21,457	1	1	2	2	3	11
Terschelling	4,751	1	1	1	1	2	10
OVERALL	2,774,476	1	2	3	4	6	40

Our data contains 34,790 different anonymity sets for QID_B. 2,186 people, ~0.07%, are unambiguously identifiable by QID_B. Furthermore, we found 3,552 citizens to have $k_B = 2$; 5,064 to have $k_B = 3$ and 5,508 to have $k_B = 4$. The total percentage of citizens identifiable to a group of 10 or less is ~2.14% (59,597 out of 2,774,476). The single largest anonymity set is found in Amsterdam and is of size 733.

Table 5 lists the statistical properties of k_B per municipality. The numbers show that regarding the BFS, large differences in anonymity exist between citizens of different-sized municipalities: the range is 1–733. The highest median anonymity set size is 310, found in Amsterdam, the lowest is 2, found in Terschelling. Municipality size and median anonymity set size have a Pearson correlation coefficient of .99; the median anonymity set size is rather constant at ~0.04% (1/2,500) of the population size.

Figure 2 visually represents the numbers in table 5. Note that the range on the vertical axis is much larger than in figure 1. It is clear that citizens from large municipalities tend to have much stronger anonymity than citizens from small municipalities, which is something to remember when dealing with de-identified data about citizens from small municipalities.

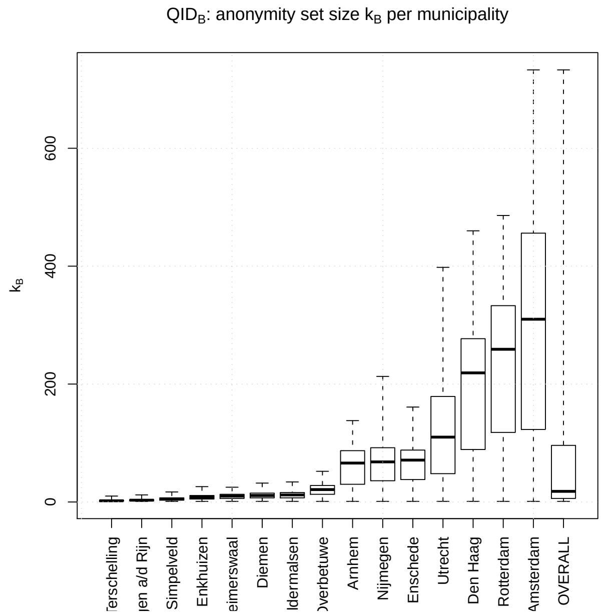


Fig. 2. Box-and-whisker plot showing anonymity set sizes k_B, per municipality. Whiskers denote min-max values.

Table 5. Statistical summary of k_B, divided by municipality (ordered by median)

Municipality:	# of citizens	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Amsterdam	766,656	1	123	310	296	456	733
Rotterdam	591,046	1	118	259	228	333	486
Den Haag	487,582	1	89	219	188	277	460
Utrecht	305,845	1	48	110	121	179	398
Enschede	156,761	1	38	71	64	88	161
Nijmegen	161,882	1	36	68	66	92	213
Arnhem	147,091	1	30	66	60	87	138
Overbetuwe	45,548	1	13	21	20	28	52
Geldermalsen	26,097	1	7	12	12	16	34
Diemen	24,679	1	7	11	11	15	32
Reimerswaal	21,457	1	6	10	10	13	25
Enkhuizen	18,158	1	5	8	8	11	26
Simpelveld	11,019	1	3	5	5	7	17
Millingen a/d Rijn	5,915	1	2	3	3	4	12
Terschelling	4,751	1	1	2	3	3	10
OVERALL	2,774,476	1	6	18	80	96	733

4 Related Work

Various extensions and enhancements on k-anonymity have been devised, such as l-diversity [8] and t-closeness [7]. k-anonymity attempts to make it hard for an adversary to link records to individuals, i.e., it protects against identity disclosure, but still allows adversaries focussing on some subset of k-anonymous records to make educated guesses about specific variables by looking at the distribution of those variables. l-diversity and t-closeness, for example, attempt to also make it hard for an adversary to do this, and are applied as a complement to k-anonymity.

In 2006, Arvind Narayanan and Vitaly Shmatikov demonstrated new statistical de-anonymisation attacks against the publicly released Netflix Prize data set containing de-identified movie ratings of about 500,000 subscribers of Netflix [9]. The authors showed that, given a little prior knowledge of a certain subscriber, it is possible to identify, with high certainty, records related to that subscriber in the anonymised data set. The authors show that their findings apply in general to multi-dimensional microdata.

In his short paper revisiting Sweeney's work, Philippe Golle mentions a lack of available details about the data collection and analysis involved Sweeney's work as a reason for being unable to explain the big difference between the outcome between both studies: in Golle's study of the 2000 U.S. Census data, only ∼63% of U.S. citizens turned out to be uniquely identifiable, as opposed to ∼87% that Sweeney determined by studying the 1990 U.S. Census data. This may be attributed to inaccuracies in the source data. By using registry office data we are confident that our results (for the Dutch population) are likely to be highly accurate.

5 Discussion

We determined the identifiability of Dutch citizens using information about postal code, date of birth and gender. We studied real registry office data of approximately 2.7 million citizens, ∼16% of the total population, obtained from 15 of 441 Dutch municipalities of varying size. We assessed the re-identifiability of records about these individuals in known data sets about hospital admissions and welfare fraud.

It turns out that approximately 99.4% of the sampled population is unambiguously identifiable using PC6 postal code, gender and date of birth, and 67.0% by PC4 and date of birth alone. Regarding the quasi-identifier found in the LMR data set, approximately 4.8% of the sampled population is unambiguously identifiable and 79.1% is identifiable to a group of 10 or less. Regarding the quasi-identifier found in the BFS data set, approximately 0.07% of the sampled population is unambiguously identifiable and 2.14% is identifiable to a group of 10 or less; for small municipalities, however, the anonymity set sizes become much smaller and re-identifiability higher.

As far as we know, we are the first to study re-identifiability using authoratitive registry office data. Comparing to Sweeney and Goll (who used census data), our study uses registry office data, which is the authoritative data source during passport issuance. Our data was not prone to the intricacies of survey-based data collection. We only cover a portion of the Dutch citizens, ∼16%, but are confident that the results for that portion are accurate. We also provide the minimum and maximum anonymity set sizes that can be expected to be found anywhere in the Netherlands.

The results suggest that, considering the quasi-identifier in the National Medical Registration data set, someone who is able to access registry office data can re-identify a large portion of records with relatively high certainty. Considering the quasi-identifier in the Welfare Fraud Statistics data set, the re-identification risk is generally lower, but strongly depends on municipality size.

One could argue about the plausibility of the threat scenario underlying the two cases we picked: we assume an adversary who is able to access non-public records from both registry offices and Statistics Netherlands. Access to the data sets at Statistics Netherlands is only granted to qualified applicants, for specific purposes, under specific conditions of confidentiality [15]. Thus, obtaining data may require an investment that is disproportional to the expected gain of re-identifying records from these particular data sets to begin with. We note, however, that our results apply to *any* de-identified data set containing the assessed quasi-identifiers. Also, registry offices are not the only source for identified data, and *any* identified database containing these quasi-identifiers with sufficiently large coverage of the total population may be used; suitable data sets may also exist at, e.g., information brokers, marketing agencies and public transport companies. Besides, preventing registry office data itself from being used for re-identification may be difficult: the 441 municipalities are autonomous gatekeepers to their citizen's data and that citizen data is already necessarily exchanged on a regular basis for a variety of legitimate purposes [11]. It is hard to protect data that has many legitimate users and uses.

We believe that our results are useful as input for privacy impact assessments involving data about Dutch citizens. It remains a matter of policy what value of k can be considered *sufficiently strong* anonymity for particular personal information.

References

1. Tieto Netherlands Healthcare BV. *LMR Gebruikershandleiding*, 2009.
2. CBS. *Documentatierapport Landelijke Medische Registratie (LMR) 2005V1*, March 2007.
3. CBS. *Documentatierapport Bijstandsfraudestatistiek (BFS) 200901-06V1*, November 2009.
4. CBS. *Documentatierapport Landelijke Medische Registratie (LMR) 2007V1*, July 2009.

5. CBS. Website: Cbs - gemeentelijke indeling op 1 januari 2009, 2009. `http://www.cbs.nl/`.
6. CBS. Website: Cbs - ziekenhuisopnamen - dataverzameling, 2009. `http://www.cbs.nl/`.
7. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *23rd International Conference on Data Engineering*, pages 106–115, 2007.
8. Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3, 2007.
9. Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.
10. Atzo Nicolaï. Kst99754: Modernisering gemeentelijke basisadministratie persoonsgegevens, 2006.
11. NVVB. *Schema voor schriftelijke verzoeken om gegevensverstrekking uit de GBA*, January 2010.
12. Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, December 2009. v0.32.
13. Latanya Sweeney. Uniqueness of simple demographics in the u.s. population, 2000.
14. Latanya Sweeney. *Computational disclosure control: a primer on data privacy protection*. PhD thesis, Massachusetts Institute of Technology, 2001. Supervisor: Abelson, Hal.
15. Leon Willenborg and Ton de Waal. *Statistical Disclosure Control in Practice*, volume 111 of *Lecture Notes in Statistics*. Springer, 1996. ISBN: 978-0-387-94722-8.

Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL

Christopher Soghoian
csoghoian@gmail.com

Sid Stamm
sidstamm@gmail.com

"Cryptography is typically bypassed, not penetrated."
— Adi Shamir [47]

"Just because encryption is involved, that doesn't give you a talisman against a prosecutor. They can compel a service provider to cooperate."
— Phil Zimmerman [48]

ABSTRACT

This paper introduces the *compelled certificate creation attack*, in which government agencies may compel a certificate authority to issue false SSL certificates that can be used by intelligence agencies to covertly intercept and hijack individuals' secure Web-based communications. Although we do not have direct evidence that this form of active surveillance is taking place in the wild, we show how products already on the market are geared and marketed towards this kind of use—suggesting such attacks may occur in the future, if they are not already occurring. Finally, we introduce a lightweight browser add-on that detects and thwarts such attacks.

1. INTRODUCTION

Consider a hypothetical situation where an American executive is in France for a series of trade negotiations. After a day of meetings, she logs in to her corporate webmail account using her company-provided laptop and the hotel wireless network. Relying on the training she received from her company's IT department, she makes certain to look for the SSL encryption lock icon in her web browser, and only after determining that the connection is secure does she enter her login credentials and then begin to upload materials to be shared with her colleagues. However, unknown to the executive, the French government has engaged in a sophisticated man-in-the-middle attack, and is able to covertly intercept the executive's SSL encrypted connections. Agents from the state security apparatus leak details of her communications to the French company with whom she is negotiating, who use the information to gain an upperhand in the negotiations. While this scenario is fictitious, the vulnerability is not.

The security and confidentiality of millions of Internet transactions per day depend upon the Secure Socket Layer (SSL)/Transport Layer Security (TLS) protocol. At the core of this system are a number of *Certificate Authorities* (CAs), each of which is responsible for verifying the identity of the entities to whom they grant SSL certificates. It is because of the confidentiality and authenticity provided by the CA based *public key infrastructure* that users around the world can bank online, engage in electronic commerce and communicate with their friends and loved ones about the most sensitive of subjects without having to worry about malicious third parties intercepting and deciphering their communications.

While not completely obvious, the CAs are all trusted equally in the SSL public key infrastructure, a problem amplified by the fact that the major web browsers trust hundreds of different firms to issue certificates for any site. Each of these firms can be compelled by their national government to issue a certificate for any particular website that all web browsers will trust without warning. Thus, users around the world are put in a position where their browser entrusts their private data, indirectly, to a large number of governments (both foreign and domestic) whom these individuals may not ordinarily trust.

In this paper, we introduce a new attack, the *compelled certificate creation attack*, in which government agencies compel (via a court order or some other legal process) a CA to issue false certificates that are then used by law enforcement and intelligence agencies to covertly intercept and hijack individuals' secure communications.

We also show how currently available surveillance products are advertised in a way that suggests that this attack is more than a theoretical concern, but is likely in active use; at least one private company is supplying government customers with specialized covert network appliances specifically designed to intercept SSL communications using deceptively created certificates.

In order to protect users from these powerful government adversaries, we introduce a lightweight defensive browser add-on that detects and thwarts such attacks. Finally, we use reductive analysis of governments' legal capabilities to perform an adversarial threat model analysis of the attack and our proposed defensive technology. We believe that this form of legal threat model analysis is itself new to the computer security literature.

In section 2 we provide a brief introduction to CAs, web

Both authors of this paper have written it in their personal capacities as academic researchers. All statements, opinions and potential mistakes are their own, and do not reflect the official positions of their respective employers.

browsers and the man-in-the-middle attacks against them. In section 3 we discuss the presence of government-controlled CAs in the browsers. In section 4, we describe the compelled certificate creation attack and then in section 5, we present evidence that suggests it is being used. In section 6 we introduce our browser based add-on, and in section 7, we analyze its effectiveness via a threat model based analysis. Finally, we present related work in section 8 and conclude in section 9.

2. CERTIFICATE AUTHORITIES AND THE BROWSER VENDORS

In this section, we provide a brief overview of the roles played by the Certificate Authorities in the public key infrastructure, the browser vendors in picking the certificate authorities that they include in the browsers, and existing man-in-the-middle-attack techniques that circumvent SSL based security.

2.1 Certificate Authorities

> *"[Browser vendors] and users must be careful when deciding which certificates and certificate authorities are acceptable; a dishonest certificate authority can do tremendous damage."*
> — RFC 2246, The TLS Protocol 1.0 [15]

CAs play a vital role in the SSL *public key infrastructure* (PKI). Each CA's main responsibility is to verify the identity of the entity to which it issues a certificate.[1] Thus, when a user visits `https://www.bankofamerica.com`, her browser will inform her that the bank's certificate is valid, was issued by VeriSign, and that the website is run by Bank of America. It is because of the authenticity and confidentiality guaranteed by SSL that the user can continue with her transaction without having to worry that she is being phished by cyber-criminals.

CAs generally fall into one of three categories: Those trusted by the browsers ("root CAs"), those trusted by one of the root CAs ("intermediate CAs" or "subordinate CAs"), and those neither trusted by the browsers nor any intermediate CA ("untrusted CAs"). Furthermore, intermediate CAs do not necessarily have to be directly verified by a root CA — but can be verified by another intermediate CA, as long as the *chain of trust* eventually ends with a root CA.[2]

From the end users' perspective, root CAs and intermediate CAs are functionally equivalent. A website that presents a certificate signed by either form of CA will cause the users' browser to display a lock icon and to change the color of the location bar. Whereas certificates verified by an untrusted CA and those self-signed by the website owner will result in the display of a security warning, which for many non-technical users can be scary [40], confusing, and difficult to bypass in order to continue navigating the site [56].

As the CA system was originally designed and is currently implemented, all root CAs are equally trusted by the browsers. That is, each of the 264 root CAs trusted by Microsoft, the 166 root CAs trusted by Apple, and the 144 root CAs trusted by Firefox are capable of issuing certificates for any website, in any country or top level domain [18]. For example, even though Bank of America obtained its current SSL certificate from VeriSign, there is no technical reason why another CA, such as GoDaddy, cannot issue another certificate for the same site to someone else. Should a malicious third party somehow obtain a certificate for Bank of America's site and then trick a user into visiting their fake web server (for example, by using DNS or ARP spoofing), there is no practical, easy way for the user to determine that something bad has happened, as the browser interface will signal that a valid SSL session has been established.[3]

Of course, GoDaddy is extremely unlikely to knowingly provide such a certificate to a malicious third party. Doing so would almost certainly lead to significant damage to its reputation, a number of lawsuits, as well as the ultimate threat of having its trusted status revoked by the major web browsers.[4] Therefore, it is in each CAs' self-interest to ensure that malicious parties are not able to obtain a certificate for a site not under their own control.

It is important to note that there are no technical restrictions in place that prohibit a CA from issuing a certificate to a malicious third party. Thus, both the integrity of the CA based public key infrastructure and the security users' communications depend upon hundreds of CAs around the world choosing to do the right thing. Unfortunately, as will soon be clear, any one of those CAs can become the weakest link in the chain.

2.2 Web Browsers

There is no technical standard that specifies how web browsers should select their list of trusted CAs. As a result, each browser vendor has created their own set of policies to evaluate and approve CAs [3, 2, 1]. Since there is no evidence to suggest that any browser has knowingly or incompetently approved a rogue CA, we do not discuss each particular vendors' policies in depth.

[1] The level of verification performed by the CA depends upon the type of certificate purchased. A domain registration certificate can be obtained for less than $15, and will typically only require that the requester be able to reply to an email sent to the administrative address listed in the WHOIS database. Extended Validation (EV) certificates require a greater de of verification.

[2] Dan Kaminsky describes this aspect of the CA chain of trust as: "You can just walk up to a certificate authority and say, 'Yeah, so I spent a lot of money on my CA and it doesn't work with anyone outside my company. Um, here's a pile of money and I promise to be good.' No really, you can just buy a root certificate, effectively. It's not expensive, it's not that difficult, and there's an unknown number of companies out there – not just the certificate authorties but all of the companies that have intermediate certificates – they can all issue certificates for your domain [30]."

[3] Even if the user examines the more complex security information listed in the browser's SSL interface, she will still lack the information necessary to make an informed trust decision. Since GoDaddy is a valid certificate authority and has issued millions of other valid certificates, there is no way for the user to determine that any one particular certificate was improperly issued to a malicious third party.

[4] The browser vendors wield considerable theoretical power over each CA. Any CA no longer trusted by the major browsers will have an impossible time attracting or retaining clients, as visitors to those clients' websites will be greeted by a scary browser warning each time they attempt to establish a secure connection. Nevertheless, the browser vendors appear loathe to actually drop CAs that engage in inappropriate behavior — a rather lengthy list of bad CA practices that have not resulted in the CAs being dropped by one browser vendor can be seen in [38].

What does merit further attention is the method by which the browser vendors deliver and update their list of root CAs and the in-browser user interface provided to end-users to view and manage them.

The major browsers (Internet Explorer, Firefox, Chrome and Safari) have all adopted slightly different policies for managing and displaying the list of trusted CAs: Firefox is the only major browser to maintain its own database of trusted CAs, while the other three browsers instead rely upon a list of CAs provided by the operating system. However, since two of these three browser vendors are also major players in the computer operating system business, the line between browser and operating system tends to be rather blurry.

In years past, Microsoft, like the other vendors, included hundreds of CAs in its Windows operating system *Trusted Root Store*. Users who discovered the relevant user interface were able to view and manage the full list of CAs. However, in response to criticism from large enterprise customers, Microsoft reduced the number of certificates in the trusted store in subsequent OS versions down to just a handful.[5]

It would be easy for a naive user (or security researcher) comparing the various CA databases through the user interfaces provided by Microsoft, Apple and Mozilla to conclude that Microsoft has adopted a far more cautious approach in trusting CAs than its competitors, since the user interface of a fresh installation of Windows Vista or Windows 7 will list less than 15 CAs in the operating system's Trusted Root Store. Unfortunately, this interface is extremely misleading as it does not reveal the fact that Microsoft has opted to trust 264 different CAs. The company's own documentation reveals that:

> "Root certificates are updated on Windows Vista [and Windows 7] automatically. When a user visits a secure Web site (by using HTTPS SSL) [...] and encounters a new root certificate, the Windows certificate chain verification software **checks the appropriate Microsoft Update location for the root certificate**. If it finds it, it downloads it to the system. To the user, the experience is seamless. **The user does not see any security dialog boxes or warnings. The download happens automatically, behind the scenes** [3]."

Thus, any web browser that depends upon Microsoft's Trusted Root Store (such as Internet Explorer, Chrome and Safari for Windows) ultimately trusts 264 different CAs to issue certificates without warning, although only a handful of them are listed in the operating system's user interface. While Microsoft clearly describes this in its online developer documentation [3], no mention of this rather important design decision is made in the browser or the operating system certificate management user interface, where interested users are most likely to look.

2.3 Man in The Middle

> "Any website secured using TLS can be impersonated using a rogue certificate issued by a rogue CA. This is irrespective of which CA issued the website's true certificate and of any property of that certificate."
> — Marc Stevens *et al.* [55]

While an exhaustive explanation of man in the middle attacks against SSL is beyond the scope of this article, we at least provide a brief introduction to the subject. Over the past few years, the SSL protocol has been subject to a series of successful attacks by security researchers, some exploiting flaws in deployed systems while others made use of social engineering and other forms of deception [32, 52, 34, 43, 45].

It is because SSL protected web connections flow over a number of other insecure protocols that it is possible for attackers to intercept and hijack a connection to a SSL protected server (these are known as *man in the middle attacks*). It is only once the browser has received and verified a site's SSL certificate that the user can be sure that her connection is safe.

However, this step alone is often not enough to protect users. Sites that supply self-signed certificates, or that exploit unpatched vulnerabilities in the certificate handling code in the browsers can still trigger the display of the SSL lock icon, yet without providing the user with the associated security protections that they would normally expect.

Security researcher Moxie Marlinspike has repeatedly attacked the SSL based chain of trust, revealing exploits that leverage both browser design flaws, as well as social engineering attacks against end-users. His *sslsniff* [35] and *sslstrip* [36] tools automate the task of performing a man-in-the-middle attacks, and when supplied with a valid SSL certificate (obtained via a rogue CA for example), can be used to intercept users' communications without triggering any browser warnings.

3. BIG BROTHER IN THE BROWSER

Microsoft, Apple and Mozilla all include a number of national government CAs certificates in their respective CA databases.[6] These government CAs, like all other root CAs included by the browsers, must satisfy the requirements detailed in each browser vendor's CA policies, and are included for legitimate reasons: Many governments embed cryptographic public keys in their national ID cards, or do not wish to outsource their own internal certificate issuing responsibilities to private companies.

While it may be quite useful for Estonian users of Internet Explorer to trust their government's CA by default (thus enabling them to easily engage in secure online tasks that leverage their own national ID card), the average resident of Lebanon or Peru has far less to gain by trusting the Estonian government with the blanket power to issue SSL certificates for any website. Thus, users around the world are put in

[5] The former product manager for Internet Explorer told the authors that "a very few enterprises who chose to control their own trust decisions raised concerns regarding a trusted store pre-loaded with 70–100 root CAs as a potential for abuse. For this and several other reason Microsoft has since reduced the number of root certificates in the trusted store [53]."

[6] For example, Microsoft's Root Certificate Program includes the governments of Austria, Brazil, Finland, France, Hong Kong, India, Japan, Korea, Latvia, Macao, Mexico, Portugal, Serbia, Slovenia, Spain, Switzerland, Taiwan, The Netherlands, Tunisia, Turkey, United States and Uruguay [4].

a position where their browser entrusts their private data, indirectly, to a number of foreign governments whom those individuals may not ordinarily trust.

As an illustrative and hypothetical example of what is currently possible the Korean Information Security Agency is able to create a valid SSL certificate for the Industrial and Commercial Bank of China (whose actual certificate is issued by VeriSign, USA), that can hypothetically be used to perform an effective man-in-the-middle attack against users of Internet Explorer.

While this might at first seem like an extremely powerful attack, there are several reasons why governments are unlikely to use their own CAs to perform man in the middle attacks.

First, while *some* governments have succesfully petitioned the browser vendors to include their CA certificates, not all governments have done so. Thus, for example, the governments of Singapore, the United Kingdom and Israel (among many others) do not have state-run CAs that are included by any of the major browsers. These governments are therefore unable to create their own fake certificates for use in intelligence and other law enforcement investigations where snooping on a SSL session might be useful.

Second, due to the fact that the SSL chain of trust is *non-repudiable*, any government using its own CA to issue fake certificates in order to try and spy on someone else's communications will leave behind absolute proof of its involvement. That is, if the Spanish government opts to issue a fake certificate for Google Mail, and the surveillance is somehow discovered, anyone with a copy of the fake certificate and a web browser can independently trace the operation back to the Spanish government.

4. COMPELLED ASSISTANCE

Many governments routinely compel companies to assist them with surveillance. Telecommunications carriers and Internet service providers are frequently required to violate their customers' privacy — providing the government with email communications, telephone calls, search engine records, financial transactions and geo-location information.

In the United States, the legal statutes defining the range of entities that can be compelled to assist in electronic surveillance by law enforcement[7] and foreign intelligence investigators[8] are remarkably broad.[9] Examples of compelled assistance using these statutes include a secure email provider that was required to place a covert back door in its product in order to steal users' encryption keys [48], and a consumer electronics company that was forced to remotely enable the microphones in a suspect's auto-mobile dashboard GPS navigation unit in order to covertly record their conversations [37].

Outside of the United States, and other democratic countries, specific statutory authority may be even less important. The Chinese government, for example, has repeatedly compelled the assistance of telecommunications and technology companies in assisting it with its surveillance efforts [33, 29].

Just as phone companies and email providers can be forced to assist governments in their surveillance efforts, so too can SSL certificate authorities. The *compelled certificate creation attack* is thus one in which a government agency requires a domestic certificate authority to provide it with false SSL certificates for use in surveillance.

The technical details of this attack are simple, and do not require extensive explanation.[10] Each CA already has an infrastructure in place with which it is able to issue SSL certificates. In this compelled assistance scenario, the CA is merely required to skip the identity verification step in its own SSL certificate issuance process.

For the purposes of our analysis, we assume that a CA cannot refuse to comply with a lawful court order. However, it may be possible, via a *warrant canary* or a similar technique, for a CA to communicate the existence of a secret court order to the Internet community [44]. For example, a representitive from one CA has informed us that his organization's disaster contigency plans include court orders, and that his technical infrastructure includes a "kill switch" that enables him to move to a new physical location, and nullify data at the data center [39]. We do not evaluate the effectiveness of such measures in this paper.

When compelling the assistance of a CA, the government agency can either require the CA to issue it a specific certificate for each website to be spoofed, or, more likely, the CA can be forced to issue a intermediate CA certificate that can then be re-used an infinite number of times by that government agency, without the knowledge or further assistance of the CA.

In one hypothetical example of this attack, the US National Security Agency (NSA) can compel VeriSign to produce a valid certificate for the Commercial Bank of Dubai (whose actual certificate is issued by Etisalat, UAE), that can be used to perform an effective man-in-the-middle attack against users of all modern browsers.

5. SURVEILLANCE APPLIANCES

In October 2009, one of the authors of this paper attended an invitation only conference for the surveillance and lawful interception industry in Washington, DC.[11] Among the many vendor booths on the trade show floor was Packet Forensics, an Arizona based company that sells extremely small, covert surveillance devices for networks.

The marketing materials for the company's 5-series device reveal that it is a 4 square inch "turnkey intercept so-

[7] See: 18 U.S.C. §2518(4).

[8] See: 50 U.S.C. §1805(c)(2)(B).

[9] A thorough survey of the ways in which technology firms can and have been compelled to violate their customers' privacy can be found in [51].

[10] The legal issues relating to this kind of compelled assistance are far more complex. Any US government agencies compelling such CA assistance would almost certainly rely on the assistance provisions highlighted earlier. However, it is unclear if such compelled assistance would be lawful, due to the fact that it would interfere with the CA's ability to provide identity verification services. Such compelled assistance would also raise serious First Amendment concerns, due to to the fact that the government would be ordering the CA to affirmatively lie about the identity of a certificate recepient.

[11] The author caused national headlines in December of 2009, when he released an audio recording of one of the panel discussions at the same conference in which telecommunications company employees bragged about the extent of their cooperation with government agencies, including the extent to which they provide consumers' GPS location information [50, 61].

lution," designed for "defense and (counter) intelligence applications," capable of "packet modification, injection and replay capabilities" at Gb/sec throughput levels. The company proudly boasts that the surveillance device is perfect for the "Internet cafe problem." Most alarming is the device's ability to engage in active man-in-the-middle attacks:

> "Packet Forensics' devices are designed to be inserted-into and removed-from busy networks without causing any noticeable interruption [...] This allows you to conditionally intercept web, e-mail, VoIP and other traffic at-will, even while it remains protected inside an encrypted tunnel on the wire. Using 'man-in-the-middle' to intercept TLS or SSL is essentially an attack against the underlying Diffie-Hellman cryptographic key agreement protocol [...] To use our product in this scenario, [government] users have the ability to *import a copy of any legitimate key they obtain* ***(potentially by court order)*** or they can generate 'look-alike' keys designed to give the subject a false sense of confidence in its authenticity [42]."

The company has essentially packaged software equivalent to *sslstrip* into a 4 square inch appliance, ready for government customers to drop onto networks, at a price that is "so cost effective, they're disposable."

When contacted by a journalist from Wired News in March 2010, Packet Forensics spokesman Ray Saulino initially denied the product performed as advertised in its sales materials, or that anyone used it. But in a follow-up call the next day, Saulino changed his stance, telling the journalist that:

> "The technology we are using in our products has been generally discussed in internet forums and there is nothing special or unique about it [...] Our target community is the law enforcement community [49]."

Furthermore, while Packet Forensics has not disclosed a list of its customers, the firm's website reveals that the 5-series device was authorized for export to foreign firms and governments by the United States Bureau of Industry and Security on July 7, 2009 [41].

6. PROTECTING USERS

The major web browsers are currently vulnerable to the compelled certificate creation attack, and we do not believe that any of the existing privacy enhancing browser add-ons sufficiently protect users without significantly impacting browser usability.

In an effort to significantly reduce the impact of this attack upon end-users, we have created *Certlock*, a lightweight add-on for the Firefox browser. Our solution employs a Trust-On-First-Use (TOFU) policy (this is also known as 'leap-of-faith' authentication) [54, 10], reinforced with a policy that the country of origin for certificate issuing does not change in the future. Specifically, our solution relies upon caching CA information, that is then used to empower users to leverage country-level information in order to make common-sense trust evaluations.

In this section, we will outline the motivations that impacted the design of our solution, discuss our belief in the potential for users to make wise country-level trust decisions, and then explore the technical implementation details of our prototype add-on.

6.1 Design Motivations

The compelled certificate creation attack is a classic example of a low probability, high impact event [12]. The vast majority of users are extremely unlikely to experience it, but for those who do, very bad things are afoot. As such, it is vital that any defensive technique have an extremely low false positive rate, yet be able to get the attention of users when an attempted SSL session hijacking is detected.

Most users are unlikely to know that this threat even exists, and so it is important that any protective system not require configuration, maintenance, nor introduce any noticeable latency to users' connections. Given the low likelihood of falling victim to this attack, most rational users will avoid any protective technology that requires configuration or slows down their Web browsing [25].

Furthermore, to achieve widespread adoption (even moreso if the browser vendors are to add similar functionality to their own products), any protective technology must not sacrifice user privacy for security. Information regarding users' web browsing habits should not be leaked to any third party, even if that party is 'trusted' or if it is done so anonymously. The solution must therefore be self-contained, and capable of protecting the user without contacting any remote servers.

We believe that most consumers are unaware of how SSL functions, what a CA is, the role it performs, and how many companies are trusted by their browser to issue certificates. Expecting consumers to learn about this process, or to spend their time evaluating the business practices and trustworthiness of these hundreds of firms is unreasonable. Nevertheless, the security of the current system requires each user to make trust decisions that that they are ill equipped (nor willing) to perform.

We also believe that consumers do not directly trust CAs. Aside from the biggest CAs such as VeriSign and large telecommunications firms local to their country,[12] it is unlikely that consumers have ever heard of the vast majority of the hundreds of companies entrusted by their web browser to issue certificates. Thus, it is just as unreasonable to expect an American consumer to make a trust decision regarding a certificate issued by Polish technology firm Unizeto Technologies as it is to expect a Japanese consumer to evaluate a certificate issued by Bermuda based QuoVadis. However, both of these CAs are trusted by the major browsers, by default.

Consumers are simply told to look for the lock icon. What happens in the browser to produce that lock icon, is assumed by users to be reliable. We believe that it is our responsibility as security technologists to make sure that what happens behind the scenes does in fact protect the average users' privacy and security.

This is not to say that we think that users are clueless — merely that browsers currently provide them with little to no useful contextual information without which such complex decisions are extremely difficult.

[12] For example, Verizon in the United States, Deutsche Telekom in Germany or Swisscom in Switzerland.

6.2 Country-Based Trust

We believe that many consumers are quite capable of making basic trust decisions based on country-level information. We are not alone in this belief. Since March 2010, Google has been providing country-level warnings to users of its Google Mail service when it detects that their account has been accessed from a potentially suspect IP address in a different country [16].

Thus, a consumer whose banking sessions are normally encrypted by a server presenting a certificates signed by a US based CA might become suspicious if told that her US based bank is now using a certificate signed by a Tunisian, Latvian or Serbian CA.

To make this trust evaluation, she doesn't have to study the detailed business policies of the foreign CA, she can instead rely on common sense, and ask herself why her Iowa based bank is suddenly doing business in Eastern Europe. In order to empower users to make such country-level evaluations of trust, CertLock leverages the wealth of historical browsing data kept by the browser.

Individuals living in countries with laws that protect their privacy from unreasonable invasion have good reason to avoid trusting foreign governments (or foreign companies) to protect their private data. This is because individuals often receive the greatest legal protection from their own governments, and little to none from other countries. For example, US law strictly regulates the ability of the US government to collect information on US persons. However, the government can freely spy on foreigners around the world, as long as the surveillance is performed outside the US. Thus, Canadians, Swedes and Russians located outside the United States have absolutely no reason to trust the US government to protect their privacy.

Likewise, individuals located in countries with oppressive governments may wish to know if their communications with servers located in foreign democracies are suddenly being facilitated by a domestic (or state controlled) CA.

6.3 Avoiding False Positives

A simplistic defensive add-on aimed at protecting users from compelled certificate creation attacks could simply cache all certificates encountered during browsing sessions, and then warn the user any time they encounter a certificate that has changed. In fact, such an add-on, Certificate Patrol, already exists [5].

The problem with such an approach is that it is likely to suffer from an extremely high false positive rate. Each time a website intentionally changes its certificate, the browser displays a warning that will needlessly scare and soon desensitize users. There are many legitimate scenarios where certificates change. For example: Old certificates expire; certificates are abandoned and or revoked after a data breach that exposed the server private key; and many large enterprises that have multiple SSL accelerator appliances serving content for the same domain use a different certificate for each device [31].

By adopting a Trust-On-First-Use policy, we assume that if a website starts using a different certificate issued by the same CA that issued its previous certificate, there is no reason to warn the user. This approach enables us to significantly reduce the false positive rate, while having little impact on our ability to protect users from a variety of threats.

We also believe that there is little reason to warn users if a website switches CAs within the same country. As our threat model is focused on a government adversary with the power to compel any domestic CA into issuing certificates at will, we consider CAs within a country to be equals. That is, a government agency able to compel a new CA into issuing a certificate could just as easily compel the original CA into issuing a new certificate for the same site. Since we have already opted to not warn users in that scenario (described above), there is no need to warn users in the event of a same-country CA change.

By limiting the trigger of the warnings to country-level changes, we believe that we have struck a balance that will work in most situations.

6.4 Implementation Details

Our Certlock solution is currently implemented as an add-on to the Firefox browser.

The Firefox browser already retains history data for all visited websites. We have simply modified the browser to cause it to retain slightly more information. Thus, for each new SSL protected website that the user visits, a Certlock enabled browser also caches the following additional certificate information:

A hash of the certificate.
The country of the issuing CA.
The name of the CA.
The country of the website.
The name of the website.
The entire chain of trust up to the root CA.

When a user re-visits a SSL protected website, Certlock first calculates the hash of the site's certificate and compares it to the stored hash from previous visits. If it hasn't changed, the page is loaded without warning. If the certificate has changed, the CAs that issued the old and new certificates are compared. If the CAs are the same, or from the same country, the page is loaded without any warning. If, on the other hand, the CAs' countries differ, then the user will see a warning (See Figure 1).

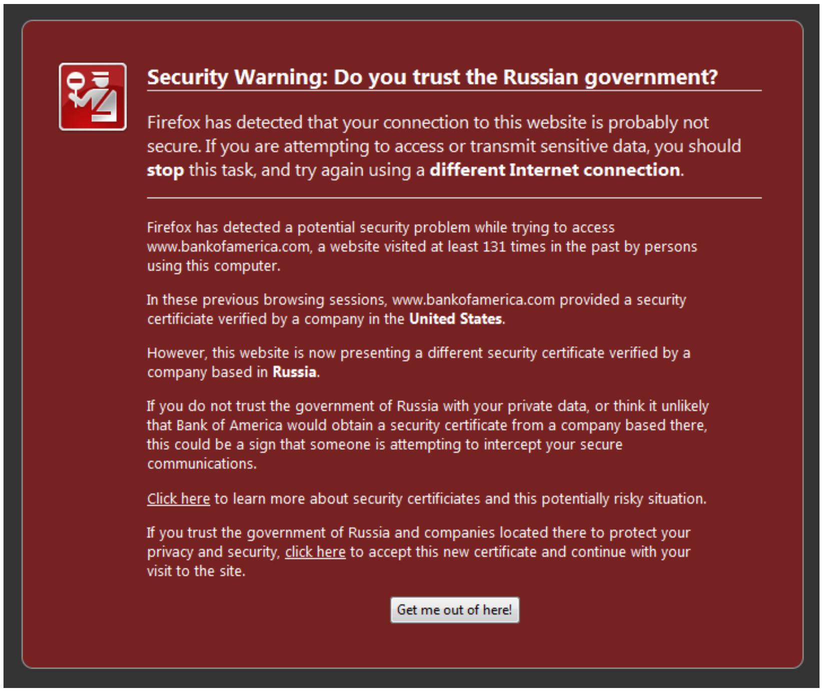
At a high level, this algorithm is quite simple. However, there are a few subtle areas where some complexity is required.

Because governments can compel CAs to create both regular site certificates as well as intermediate CA certificates, any evaluation of a changed site certificate must consider the type of CA that issued it.

While the web browser vendors do not vouch for the trustworthiness of any of the root CAs that they include, we believe it is reasonable to assume that the browser vendors do at least verify the country information listed in each of their root CAs. Therefore, we are able to trust this information as we evaluate changed certificates.

When Certlock detects a changed certificate, it must also determine the type of CA that issued the new certificate. If the new certificate was issued by a root CA, then Certlock can easily compare the country of the old certificate's CA to the country of the new root CA. However, if the new certificate was issued by an intermediate CA, then we have no way of verifying that the issuing CA's country information is accurate.

As an illustrative and hypothetical example of what is currently possible, the Spanish government could compel a Spanish CA to issue an intermediate CA certificate that



Security Warning: Do you trust the Russian government?

Firefox has detected that your connection to this website is probably not secure. If you are attempting to access or transmit sensitive data, you should **stop** this task, and try again using a **different Internet connection**.

Firefox has detected a potential security problem while trying to access www.bankofamerica.com, a website visited at least 131 times in the past by persons using this computer.

In these previous browsing sessions, www.bankofamerica.com provided a security certificiate verified by a company in the **United States**.

However, this website is now presenting a different security certificate verified by a company based in **Russia**.

If you do not trust the government of Russia with your private data, or think it unlikely that Bank of America would obtain a security certificate from a company based there, this could be a sign that someone is attempting to intercept your secure communications.

Click here to learn more about security certificiates and this potentially risky situation.

If you trust the government of Russia and companies located there to protect your privacy and security, click here to accept this new certificate and continue with your visit to the site.

Figure 1: The warning displayed to users of Certlock.

falsely listed the country of the intermediate CA as the United States. This rogue intermediate CA would then be used to issue site certificates for subsequent surveillance activities. In this hypothetical scenario, let us imagine that the rogue CA issued a certificate for Bank Of America, whose actual certificate was issued by VeriSign in the United States. Were CertLock to simply evaluate the issuing CA's country of the previously seen Bank of America certificate, and compare it to the issuing country of the rogue intermediate CA (falsely listed as the United States), CertLock would not detect the hijacking attempt. In order to detect such rogue intermediate CAs, a more thorough comparison must be conducted.

Thus, in the event that a new certificate has been issued by an intermediate CA, Certlock follows the chain of trust up to the root CA, noting the country of every CA along the path. If any one of these intermediate CAs (or the root CA itself) has a different country than the CA that issued the original certificate, then the user is warned.

7. THREAT MODEL ANALYSIS

In this section, we outline several *hypothetical* scenarios in which a man-in-the-middle attack may be desired. In each example scenario, we examine the government's available surveillance options, consider the suitability of the compelled certificate creation attack, and evaluate the ability of CertLock to detect and thwart the attack. A condensed summary of the threats that CertLock defends against is also presented in Figure 2.

7.0.1 Scenario A

Actual CA	VeriSign (USA)
Compelled CA	VeriSign (USA)
Website	Citibank (USA)
Location of Suspect	USA
Spying Government	USA

In this scenario, the United States government compels VeriSign to issue a certificate for use by a law enforcement agency wishing to spy on communications between a suspect located in the United States and Citibank, her United States based bank.

This attack is impossible for CertLock to detect, because the CA issuing the fake certificate is also the same that issued the legitimate certificate. However, we believe that this scenario is extremely unlikely to occur in the investigations of end users. This is because if a government adversary is able to obtain a court order compelling VeriSign's cooperation, it can just as easily obtain a court order compelling Citibank to disclose the suspect's account information.

While there are perhaps a few volunteer run Internet providers that will do anything possible to avoid delivering user data to government agents, we believe that the vast majority of corporations will eventually comply. Outright refusal could potentially result in seizure of corporate assets, and the jailing of executives—consequences that profit focused shareholders would likely wish to avoid. As a related example, in 2006, Google very publicly fought a subpoena from the US Deparment of Justice requesting aggregate search request records. However, once a court ruled on the matter, the company complied and provided the government with 50,000 URLs from the Google search index [59]. As such,

Spying Government	Country of Actual CA	CertLock Protects?
X	X	No
X	Y	Yes

Figure 2: A trust matrix evaluating CertLock. In short, the tool only protects users from compelled certificate creation attacks when the Spying Government and the Country of the Actual CA are not the same.

our threat model specifically excludes the rare category of ISPs willing to say no to government requests at all costs, and instead focuses on typical, law-abiding corporations that provide services to most users.

7.0.2 Scenario B

Actual CA	VeriSign (USA)
Compelled CA	GoDaddy (USA)
Website	Citibank (USA)
Location of Suspect	USA
Spying Government	USA

In this scenario, the United States government compels GoDaddy, a CA located in the United States to issue a certificate for an intelligence agency wishing to spy on communications between a suspect located in the United States and a bank also located in the United States (CitiBank), which obtained its legitimate SSL certificate from VeriSign.

Just as with Scenario A, this attack is extremely unlikely to occur. This is because any government agency able to compel GoDaddy is also capable of obtaining a court order to compel VeriSign or Bank of America. By simple reduction, any attacker capable of Scenario B is also capable of Scenario A. CertLock does not detect attacks of this type.

7.0.3 Scenario C

Actual CA	VeriSign (USA)
Compelled CA	VeriSign (USA)
Website	*Poker.com* (USA)
Location of Suspect	USA
Spying Government	USA

In this scenario, US law enforcement agents are investigating a US-based online gambling website and the US-based users of the service. The agents wish to first obtain evidence that illegal activity is occuring, by monitoring the bets as they are placed via SSL encrypted sessions, before they later raid the offices of the company and seize their servers. In order to surveil the communications between users and the gambling website, law enforcement officials compel VeriSign to issue an additional certificate for the site, which is then used to intercept all communications to and from the website.

In this scenario, where both ends of the SSL connection are under investigation by the government, the compelled certificate attack is a highly effective method for covertly gathering evidence. However, because the issuing CA does not change, CertLock is unable to detect this attack and warn users.

In general, attack scenarios in which both the end-user and the website are under surveillance are beyond the scope of our threat model.

7.0.4 Scenario D

Actual CA	VeriSign (USA)
Compelled CA	TeliaSonera (Finland)
Website	Aktia Bank (Finland)
Location of Suspect	Finland
Spying Government	Finland

In this scenario, a resident of Finland is accessing her Aktia Savings Bank online account, which obtained its legitimate SSL certificate from VeriSign, a US firm. The Finnish intelligence services are interested in getting access to the suspect's online transaction data, and thus seek to compel TeliaSonera, a domestic CA to issue a certificate for the surveillance operation.

This scenario is not identical to scenario A, however it is quite similar. Again, if the Finnish government is able to compel a domestic CA into assisting it, we assume that it could just as easily compel the Finnish bank into providing the suspect's account details. While we believe that this attack scenario is unlikely, should it occur, CertLock will detect it.

7.0.5 Scenario E

Actual CA	VeriSign (USA)
Compelled CA	TeliaSonera (Finland)
Website	Google Mail (USA)
Location of Suspect	Finland
Spying Government	Finland

In this scenario, a US executive is travelling in Finland for business, and is attempting to access her secure, US-based webmail account using the Internet connection in her hotel room. Finnish authorities wish to intercept her communications, but due to Google's use of SSL by default for all webmail communications [46], the government must employ a man-in-the-middle attack. This scenario is thus an ideal candidate for a compelled certificate creation attack, since the Finnish authorities have no leverage to compel the assistance of Google or VeriSign. This scenario is also one that is easily detected by CertLock.

7.0.6 Scenario F

Actual CA	VeriSign (USA)
Compelled CA	VeriSign (USA)
Website	CCB (China)
Location of Suspect	USA
Surveilling Government	USA

In this scenario, a Chinese executive is travelling in the United States for business, and is attempting to acccess her China Construction Bank account using the Internet connection in her hotel room. US Government authorities wish to get access to her financial records, but are unwilling to let the Chinese government know that one of their citizens is under investigation, and so have not requested her records via official law enforcement channels.

This scenario is almost identical to scenario E, however, there is one key difference: The legitimate certificate used by the Chinese bank was issued by a CA located in the United States and the US government has turned to the same US based CA to supply it with a false certificate. Thus, while this scenario is an ideal candidate for a compelled certificate creation attack, it is not one that can easily be detected by looking for country-level CA changes. As such, CertLock is not able to detect attacks of this type.

7.1 Why Sites Should Consider the Country of the CA They Use

Building on the information presented thus far in this paper, we can draw the following conclusions:

- Users are currently vulnerable to compelled certificate creation attacks initiated by the government of any country in which there is at least one certificate authority that is trusted (directly or indirectly) by the browser vendors.
- When users provide their private data to a company, the government of the country in which their data is located may be able to compel the provider to disclose their private data.
- When users provide their private data to a company that holds the data in country X, but uses a SSL certificate provided by a CA in country Y, users are vulnerable to both the compelled disclosure of their data by the government of country X, and interception of their private data through a compelled certificate creation attack by country Y.
- Thus, when a company that uses a certificate authority located in a country different than the one in which it holds user data, it needlessly exposes users' data to the compelled disclosure by an additional government.

It is based on this that we believe that websites best serve their users when they rely on a SSL certificate from a CA located in the same country in which their private data is stored.[13] Unfortunately, this is not a widespread practice in the industry; instead American CAs totally dominate the certificate market, and are used by many foreign organizations.

As just one example — a number of the big banks in Pakistan, Lebanon and Saudi Arabia (countries in which the US has a strong intelligence interest) all use certificates obtained from US-based CAs to secure their online banking sites.

It is because of the dominance of US CAs that CertLock is not able to equally protect users from different countries. Certlock can effectively protect users of US based services from compelled certificate disclosure attacks performed by non-US governments. Thus, it is useful for Americans travelling out of the country who may be subject to surveillance by the national government of the country in which they are travelling, and non-US persons who use US-based services and who do not wish for their own governments to get access to their data.

However, as long as companies around the world continue to rely on SSL certificates issued by American CAs, the US government will maintain the ability to perform man in the middle attacks that are practically impossible to detect with CertLock or any other country based detection mechanism.

8. RELATED WORK

Over the past decade, many people in the security community have commented on the state of the SSL public key infrastructure, and the significant trust placed in the CAs [20, 11, 19].

In 1998, James Hayes of the US National Security Agency published a paper that focused specifically on the threat of rogue insiders within a Certificiate Authority. Although the technical details of the threat outlined by Hayes are largely the same as the scenario on which we have focused (albeit with vasty different legal and policy consequences), Hayes did not address the threat of government compelled certificate creation. It is unclear if he was simply unaware of this scenario, or if the topic was too sensitive for him to discuss, given his employer. In his paper, Hayes proposed a technical solution to address the insider threat, which relied on users configuring various per-site attributes within their browser that would be used to evaluate each new site's certificate.

Crispo and Lomas also proposed a certification scheme designed to detect rogue CAs [14], while the Monkeysphere project has created a system that replaces the CA architecture with the OpenPGP web of trust [6].

Ian Grigg has repeatedly sought to draw attention to both the potential conflict of interest that some CAs have due to their involvement in other forms of surveillance, and the power of a court order to further compel these entities to assist government investigations [21, 22, 23]. In particular, in 2005, Grigg and Shostack filed a formal complaint with ICANN over the proposal to award VeriSign control of .net domain name registration, arguing that the firm's surveillance products created a conflict of interest [24].

In recent years, several browser-based tools have been created to help protect users against SSL related attacks. Kai Engert created Conspiracy, a Firefox add-on that provides country-level CA information to end-users in order to protect them from compelled certificate creation attacks. The Conspiracy tool displays the flag of the country of each CA in the chain of trust in the browser's status bar [17]. Thus, users must themselves remember the country of the CAs that issue each certificate, and detect when the countries have changed. We believe, like Herley [25], that this is an unreasonable burden to place upon end-users, considering how rarely the compelled certificate creation attack is likely to occur.

Wendlandt *et al.* created Perspectives, a Firefox add-on that improves the Trust-On-First-Use model used for websites that supply self-signed SSL certificates [58]. In their system, the user's browser securely contacts one of several notary servers, who in turn independently contact the webserver and obtain its certificate. In the event that an attacker is attempting to perform a man in the middle attack upon the user, the fact that the attacker-supplied SSL certificate, and those supplied by the Perspectives notary

[13] For example, all of the Hungarian banks surveyed by the authors use certificates provided by NetLock Ltd., a Hungarian CA.

servers differ will be a strong indicator that something bad has happened.

Unfortunately, the Perspectives system requires that users provide the Perspectives notaries with a real-time list of the secure sites they visit.[14] Although the scheme's designers state that "all servers adhere to a strict policy of never recording client IP addresses, period," we still don't think it is a good idea to provide users' private web browsing data to a third party, merely based on the fact that they promise not to log it.

Alicherry and Keromytis have improved upon the Perspectives design with their DoubleCheck system [8], substituting Tor exit nodes for special notary servers. Because the Tor network anonymizes the individual user's IP address, there is no way for the Tor exit nodes to know who is requesting the certificate for a particular SSL website. While the authors solved the major privacy issues that plague the Perspectives scheme, their choice of Tor carries its own cost: Latency. Their system adds an additional second of latency to every new SSL connection, and up to 15 seconds for visits to new self-signed servers. We believe that this additional latency is too much to ask most users to bear, particularly if the chance of them encountering a rogue CA is so low.

Herzberg and Jbara created TrustBar, a Firefox add-on designed to help users detect spoofed websites. The browser tool works by prominently displaying the name of the CA that provided the site's certificate, as well as allowing the user to assign a per-site name or logo, to be displayed when they revisit to each site [26].

Tyler Close created Petname Tool, a Firefox add-on that caches SSL certificates, and allows users to assign a per-site phrase that is displayed each time they revisit the site in the future. In the event that a user visits a spoofed website, or a site with the same URL that presents a certificate from a different CA, the user's specified phrase will not be displayed [13].

In May 2008, a security researcher discovered that the OpenSSL library used by several popular Linux distributions was generating weak cryptographic keys. While the two-year old flaw was soon fixed, SSL certificates created on computers running the flawed code were themselves open to attack [7, 60]. Responding to this flaw, German technology magazine Heise released the Heise SSL Guardian for the Windows operating system, which warns users of Internet Explorer and Chrome when they encounter a weak SSL certificate [57].

In December 2008, Stevens *et al.* demonstrated that flaws in the MD5 algorithm could be used to create rogue SSL certificates (without the knowledge or assistance of the CA). In response, CAs soon accelerated their planned transition to certificates using the SHA family of hash functions [55]. As an additional protective measure, Márton Anka developed an add-on for the Firefox browser to detect and warn users about certificate chains that use the MD5 algorithm for RSA signatures [9].

Jackson and Barth devised the ForceHTTPS system to protect users who visit HTTPS protected websites, but who are vulnerable to man in the middle attacks due to the fact that they do not type in the `https://` component of the URL [28]. This system has since been formalized into the Strict Transport Security (STS) standard proposal [27], to which multiple browsers are in the process of adding support. While this system is designed to enable a website to hint to the browser that future visits should always occur via a HTTPS connection, this mechanism could be extended to enable a website to lock a website to a particular CA, or CAs of a specific country.

9. CONCLUSION AND FUTURE WORK

In this paper, we introduced the compelled certificate creation attack and presented evidence that suggests that governments may be subverting the CA based public key infrastructure. In an effort to protect users from these powerful adversaries, we introduced a lightweight defensive browser based add-on that detects and thwarts such attacks. Finally, we use reductive analysis of governments' legal capabilities to perform an adversarial threat model analysis of the attack and our proposed defensive technology.

Our browser add-on is currently just a prototype, and we plan to improve it in the future. First, our currently used warning dialog text is far from ideal, and could be greatly improved with the help of usability and user experience experts. We also plan to explore the possibility of expanding the country-level trust model to regions, such as the European Union, where, for example, residents of France may be willing to trust Spanish CAs. Finally, We are considering adding a feature that will enable users to voluntarily submit potentially suspect certificates to a central server, so that they can be studied by experts. Such a feature, as long as it is opt-in, does not collect any identifiable data on the user, and only occurs when potentially rogue certificates are discovered, would have few if any privacy issues.

Ultimately, the threats posed by the compelled certificate creation attack cannot be completely eliminated via our simple browser add-on. The CA system is fundamentally broken, and must be overhauled. DNSSEC may play a significant role in solving this problem, or at least reducing the number of entities who can be compelled to violate users' trust. No matter what system eventually replaces the current one, the security community must consider compelled government assistance as a realistic threat, and ensure that any solution be resistant to such attacks.

10. ACKNOWLEDGEMENTS

Thanks to Kevin Bankston, Matt Blaze, Jon Callas, Allan Friedman, Jennifer Granick, Markus Jakobsson, Dan Kaminsky, Moxie Marlinspike, Eddy Nigg, Eli O and Adam Shostack for their useful feedback.

11. REFERENCES

[1] Apple Root Certificate Program. `www.apple.com/certificateauthority/ca_program.html`.

[2] Mozilla CA Certificate Policy (Version 1.2). `www.mozilla.org/projects/security/certs/policy/`.

[3] Microsoft Root Certificate Program, January 15 2009. `technet.microsoft.com/en-us/library/cc751157.aspx`.

[4] Windows Root Certificate Program Members, November 24 2009. `download.microsoft.com/`

[14] Modern browsers already leak information about the secure web sites that users visit, as they automatically contact CAs in order to verify that the certificates have not been revoked (using the OCSP protocol). While this is currently unavoidable, we wish to avoid providing private user web browsing data to any additional parties.

download/1/4/f/14f7067b-69d3-473a-ba5e-70d04aea5929/windows\%20root\%20certificate\%20program\%20members.pdf.

[5] Certificate patrol, 2010. patrol.psyced.org/.

[6] Monkeysphere, 2010. web.monkeysphere.info/.

[7] D. Ahmad. Two Years of Broken Crypto: Debian's Dress Rehearsal for a Global PKI Compromise. *IEEE Security and Privacy*, 6:70–73, September 2008.

[8] M. Alicherry and A. D. Keromytis. Doublecheck: Multi-path verification against man-in-the-middle attacks. In *ISCC 2009: IEEE Symposium on Computers and Communications*, pages 557–563, Piscataway, NJ, USA, 2009. IEEE.

[9] M. Anka. SSL Blacklist 4.0, January 31 2010. www.codefromthe70s.org/sslblacklist.aspx.

[10] J. Arkko and P. Nikander. Weak authentication: How to authenticate unknown principals without trusted parties. In B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, editors, *Security Protocols Workshop*, volume 2845 of *Lecture Notes in Computer Science*, pages 5–19. Springer, 2002.

[11] P. S. Bance. Ssl: Whom do you trust?, April 20 2005. www.minstrel.org.uk/papers/2005.04.20-ssl-trust.pdf.

[12] M. Bussiere and M. Fratzscher. Low probability, high impact: Policy making and extreme events. *Journal of Policy Modeling*, 30(1):111–121, 2008.

[13] T. Close. Petname tool, 2005. www.waterken.com/user/PetnameTool/.

[14] B. Crispo and M. Lomas. A certification scheme for electronic commerce. In *In Security Protocols International Workshop*, page pages. Springer-Verlag, 1996.

[15] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246 (Proposed Standard), Jan. 1999. Obsoleted by RFC 4346, updated by RFCs 3546, 5746.

[16] P. Diwanji. Detecting suspicious account activity. *The Official Gmail Blog*, March 24 2010. gmailblog.blogspot.com/2010/03/detecting-suspicious-account-activity.html.

[17] K. Engert. Conspiracy — A Mozilla Firefox Extension, March 18 2010. kuix.de/conspiracy/.

[18] E. Felten. Web Certification Fail: Bad Assumptions Lead to Bad Technology. *Freedom To Tinker*, February 23 2010. www.freedom-to-tinker.com/blog/felten/web-certification-fail-bad-assumptions-lead-bad-technology.

[19] E. Gerck. First published online by the MCWG at http://mcwg.org/cert.htm (April, 1997). Invited talk at the Black Hat Briefings '99, Las Vegas, NV, July 7-8, 1999. Published by The Bell, ISSN 1530-048X, Vol. 1, No. 3, p. 8, July 2000, and available online at http://www.thebell.net/papers/certover.pdf.

[20] D. K. Gillmor. Technical Architecture shapes Social Structure: an example from the real world, February 21 2007. lair.fifthhorseman.net/~dkg/tls-centralization/.

[21] I. Grigg. VeriSign's conflict of interest creates new threat. *Financial Cryptography (blog)*, September 1 2004. financialcryptography.com/mt/archives/000206.html.

[22] I. Grigg. PKI considered harmful. October 14 2008. iang.org/ssl/pki_considered_harmful.html.

[23] I. Grigg. Why the browsers must change their old SSL security (?) model. *Financial Cryptography (blog)*, March 24 2010. financialcryptography.com/mt/archives/001232.html.

[24] I. Grigg and A. Shostack. VeriSign and Conflicts of Interest, February 2 2005. forum.icann.org/lists/net-rfp-verisign/msg00008.html.

[25] C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *NSPW '09: Proceedings of the 2009 workshop on New security paradigms workshop*, pages 133–144, September 2009.

[26] A. Herzberg and A. Jbara. Security and identification indicators for browsers against spoofing and phishing attacks. *ACM Trans. Internet Technol.*, 8(4):1–36, 2008.

[27] J. Hodges, C. Jackson, and A. Barth. Strict Transport Security, December 18 2009. lists.w3.org/Archives/Public/www-archive/2009Dec/att-0048/draft-hodges-strict-transport-sec-06.plain.html.

[28] C. Jackson and A. Barth. Forcehttps: protecting high-security web sites from network attacks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 525–534, New York, NY, USA, 2008. ACM.

[29] A. Jacobs. China requires censorship software on new pcs. *The New York Times*, June 8 2009. www.nytimes.com/2009/06/09/world/asia/09china.html.

[30] D. Kaminsky. Black Ops of PKI. *26th Chaos Communication Congress*, December 29 2009. events.ccc.de/congress/2009/Fahrplan/events/3658.en.html.

[31] D. Kaminsky. Email conversation with author, February 28 2010.

[32] D. Kaminsky, M. L. Patterson, and L. Sassaman. PKI Layer Cake: New Collision Attacks Against the Global X.509 Infrastructure. In *Proceedings of Financial Cryptography and Data Security - 14th International Conference (FC 2010)*, 2010.

[33] J. Markoff. Surveillance of skype messages found in china. *The New York Times*, October 1 2008. www.nytimes.com/2008/10/02/technology/internet/02skype.html.

[34] M. Marlinspike. More Tricks For Defeating SSL In Practice. *BlackHat USA*, 2009. www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatSSL-SLIDES.pdf.

[35] M. Marlinspike. sslsniff, July 3 2009. www.thoughtcrime.org/software/sslsniff/.

[36] M. Marlinspike. sslsniff, December 18 2009. www.thoughtcrime.org/software/sslstrip/.

[37] D. McCullagh. Court to FBI: No spying on in-car computers. *CNET News*, November 19 2003. news.cnet.com/2100-1029_3-5109435.html.

[38] Mozilla. Potentially problematic CA practices , 2010. wiki.mozilla.org/CA:Problematic_Practices.

[39] E. Nigg. Email conversation with author, March 27 2010.

[40] J. Nightingale. SSL Question Corner. *meandering wildly (blog)*, August 5 2008. blog.johnath.com/2008/08/05/ssl-question-corner/.

[41] Packet Forensics. Export and Re-Export Requirements, 2009. www.packetforensics.com/export.safe.

[42] Packet Forensics. Marketing materials for 5-series products, 2009. http://files.cloudprivacy.net/packet-forensics-materials.pdf.

[43] M. Ray and S. Dispensa. Renegotiating TLS, November 4 2009. extendedsubset.com/wp-uploads/2009/11/renegotiating_tls_20091104_pub.zip.

[44] rsync.net. rsync.net warrant canary, 2010. http://www.rsync.net/resources/notices/canary.txt.

[45] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer. The emperor's new security indicators. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 51–65, Washington, DC, USA, 2007. IEEE Computer Society.

[46] S. Schillace. Default https access for Gmail. *The Official Gmail Blog*, January 12 2010. gmailblog.blogspot.com/2010/01/default-https-access-for-gmail.html.

[47] A. Shamir. Cryptography: State of the science. *ACM A. M. Turing Award Lecture*, June 8 2003. awards.acm.org/images/awards/140/vstream/2002/S/s-pp/shamir_1files_files/800x600/Slide8.html.

[48] R. Singel. PGP Creator Defends Hushmail. *Wired News Threat Level Blog*, November 19 2007. www.wired.com/threatlevel/2007/11/pgp-creator-def.

[49] R. Singel. Law Enforcement Appliance Subverts SSL. *Wired News Threat Level Blog*, March 24 2010. www.wired.com/threatlevel/2010/03/packet-forensics/.

[50] C. Soghoian. 8 Million Reasons for Real Surveillance Oversight. *Slight Paranoia blog*, December 1 2009. paranoia.dubfire.net/2009/12/8-million-reasons-for-real-surveillance.html.

[51] C. Soghoian. Caught in the cloud: Privacy, encryption, and government back doors in the web 2.0 era. In *Journal on Telecommunications and High Technology Law*, Forthcoming.

[52] A. Sotirov and M. Zusman. Breaking the Security Myths of Extended Validation SSL Certificates. *BlackHat USA*, 2009. www.blackhat.com/presentations/bh-usa-09/SOTIROV/BHUSA09-Sotirov-AttackExtSSL-SLIDES.pdf.

[53] C. Spiezle. Email conversation with author, February 15 2010.

[54] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, London, UK, 2000. Springer-Verlag.

[55] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. Weger. Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In *Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology*, pages 55–69, Berlin, Heidelberg, 2009. Springer-Verlag.

[56] J. Sunshine, S. Egelman, H. Almuhimedi, N. Atri, and L. F. Cranor. Crying wolf: An empirical study of SSL warning effectiveness. In *Proceedings of the 18th Usenix Security Symposium*, August 2009.

[57] The H Security. heise SSL Guardian: Protection against unsafe SSL certificates, July 4 2008. www.h-online.com/security/features/Heise-SSL-Guardian-746213.html.

[58] D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: improving ssh-style host authentication with multi-path probing. In *ATC'08: USENIX 2008 Annual Technical Conference on Annual Technical Conference*, pages 321–334, Berkeley, CA, USA, 2008. USENIX Association.

[59] N. Wong. Judge tells DOJ "No" on search queries. *The Official Google Blog*, March 17 2006. googleblog.blogspot.com/2006/03/judge-tells-doj-no-on-search-queries.htmll.

[60] S. Yilek, E. Rescorla, H. Shacham, B. Enright, and S. Savage. When private keys are public: results from the 2008 Debian OpenSSL vulnerability. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 15–27, New York, NY, USA, 2009. ACM.

[61] K. Zetter. Feds 'Pinged' Sprint GPS Data 8 Million Times Over a Year. *Wired News Threat Level Blog*, December 1 2009. www.wired.com/threatlevel/2009/12/gps-data/.

Private Sharing of User Location over Online Social Networks

Julien Freudiger, Raoul Neu, and Jean-Pierre Hubaux

School of Computer and Communication Sciences, EPFL, Switzerland
`firstname.lastname@epfl.ch`

Abstract. Online social networks increasingly allow mobile users to share their location with their friends. Much to the detriment of users' privacy, this also means that social network operators collect users' location. Similarly, third parties can learn users' location from localization and location visualization services. Ideally, third-parties should not be given complete access to users' location. To protect location privacy, we design and implement a platform-independent solution for users to share their location in a private fashion over online social networks. Our solution relies on encryption to enforce access control and uses dummy queries and caching to protect localization and location visualization.

1 Introduction

Cellphone users increasingly share their location on the Internet with location-based services (LBSs) or online social networks (OSNs). These third-parties use location information to infer users' context and provide customized services [21]. We focus on *location-sharing services* (LSSs) that enable friends to share their locations with each other [18,27,38,39]: such services combine location-based services with online social networks.

While accessing LSSs, users often rely on two other components: a *localization* component, with which users obtain their location [4,12] and a *visualization* component, with which users render their location on a map and that of their friends [1,5,17,41].

Much to the detriment of users' privacy, third-parties running location-sharing services can collect users' location. Similarly, some localization services locate users based on wireless access points in users' proximity and thus learn users' location. Also, users often visualize their location and that of their friends on *online* maps thus revealing locations to the map operator. Because visited locations are correlated to users' identity, third-parties can not only *localize* users but also *profile* them over time [29,34,36,37]. Online social networks already know users' identity and their social graph, but with location information, they obtain a deeper insight into users' activities.

Ideally, location-sharing services should not be given complete access to users' location. Previous work on location-based services proposed to disclose minimum information relying on obfuscation and anonymization [20,30,32,33,34,35,42,44].

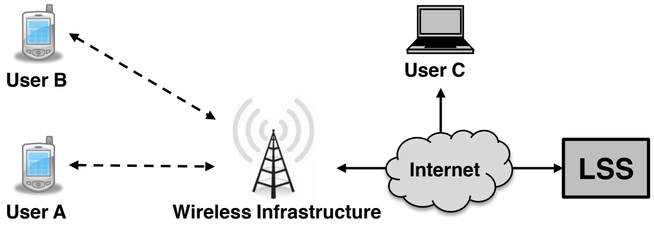


Fig. 1. System model. Users upload their location to a central server that provides location-sharing services. Users A, B and C can then see each others location.

These solutions aim at reducing the information shared with LBSs while still proposing a minimum quality of service. In the case of LSSs, the requirements differ: users aim at sharing their location with friends over third-parties that do not provide specific services but only facilitate users' interactions. Thus, obfuscation and anonymization are not viable options.

In this work, we design a platform-independent solution for users to share their location in a private fashion over LSSs. Our solution does not require any changes to the infrastructure and can operate with any third-party LSS. We implement our solution as a prototype application for mobile phones codenamed PrivL (Private Locations). It uses encryption to protect users' location from third-party servers while still allowing them to share it with friends. It can also make use of distributed Hash tables (DHTs) to provide ephemeral data storage [28]. We also discuss the problem of protecting privacy with respect to localization and map visualization services and propose a solution based on dummy queries and caching. We discuss the challenge of generating dummy queries to confuse an adversary and use the concept of *virtual identities*.

2 Models

2.1 System Model

We study a network (Fig. 1) that involves mobile users equipped with wireless devices, third-party operators running location-sharing services and a wireless infrastructure. Let us define $\mathcal{U} = \{U_1, U_2, ..., U_n\}$ as the set of users in the network, where n is the total number of users. For simplicity, we consider that each user owns a single communication device. We study a discrete time system with initial time $t = 0$ and consider the location of users at each discrete time t.

Wireless devices feature localization technology such as GPS, or wireless triangulation that lets users locate themselves [4,12]. The geographic *location* of a user is denoted by $l = (lon, lat)$, where lon is the longitude, and lat is the latitude. The wireless infrastructure relies on technology such as WiFi, GSM or 3G to let users connect to the Internet. LSSs are operated by independent third-party entities that intend to provide services based on users' location.

Cellphone users send their location to an LSS operator through the wireless infrastructure. For each request sent, users may have to identify themselves to the LSS using proper credentials. For example, some services may require users to have an account, and to provide the corresponding username and password with each request, whereas other services may be open to everyone and use HTTP cookies to recognize several visits from the same user. In general, we assume that users are identified with *pseudonyms* (i.e., fictitious identifiers), such as their username, their HTTP cookie or their IP address. In this work, we focus on LSSs enabling users to share their location with friends such as Google Latitude [38], FireEagle [26], and Loopt [39]. We define a function $F(U_i)$ that provides the set of friends of every user U_i.

Each location sample shared by a user is called an *event*. Each event is denoted by a triplet $< i, t, l >$, where i is the pseudonym of a user, t is the time instance at which the event occurred, and l is the location of the event. We consider a simple communication model in which users upload their location to LSSs with a post request:

1. U: Compute location l
2. $U \rightarrow LSS$: Post $< i, t, l >$
3. $LSS \rightarrow U$: Ack

Users can obtain the location of their friends with a get request:

1. $U \rightarrow LSS$: Get location of $F(U)$
2. $LSS \rightarrow U$: $< j, t, l >, \forall j \in F(U)$
3. U: Visualize on online map

Users then visualize their location and that of their friends on an online [1,5,17] or offline [41] map based on event triplets.

2.2 Threat Model

In this work, we investigate the ability of third-parties to learn users' location. We consider the potential privacy threat of each component of LSS applications. In the localization component, a third-party operator running the localization service learns users' location [13]. In the sharing component, LSS operators passively collect information about the locations of pseudonymous users over time. In the visualization component, a third-party operator running an online map learns users' location.

We consider that the adversary passively collects data from users and may be active or retro-active [28]: the adversary can interact with users to obtain their *current* location or interact with the servers storing the location data to obtain their *past* location. The adversary knows the probability that users visit specific locations on a map based on statistical information obtain from various sources such as Citysense [2], geographic information systems [23], or the Census bureau [29]. The adversary may also host a social network (e.g. Loopt [39]) and thus knows users' social graph and identity.

3 Our Solution

We rely on well-known privacy-preserving mechanisms to protect the privacy of LSS users: we use broadcast encryption to share users' location over third-parties and dummy queries with caching to obtain and visualize users location. Our solution does not require any changes from the infrastructure, and can operate with any LSS. We implement these mechanisms in a prototype application for mobile phones.

3.1 Privacy-Preserving Location Sharing

We propose to use encryption mechanisms to protect data privacy. We make use of cellphones communication capabilities to establish the keying material. We also suggest the use of ephemeral storage to thwart retroactive adversaries.

Encryption Mechanisms Cryptographic mechanisms can limit the access of LSSs to location information. By encrypting the location information, only qualified users (e.g., friends) can decrypt it: a user U_i can share his location only with his friends $F(U_i)$. To do so, we use *Broadcast encryption* schemes [25]. The challenge of broadcast encryption is to limit the size of the communication packet, of the encryption overhead, the number of keys and to allow for the revocation of users. Several solutions exist offering various tradeoffs among these constraints.

Symmetric Scheme The simplest broadcast encryption scheme consists in using symmetric cryptography: U_i establishes a pair-wise secret with each of his friends. User U_i then posts its location encrypted with the pair-wise secrets.

Still, this approach does not scale well: the number of key establishments, the packet size and the computation overhead increase linearly with the number of friends.

Trivial Asymmetric Scheme Assume that each user owns a public and private key pair. A simple broadcast encryption scheme based on asymmetric cryptography consists in using *hybrid encryption*: location information is encrypted with a secret and the secret is encrypted with the public keys of each user.

This scheme simplifies the establishment and management of secrets for groups of users. Still, as before, it generates packet sizes and computation overhead linear in the number of friends.

Dynamic Scheme Recently, advanced schemes were proposed achieving shorter cipher texts and key length [22] and offering to dynamically add new users to the set of friends at a low cost [24]. Nevertheless, such schemes still lack efficient implementations and are thus not easy to deploy in practice.

In LSSs, we believe that users will most of the time share their location with a subset of all their friends. Consequently there will be a small number of destinations, and as shown in [24], the trivial asymmetric scheme and the symmetric scheme are preferable in such case. Hence, our solution makes use of the symmetric scheme and the trivial asymmetric scheme implemented with OpenSSL.

Key Establishment Secret keys can be retrieved from a central repository or directly exchanged between users when they are in physical proximity using peer-to-peer wireless communications.

In this work, we consider that users communicate their keys in a peer-to-peer fashion using Bluetooth, text messages (SMSs), or phone calls. In other words, we rely on the existing cellular infrastructure or physical proximity to guarantee the authenticity of the keying material.

Distributed Hash Tables (DHTs) The encrypted location data can be stored directly on the third-party LSS. However, a retro-active adversary may try to obtain the location data from the LSS in the future. As the data is stored encrypted, the retro-active adversary will also have to obtain the secret used for encryption to obtain users' location.

The encrypted location data can alternatively be stored on an ephemeral medium such as a DHT. Distributed Hash Tables are a peer-to-peer (P2P) storage network consisting of multiple participating nodes. They offer a natural time to live (TTL) of data as peers enter and exit the DHT over time. DHTs have two advantages: the encrypted location information cannot be obtained by a retro-active adversary [28] and if an LSS refuses to host encrypted material, we can upload on the LSS links to a DHT. In other words, DHTs offer a temporary storage solution in which the LSS becomes a transparent server that maintains the relation with friends but does not contain any location information.

In our solution, users can choose to upload their location either on a DHT or on the LSS directly.

3.2 Privacy-Preserving Localization

Most cell phones are equipped with GPS and can locate themselves privately. Yet, GPSs are battery consuming, work poorly indoors and are slow to obtain users' location. Hence, mobile devices sometimes rely on wireless signal triangulation for localization: users reveal contextual information such as the list of WiFi access points in proximity to a localization server that computes users' location [4,12].

A mobile device can protect its location privacy with respect to localization servers by altering localization requests in two ways: i) modify the list of WiFi access points in proximity or ii) send multiple queries in parallel. The first solution will degrade the localization precision. The second solution can be implemented using dummy localization requests and caching.

Caching In order to avoid contacting localization servers altogether, users can cache all access points locations in predefined regions of interest. We use publicly available sources of information to create such caches. In practice, users tend to have repeated mobility patterns, and caching should thus significantly reduce the number of queries sent to localization servers.

However, caching may not scale well as mobile devices may not have enough memory to store all the required information. Similarly, the cache may not contain localization information for some regions because the publicly available source does not contain localization information about that region. In this case, dummy requests could be used.

Dummy Queries Following the principle of k-anonymity, users can create $k-1$ queries to the localization server. This way, the server will be uncertain as to where exactly the user is. To create queries, users can rely on public databases of existing access points [16]. The adversary may still be able to infer the real location of users out of the k queries using statistical inference attacks. We distinguish between two scenarios: users can be *traceable* or *untraceable.*

Traceable Virtual Identities Users are often traceable by third-party providers based on their IP address. The IP address tends to remain the same on data connections of mobile networks (3G and GSM). HTTP cookies can also be used by third-parties to link multiple interactions with the same user.

By linking multiple user requests, the localization service can statistically obtain users' real location out of the k possible locations by analyzing queries over time. For example, if $k-1$ locations of dummy queries are chosen at random, the adversary can estimate that they do not correspond to real *human-generated* queries based on the probability of visiting certain locations.

To avoid such attacks, users can choose $k-1$ queries in a strategic fashion. For example, users can select frequently visited locations. Nevertheless, the adversary may still statistically determine if locations in queries correspond to *human mobility* by looking at the mobility patterns of visited locations. To avoid such inference, users have to maintain a set of *virtual* identities. Each virtual identity will have its own querying profile, matching human mobility statistics.

In summary, virtual identities must respect a number of constraints. *Spatially*, the locations visited by virtual identities should correspond to realistic human locations. *Temporally*, the distance between two consecutive locations should be feasible according to the average speed of humans. Finally, the mobility of virtual identities should *statistically* match traditional mobility statistics.

Untraceable There are multiple techniques to reduce user traceability. Users can rely on DHCP to force IP address changes, or use Tor to obtain a larger IP addresses anonymity set. Similarly, users can erase their HTTP cookies.

If users are not traceable, the localization server must do a double inference. It must first estimate the relation between multiple user requests over time and then statistically derive users' possible locations.

3.3 Privacy-Preserving Visualization

In the case of online maps, the visualization of location information reveals users' location and that of his friends to the map operator. We suggest to make use of dummy map requests and local map caching to protect location privacy.

A user U_i must render the map of his own location and also render the map corresponding to location of his friends: $< j, t, l > \forall U_j \in F(U_i)$. We call m_j the map request to the online map provider containing the *lat*, *lon* coordinates of U_j's location, a radius and a zoom level.

Caching Again caching can be used to reduce the number of queries sent to the map server. A priori, users can define regions of interest where map caching is done. The size of the maps being quite large, this solution may also have problems to scale.

Dummy Queries Inspired by the k-anonymity principle, U_i can use dummy m's to confuse the map operator. In this setting, dummy queries must protect the location privacy of user U_i *and* that of U_i's friends as well.

Like in the localization case, the adversary can use statistical information to infer the real location of a user out of the k queries. In the map visualization scenario, the adversary can even correlate the mobility of a user with that of his friends.

Social Virtual Identities To avoid such attacks, users can maintain sets of virtual identities for them and their friends that have a *social life*. Indeed, the *interaction* between virtual identities must also be modeled to appear as a realistic profile for the adversary.

In our prototype application, we implement a dummy request algorithm that chooses $k - 1$ other queries from a set of frequently visited locations. We also cache every query made by the user locally. We are currently investigating the use of virtual identities in our application.

3.4 Implementation

In order to test the feasibility of the proposed privacy-preserving mechanisms, we implemented a prototype client running on mobile phones code-named PrivL.[1]

We have chosen the Symbian platform because it is a popular platform for mobile phones and it supports QT [10], a cross-platform application framework (Fig. 2). We show in Fig. 3 (a) & (b) two screenshots of the resulting application. Note that in the settings, users can establish security associations, connect to LSSs, give a preference on the positioning mode and privacy level.

[1] The source code will be open sourced and available at privl.sourceforge.net .

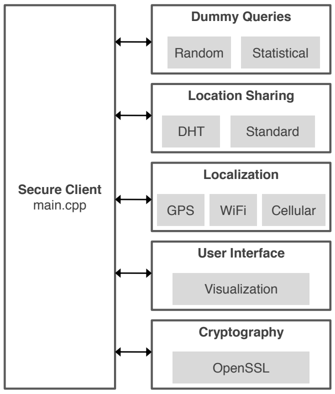


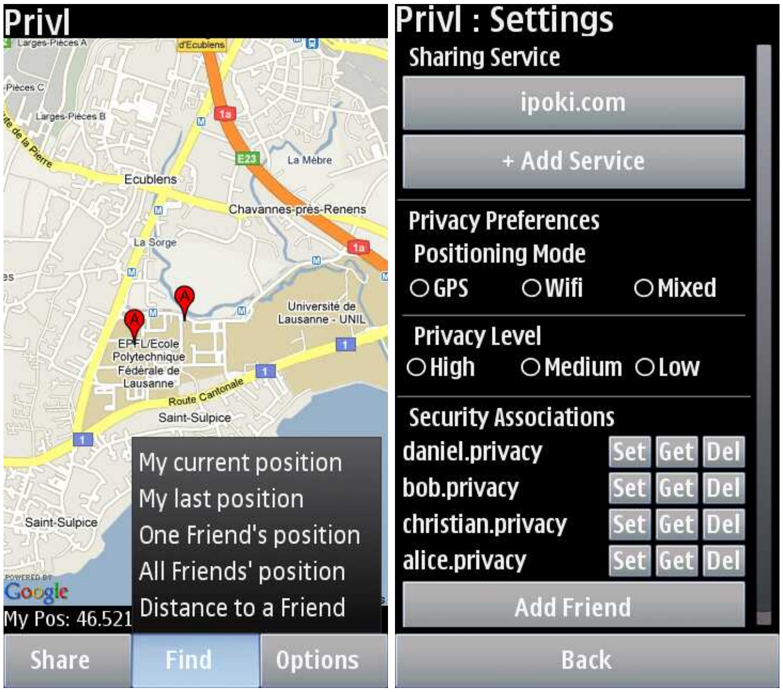
Fig. 2. Implementation overview.

Architecture The current version of the application uploads users' location on any third-party LSS. We have currently implemented an interface for ipoki [6] because it provides an open and detailed API. Users must first login in order get their Friend list and to post their location. We provide users with a new feature which allows them sharing their encrypted location: they upload the ciphertext to a DHT and a reference to it on ipoki's servers. Users have to generate random pair-wise AES session keys and share them with friends using peer-to-peer communications. Cryptographic operations as well as dummy queries are completely transparent to users and do not affect the performances of the LSS application. Our prototype was successfully tested on multiple standard mobile devices.

Technical Details To obtain users' current position, we send nearby Cell Tower information and the MAC addresses of nearby WiFi access points to the Google Geolocation API [4]. While WiFi scanning is implemented with the SDK API plug-in from Symbian C++ [15], the Cell Tower information is provided by the QT Mobility API. The latter API also provides GPS support.

The localization and the visualization component require both caching of data. While this is achieved in the first case by downloading all available access points from Wigle.net within a user defined region, map caching is done by activating the browsers native cache. These two components also make use of dummy queries to enhance privacy. To make sure that dummy positions are valid, we use locations from the list of WiFi access point of Wigle.net. Users' position is shown on a map in a Webkit based browser. The map data is retrieved from Google's static map API [14] and we position placemarks with Javascript.

To provide ephemeral storage, we currently use the academic P2P Network OpenDHT [8]. The interaction with this network is done according to the XML-RPC protocol. The encryption of the user's position relies on the AES Symmetric



(a) (b)

Fig. 3. Interface overview. (a) Main window showing users' location with markers. (b) Settings window enabling users to establish security associations, to connect to LSSs and to setup their privacy requirements.

encryption algorithms which are provided by the EVP module of OpenSSL from Nokia's OpenC plugin.

The Graphical User Interface is implemented with QT's GUI framework: it is optimized for Symbian 5th Edition devices with touch screen. We rely also on the QT Mobility API for the connection to the Internet (WiFi or 3G) and for the interactions with the short message service (SMS).

4 Related Work

There are numerous types of location-based services [18,21,27,39] offering users to connect with friends, to discover their environment or to optimize their mobility. Hence, users can share their location in return for services. In this paper, we focus on services that enable users to share their location with friends.

The need for selective access control in online social networks (OSNs) has been identified in previous works and several solutions were proposed. Most of these solutions implement add-ons to existing OSNs in order to protect user privacy. NOYB [31], for example, encrypts personal information using a pseudo-random substitution cipher. FlyByNight [40] protects private data in Facebook by storing it in encrypted form. We rely on an encryption primitive similar to that discussed in [19] to encrypt data on LSSs.

Recently, a new breed of privacy-aware online social networks were proposed [3,11]. Basically, these OSNs offer precise settings for users to control their privacy. Effectively, all user data is still shared with the third-party running the OSN. Other works proposed fine-grained control over the sharing of user location [7]. This controls enable users to specify precisely with which friends they wish to share their location and how. In this work, we focus on the protection of users' location with respect to third-parties.

Earlier work on privacy preserving localization tried to solve the problem by caching on mobile devices the location of WiFi access points used for triangulation [9]. However, it is costly to store on mobile devices entire database of access points, and such databases may not offer a complete coverage of the regions visited by users. In this work, we extend these existing mechanisms by offering the ability to dynamically cache access points, and use dummy queries when areas are not covered.

The generation of dummy location samples was previously evaluated [43] but mostly with a focus on moving trajectories over a period of time. In this work, we identified several new constraints that must also be taken into account, in particular, we mentioned the use of virtual identities.

5 Conclusion

We have considered the problem of sharing user location privately over online social networks. We show that encryption techniques can be used to protect user location from LSSs. We note that this is not sufficient because other components, such as localization and map visualization services, may also obtain users' location. We show how dummy queries and caching can help protect location privacy. We observe that dummy queries generated at random are inefficient against an adversary with statistical knowledge about user mobility. In the worst case scenario (i.e., users are traceable), we argue that it is necessary to maintain virtual identities generating dummy queries that are statistically indistinguishable from real users. We implemented the ideas of this paper in a prototype client application code-named PrivL that can connect to any LSS.

In the future, we intend to test various dummy query and caching strategies and evaluate their effectiveness in providing location privacy. We will also evaluate the incentives of LSS operators to provide services to users that hide their location. In particular, users could strategically reveal sample of their visited locations or obfuscated location information to provide incentives to LSS operators.

References

1. Bing maps. http://maps.bing.com.
2. Citysense. http://www.citysense.com.
3. Glympse. http://glympse.com.
4. Google geolocation api. http://code.google.com/apis/gears/api_geolocation.html.
5. Google maps. http://maps.google.com.
6. ipoki. http://www.ipoki.com.
7. Locaccino. http://locaccino.org.
8. OpenDHT. http://www.opendht.org.
9. A privacy-observant location system. http://www.placelab.org.
10. Qt. http://qt.nokia.com.
11. Rallyup. http://www.getupandrally.com.
12. Skyhook wifi positioning system. http://www.skyhookwireless.com.
13. Spotrank. http://www.skyhookwireless.com/spotrank.
14. Static maps api. http://code.google.com/apis/maps/documentation/staticmaps.
15. Symbian c++ sdk api plugin. http://wiki.forum.nokia.com/index.php/SDK_API_Plug-in.
16. Wireless geographic logging engine. http://wigle.net.
17. Yahoo maps. http://maps.yahoo.com.
18. Aka Aki. The discover of a lifetime. http://www.aka-aki.com.
19. F. Beato, M. Kohlweiss, and K. Wouters. Enforcing access control in social network sites. In *HotPETs*, 2009.
20. A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *PerSec*, March 2004.
21. Google Mobile Blog. http://googlemobile.blogspot.com/2010/01/finding-places-near-me-now-is-easier.html.
22. D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO*, pages 258–275, 2005.
23. Cloudmade. Makes maps differently. http://cloudmade.com.
24. C. Delerablée, P. Paillier, and D. Pointcheval. Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In *Pairing*, pages 39–59, 2007.
25. A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1994.
26. Yahooo Fireeagle. Take your location to the web. http://fireeagle.yahoo.net.
27. Foursquare. Check-in, find your friends, unlock your city. http://foursquare.com.
28. R. Geambasu, T. Kohno, A. Levy, and H. M. Levy. Vanish: Increasing data privacy with self-destructing data. In *USENIX Security Symposium*, 2009.
29. P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive*, 2009.
30. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.
31. S. Guha, K. Tang, and P. Francis. Noyb: privacy in online social networks. In *WOSP*, pages 49–54, 2008.
32. B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM*, 2005.
33. B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys*, 2008.

34. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *Pervasive Computing*, pages 38–46, 2006.
35. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *CCS*, 2007.
36. iPhone Dev Center. App store tip: Enhance your app with core location. http://developer.apple.com/iphone/news/archives/2010/february/.
37. J. Krumm. Inference attacks on location tracks. In *Pervasive*, 2007.
38. Google Latitude. See where your friends are right now. http://www.google.com/intl/en_us/latitude/intro.html.
39. Loopt. Discover the world around you. http://loopt.com.
40. M. M. Lucas and N. Borisov. Flybynight: mitigating the privacy risks of social networking. In *WPES*, pages 1–8, 2008.
41. Ovi Maps. Navigation on your nokia. for free. forever. http://maps.nokia.com/ovi-services-and-apps/ovi-maps.
42. M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *VLDB*, 2006.
43. T. You, W. Peng, and W. Lee. Protecting moving trajectories with dummies. In *PALMS*, 2007.
44. S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. Privacy-preserving location-based services for mobile users in wireless networks. Technical report, State University of New York at Buffalo, 2005.

A Unified Framework for Location Privacy

Reza Shokri, Julien Freudiger and Jean-Pierre Hubaux

LCA, EPFL, Switzerland
firstname.lastname@epfl.ch

EPFL-REPORT-148708 June 2010

Abstract. We introduce a novel framework that provides a logical structure for identifying, classifying and organizing fundamental components, assumptions, and concepts of location privacy. Our framework models mobile networks and applications, threats, location-privacy preserving mechanisms, and metrics. The flow of information between these components links them together and explains their interdependencies. We demonstrate the relevance of our framework by showing how the existing achievements in the field of location privacy are embodied appropriately in the framework. Our framework provides "the big picture" of research on location privacy and hence aims at paving the way for future research.

1 Introduction

Location-based services are becoming ubiquitous, fueled by the proliferation of mobile devices, notably smart phones. There exist numerous applications that take advantage of the latest capabilities of mobile devices, in order to share information between users in a wireless peer-to-peer manner [1–3], or to exchange location-based information with the service providers [4, 5, 17, 20, 46, 62].

Despite the popularity of these services, privacy issues such as the undesired leakage of users' location information to location-based service operators, or to external eavesdroppers is a major concern. In this paper, we are concerned with *location privacy* that, according to Duckham and Kulik [27], is defined as a *special type of information privacy which concerns the claim of individuals to determine for themselves when, how, and to what extent location information about them is communicated to others.*

The problem of protecting the location privacy of mobile users has attracted researchers from various backgrounds such as database security and anonymous communication. Several works approach this problem from different perspectives and, hence, various protection mechanisms have been proposed. However, existing proposals usually aim at solving location-privacy problem, based on incomplete (and sometimes not fully explicit) set of assumptions, that are not fully in accordance with network and adversary models and users' privacy requirements. Without a systematic identification of the possible threats, specification of available countermeasures, and designing of appropriate evaluation metrics, there is no guarantee that the location privacy of users can be actually protected in different scenarios. In particular, the adversary's knowledge, her available technologies, access rights, potential actions, and her goals should be formalized in order to enable the design of effective countermeasures. The metrics used for measuring location privacy should be carefully designed in order to reflect the users' actual gain in employing a location-privacy preserving mechanism, by considering the users' privacy requirements and the adversary's knowledge.

Due to these considerations, we are motivated to construct a unified framework for location privacy in which the different components, that affect location privacy, are defined and their interrelations are identified. A consistent structure and terminology, proposed in this framework, allows us to better understand different works and thus position them appropriately in the field of location-privacy research. We provide a thorough study of previous works and place each of them appropriately in our framework, based on the role it plays in protecting location privacy. The framework, further, paves the way for future research in this field, by introducing a common notion for location privacy. It will help identifying the shortcomings of the existing approaches and discovering neglected aspects and hence open problems of location privacy.

In our framework, we introduce the various components of location privacy according to the flow of information from the users to the adversary. A location-privacy preserving mechanism acts as a noisy channel that modifies the information that is communicated from the users (as the source of information) to the adversary (as the observer/receiver). Users' location privacy is maximized if the adversary cannot correctly link their location and identity over time. In other words, using the information she obtains by observing users' activities from behind the curtain of privacy preserving mechanisms, more distorted is users' location in the adversary's eyes, the higher their location privacy is.

The structure of the paper is as follows. First, in Section 2, we model *mobile networks* and formalize the state of the real world (i.e., the users' mobility). Further, in Section 3, we model the spatiotemporal image of users' activities after being distorted by *location-privacy preserving mechanisms*. In Sections 4 and 5, we provide an elaborate model for the *adversary*, different categories of location privacy, the methodology for measuring location privacy, and an exhaustive study of existing *metrics*. In Section 6, using our framework, we briefly model and study location privacy in an example scenario: location-based services. Finally, in Section 7, we survey the existing models in the literature, before concluding the paper in Section 8.

To the best of our knowledge, this is the first paper that takes the many aspects of location privacy into account, clarifies their interrelation, and proposes a unified framework in which the existing achievements are embodied.

2 Mobile Networks

A mobile wireless network consists of a set of mobile users equipped with wireless devices that are capable of establishing ad hoc communication among themselves and/or connecting to infrastructures (e.g., cellular networks, and WiFi access points) in order to use a *common service* using the appropriate *application*. There exist many applications that can help users to access the provided services in mobile networks. Location-based services, mobile social networking, mobile recommender systems, friend finder services, people-centric sensing systems, and ad hoc networking are some examples of the services provided in mobile networks.

Users can employ a wide range of applications in mobile networks. These applications can be categorized into subsets based on their different features. Because our focus is on how users' location can be used by an application, we divide the applications into two dimensions: *automatic* vs. *manual*, based on the way information is shared by the application; *continuous* vs. *discrete*, based on the time distribution of information sharing. As an example, people-centric sensing applications, where users upload information about their environment to a central server, are *automatic* and *continuous*. Electronic ticketing, which is used in public transportation systems, is *automatic* and *discrete*. Some of the applications such as location-based services encompass a variety of applications, each of which has different features. However, most of the popular location-based services are *manual* and *discrete*.

2.1 Users

We denote by $\mathcal{U}$ the set of users who are members of the mobile network. Depending on the provided service, users might also belong to online social networks. We assume that each and every user is associated with two distinct types of identities: A *real identity* and a set of *pseudonyms*.

The real identity of a user is any subset of his attributes (e.g., name, national identity number, and private key) that uniquely identifies the user within the set of users $\mathcal{U}$ [61]. The real identity of each user is unique and invariant over time. Let $\mathcal{I}$ denote the set of real identities associated with all users. There is a bijective relation between a user and his real identity. This is captured by function $\textbf{name} : \mathcal{U} \to \mathcal{I}$ that maps each user with his real identity.

Every user is also associated with a set of *pseudonyms*. Pseudonyms are mostly temporary identifiers that facilitate identification and authentication of a user in a communication without revealing the *user*'s real identity. MAC addresses of wireless devices, IP addresses, public keys, group signatures, physical layer fingerprints of wireless devices, signature of the applications, and the combination of these identities are different examples of pseudonyms. The user to whom a particular pseudonym refers is called the *holder* of the pseudonym. If it is shared by multiple users, a pseudonym is called a *group pseudonym*. Based on the information derived from a pseudonym *per se*, no one (except the pseudonym's holder) can identify the holder of the pseudonym [61]. Let $\widehat{\mathcal{I}}$ denote the set of pseudonyms used by all the users. The set of real identities and pseudonyms are disjoint, i.e., $\widehat{\mathcal{I}} \cap \mathcal{I} = \emptyset$. We define function $\textbf{nyms} : \mathcal{U} \to \mathcal{P}(\widehat{\mathcal{I}})$ that gives the set of pseudonyms associated with each user, where $\mathcal{P}(\widehat{\mathcal{I}})$ is the power set of $\widehat{\mathcal{I}}$. A user can use various pseudonyms over time. Pseudonyms can expire or be revoked from the network. Lastly, we define $\widehat{\mathcal{I}}(t) \subseteq \widehat{\mathcal{I}}$ to denote the set of pseudonyms that are still valid at time t.

2.2 Time and Space

The notion of time that we employ in this model is discrete, and each unit of time, which is a natural number, is called a *time instance*. We define a *time period* (t_s, t_e) to be the set of all time instances between t_s and t_e including t_s and t_e. Let $\mathcal{T}$ be the time period in which we model the system.

We employ a three-layer model in order to represent the space in which users can move, and its associated contextual information. The first layer models the *geographical space*. The second layer models the places or *location sites*, and the third layer models the *type of sites*.

We model the first layer, i.e., the geographical space in which users can move, in a discrete way. Let $\mathcal{L} \subset \mathbb{N}^2$ be the grid that represents the two dimensional space. We name any member of the set $\mathcal{L}$ a *location instance*. We also call a non-empty set of location instances a *location area*. Each location instance belongs to a *location site* that has a unique name and address (e.g., a hospital, an avenue, or a house with its unique ZIP code). In other words, any location instance has a tag name, and clearly, multiple location instances can have the same name because they belong to the same site. The location sites, which are constructed on top of the location instances, build the second layer of the model. The third layer captures the type of location sites and their similarity. A location type represents the usage of the location, e.g., shopping, cultural, sporting, residential.

Note that the granularity of time and location depends on the application and we do not make any assumption about that.

2.3 The Spatiotemporal State of Users

As users are mobile, their location is a time-dependent value. To model the users mobility, let **whereis** $: \mathcal{U} \times \mathcal{T} \rightarrow \mathcal{L}$ be a function that gives the *actual location* of users at any time instance in $\mathcal{T}$. Note that this function gives the exact location of each user, regardless of the knowledge of any entity about it (i.e., maybe even the user himself does not know his accurate position, because of, for example, the inaccuracy of his GPS device).

We model a mobile network in terms of the location-based events that occur in the network. Events in our framework reflect the spatiotemporal state of the users in the real world and also in the eyes of an observer.

We define an *event* as a 3-tuple $\langle i, t, l \rangle$, where $i \in \mathcal{I} \cup \widehat{\mathcal{I}}$ is the real identity or a pseudonym of a user, t is the time instance at which the event occurs (referred to as the *time-stamp* of the event), and $l \in \mathcal{P}(\mathcal{L})$ is the location area associated with the event (referred to as the *location-stamp* of the event). Let $\mathcal{E}$ denote the set of all possible events. We define three functions $\mathbf{id} : \mathcal{E} \rightarrow \mathcal{I} \cup \widehat{\mathcal{I}}$, $\mathbf{tm} : \mathcal{E} \rightarrow \mathcal{P}(\mathcal{T})$, and $\mathbf{loc} : \mathcal{E} \rightarrow \mathcal{P}(\mathcal{L})$ that give the identity, time-stamp, and location-stamp of an event, respectively. Thus, for any event $e \equiv \langle i, t, l \rangle$ we have $\mathbf{id}(e) = i$, $\mathbf{tm}(e) = t$, and $\mathbf{loc}(e) = l$. We further define a *trace* to be a non-empty set of events.

An event e is called an *actual event* associated with a user u if the following conditions hold: $\mathbf{id}(e) = \mathbf{name}(u)$, $\mathbf{loc}(e) = \mathbf{whereis}(u, \mathbf{tm}(e))$, and $\mathbf{tm}(e) \in \mathcal{T}$. Thus, an actual event represents the spatiotemporal status of a user in the real world. Following the definition of actual events, the actual trajectory of a user is the trace of all his actual events. Let $\mathcal{R}_u$ denote the trajectory of a user during $\mathcal{T}$. Hence, $\mathcal{R}_u = \{\langle \mathbf{name}(u), t, \{\mathbf{whereis}(u,t)\}\rangle, \forall t \in \mathcal{T}\}$.

We define relation $\sim$ between two actual events e_i and e_j, if and only if they are associated with the same user, i.e., $e_i \sim e_j \Leftrightarrow \exists u \in \mathcal{U}$ s.t. $e_i, e_j \in \mathcal{R}_u$. Let $\mathcal{R}$ denote the set of all actual events of all users. Thus, $\mathcal{R} = \bigcup_u \mathcal{R}_u$, and $\forall u, v \in \mathcal{U}$ we have $\mathcal{R}_u \cap \mathcal{R}_v = \emptyset$. Hence, the relation $\sim$ partitions the trace $\mathcal{R}$ into subsets, each representing the actual trajectory of one user. We refer to the set partition associated with $\sim$ as the *actual set partition* of $\mathcal{R}$.

3 Location-Privacy Preserving Mechanisms

Actual events represent the state of the real world, i.e., the actual locations of users over time. A user has absolutely no location privacy if an adversary has access to the set of his actual events. To protect a user's location privacy, some privacy preserving mechanisms should alter the information observable by the adversary. Ideally, the amount of information leakage should be minimal, while enabling users a proper use of the service. There are three entities that play a role in preserving location privacy: *users*, *applications*, and *privacy tools*. Each entity controls the amount of shared information and thus affects user privacy. Users and applications might *intentionally* (e.g., by being cautious about sharing unnecessary information) or *unintentionally* (e.g., by sharing incorrect information) reduce the amount of information revealed. Privacy policies influence the way applications can share information with different entities, and they are applied to the application based on the users' decisions. Various privacy tools, also, use sophisticated algorithms to guarantee users' privacy.

In order to capture the effect of the three above-mentioned entities in preserving location privacy of users, in our framework, we abstract away the entities and model a location-privacy preserving mechanism as a single unit that separates actual events of the users (i.e., the ground truth) and the adversary. Formally we define a *location-privacy preserving mechanism* as a transformation function that modifies the users' actual events before they can become observable by any observer.

Privacy tools (as the main entity in location-privacy preserving mechanisms) work in three architectures: (i) *Distributed* (user-side): They can work in a distributed way by being implemented on individual mobile devices, where each device itself transforms its events and modifies what an observer can see about the user's spatiotemporal state. This can be done either with the help of information that a device gets from other devices or exclusively with the information that the user has himself. (ii) *Centralized* (server-side): They can work in a centralized manner by using a trusted central server that acts as a privacy preserving proxy and modifies users' messages (correspond to events in our model) before being observable by an untrusted entity. (iii) *Hybrid*: They can be a hybrid of both distributed and centralized architectures.

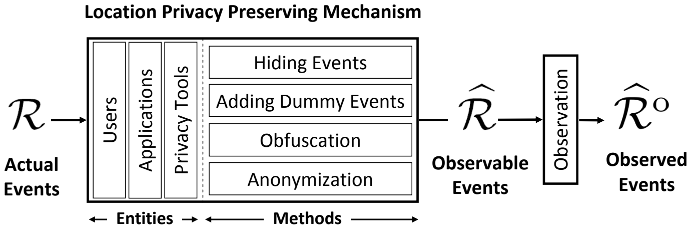


Fig. 1. Location Privacy Preserving Mechanisms

We define function $\mathbf{trns} : \mathcal{E} \to \mathcal{E} \cup \{\textsc{hidden}\}$ to denote the location privacy preserving mechanisms, where HIDDEN stands for a *hidden event* (i.e., an unobservable event). The output events of the transformation function on the set of actual events is called the set of *observable events* and is denoted by $\widehat{\mathcal{R}}$, i.e., $\widehat{\mathcal{R}} = \overrightarrow{\mathbf{trns}}(\mathcal{R}) \setminus \{\textsc{hidden}\}$, where $\overrightarrow{\mathbf{trns}}()$ is the image of function $\mathbf{trns}()$ on a trace. Mechanisms perform the transformation function by means of four different *methods* (which can also be called the primitives of location-privacy preserving mechanisms): *hiding events*, *adding dummy events*, *obfuscation*, and *anonymization*. These four methods together can model any transformation on the events: the first two (*hiding events*, and *adding dummy events*) modify the set of the events, *obfuscation* modifies time- and location-stamps, and *anonymization* modifies identity of events. Figure 1 illustrates the role of different entities that can employ a combination of these methods in order to alter the accessible information to an observer. The methods are described below.

- **Hiding Events** The most basic method for protecting users' location privacy consists in hiding information about the trajectories of users. A subset of events are removed in the transformation process. This is modeled by replacing the candidate events by HIDDEN in the **trns** function, which will not further appear in $\widehat{\mathcal{R}}$. This method is implemented mostly in distributed architectures where mobile devices refrain from transmitting information by being silent during certain time periods. Privacy sensitive users, or privacy tools (e.g., [10, 44, 45, 47, 53]) make use of mainly this method (along with other method). It can also be implemented in the centralized architecture. A service provider who follows some privacy policies, in practice, is applying this method. The mechanisms proposed in [40, 42] are examples of using event-hiding privacy tools, especially, in centralized architecture.
- **Adding Dummy Events** The other method to achieve privacy is to mislead an observer by adding some *dummy* events through the event injection method of the transformation function. This method can be effectively implemented in the centralized architecture. Mechanisms proposed in [19, 49, 51, 55, 71] employ mainly this method. Generating a trace of events that looks like a normal user's trajectory is one of the main challenges of this set of papers.

- **Obfuscation** Using this method, the location-stamp and/or the time-stamp of the actual events in $\mathcal{R}$ can be altered. Obfuscation methods result in *inaccuracy* or *imprecision* of the location/time of the events [26]. This is done by adding noise to the location- and/or time-stamp of the events or by coarse graining them. The method can be implemented in both distributed and centralized (and hence the hybrid) architectures using various algorithms. In the existing privacy preserving mechanisms, obfuscation is achieved mostly through *perturbation* [38, 56] or *generalization* [7, 8, 18, 35, 38] algorithms.
- **Anonymization** Using the anonymization method, the identity of an event in $\mathcal{R}$ is altered in order to break the link between a user and his events or to make a user's events unlinkable to each other. To this end, in the transformation function, the real identity of a user on each event can be replaced by one of his valid pseudonyms. In the centralized architecture, this method is implemented mainly by replacing all the events' identities with a single group pseudonym (full anonymization by having no identity, i.e., null pseudonym [66]), e.g., Cornielius *et al.* [21]. In distributed architectures, users themselves change their pseudonyms from time to time. This pseudonym change is done usually in some predetermined places called *mix zones* (Beresford and Stajano [10]) where users remain silent when they are inside the mix zone and change their pseudonyms when they leave the zone. Thus, these mechanisms employ the hiding method as well, e.g., Jiang *et al.* [47]. The pseudonym change can also be done in a self-organized way by using group-signatures (e.g., Calandriello *et al.* [16]) or ring-signatures (e.g., Freudiger *et al.* [32]) as group pseudonyms. The mechanism proposed by Li *et al.* [53] also makes use of group pseudonyms, and users exchange their group pseudonyms with each other when they leave mix zones. Buttyan *et al.* [15] and Freudiger *et al.* [31, 30] proposed formal models to evaluate the effectiveness of static and dynamic mix zones, respectively, (users decide on-the-fly whether to change their pseudonyms or not) in mobile ad hoc networks.

We denote the observable trace of a user $u \in \mathcal{U}$, which is the output of the location privacy preserving mechanism applied on the user's actual trace, by $\widehat{\mathcal{R}}_u = \overrightarrow{\mathbf{trns}}(\mathcal{R}_u)$. Thus, $\widehat{\mathcal{R}} = \bigcup_u \widehat{\mathcal{R}}_u$. If two actual events e_i and e_j are associated with the same user (i.e., $e_i \sim e_j$) and $\hat{e}_i = \mathbf{trns}(e_i)$ and $\hat{e}_j = \mathbf{trns}(e_j)$, then we define a relation $\sim_o$ between $\hat{e}_i$ and $\hat{e}_j$. In other words, we define $\hat{e}_i \sim_o \hat{e}_j$ if $\hat{e}_i, \hat{e}_j \in \widehat{\mathcal{R}}_u$ for some $u \in \mathcal{U}$, which represents the linkability of observable events.

4 Threat Model

Depending on the characteristics of the mobile network and the services provided for users, location privacy of users can be threatened in different ways. The adversary can be an entity who eavesdrops on wireless communications between users, or she can be the operator who provides a location-based service for her subscribers, or even she can be one of the users in the network who participates with other users in running a protocol. In our framework, the adversary

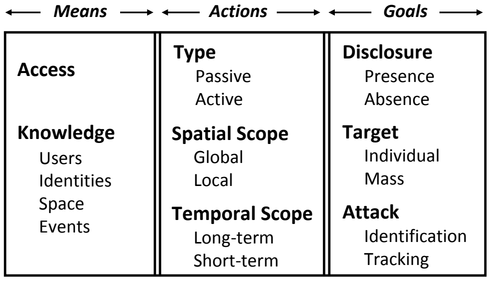


Fig. 2. Structure of the threat model and adversary's attributes

is actually the entity who observes the output of privacy preserving mechanisms and, hence, has access to a subset of the observable events $\widehat{\mathcal{R}}$. The subset of $\widehat{\mathcal{R}}$ that is accessible to a given adversary is called the set of *observed events* by that adversary and is denoted by $\widehat{\mathcal{R}}^o \subseteq \widehat{\mathcal{R}}$. The properties of this subset and the implication of this observation on the users' location privacy depend on the characteristics of the adversary. Note that the adversary might have multiple observation points from each of which she can observe a different set of events. At each observation point, the adversary, observes different transformations of the same actual events. However, the structure of transformations (not their settings) is the same. Thus, in this section, we focus on the set of adversary's observed events at a single observation point, which is shown by $\widehat{\mathcal{R}}$.

In this framework, we model an adversary based on the following three factors: her *Means*, *Actions*, and *Goals*. Each of these factors are explained in the following sections and illustrated in Figure 2.

4.1 Means

The means of the adversary are the *technologies* available to her for capturing events, her *access* credentials in the system, and her a priori *knowledge* about the system.

Access — The adversary might eavesdrop on the wireless communication of users. Based on the level of sophistication of her eavesdropping devices, the accuracy of observed events changes. It can also be the entity in the organization that provides the service for users. For example, the adversary might be the insurance company who periodically collects positions of vehicles in a Pay-As-You-Drive application [68], or can be the operator of an automatic toll collection system who can get more sparse information about location of vehicles. Fingerprinting wireless devices [14, 22, 29] is also one technique that can be used in order to extract pseudonyms correlated to the hardware used by a user rather than based on the content of his messages (e.g., IP address). As another example, the adversary might have access to the high-level transactions of a location-based service (LBS) or an environment monitoring network [39].

Knowledge — The a priori knowledge of the adversary is composed of multiple pieces. Here, we categorize the adversary's knowledge into multiple classes. The precision and confidence of the adversary's knowledge about each class determines her a priori knowledge. Her knowledge in each class can be deterministic or probabilistic and this must be clarified in each threat model.

– **Users** The adversary might know the (exact or estimated) number of users at any time, or more precisely the set of users $\mathcal{U}$, that implies knowing the real identity of active users. This knowledge can evolve over time, or she may remain oblivious about the dynamics of the set of users and their joining/leaving. This class of adversary's knowledge also includes the adversary's knowledge about the relation between users, i.e., social network graph.
– **Identities** This class specifies to what extent the adversary knows about the users' identities and the pseudonyms used by them. The adversary might know the relation between pseudonyms of a user, and also the constraints on the set of pseudonyms (e.g., how many pseudonyms a user can have). The extent to which each pseudonym is linkable to its holder's real-identity is also part of the knowledge of the adversary in this class.
– **Space** The knowledge of the adversary on the three-level model for the space in which users move falls into this class. The connection between users and places must also be specified here. For example, does the adversary know the address of the users' homes or their workplaces, which are modeled in the second layer of our space model?
– **Events** The adversary might have access to some actual events that are performed before the observation time. Moreover, in many cases the adversary has some statistics about the typical behavior of users. For example, she knows the (im)possibility or the probability that one specific actual event can be performed by a user, or that two specific events belong to the same user. Knowledge of the adversary about mobility profile of users (which represents how probable/possible it is for a specific user or a mass of users to move from one location to another location in a specific time period) falls into this class.

We assume that the adversary knows the application, employed privacy tools, and also the location-privacy metric that users (or system designers) use.

4.2 Actions

The adversary might be *passive* and only observe the network, or, in addition to that, become *active* and interact with the users in order to obtain more information about them. In most of the cases in the literature, the adversary is only a passive observer. However, one can imagine some sophisticated active attacks, such as the following ones. By relaying [60] the traffic between two different mix zones [10], the adversary can create a fake high-density area and encourage neighboring users to change their pseudonyms. Thus, users have an erroneous feeling of high privacy while the adversary can easily distinguish between pseudonyms in each mix zone and link the users' pseudonyms. The adversary can also create fake toll readers in vehicular networks by using a relay attack and force users to reveal their presence in critical locations. Here, the adversary actively participates in asking users to generate some events.

The action scope of an adversary is determined by the size of the location areas and the duration of time periods in which the adversary observes the system. Considering these factors, adversaries can consequently be divided into different categories. An adversary is *global* if she observes the observable events occurred at any location in the space. Whereas, she is called *local* if during the observation period she cannot observe the transformation of some events that are generated in specific location areas. Similarly, based on the observation time, an adversary is referred to as a *short-term* attacker if the transformation of events performed at some time periods are not observable by the adversary. In the case there is no such time restrictions, she is called a *long-term* attacker. In the case an attack is global and long-term, we have $\widehat{\mathcal{R}}^o = \widehat{\mathcal{R}}$.

4.3 Goals

Presence vs. Absence Disclosure — An adversary's goals of observing users' activities in a mobile network can be divided into two main categories: *presence disclosure* or *absence disclosure*. In the former category, the adversary's goal is to find out if a given user or a set of users are present at some place(s). Whereas, in the latter category, the adversary wants to know whether a specific set of users are *not* present at some place(s). Virtually all of the attacks presented in the literature fall into the first category. However, there are some reports about the consequences of absence disclosure attacks on people. As an obvious example, by misusing her access to an LBS database, the adversary can find out the best time to break into a person's house, or blackmail them. An implementation of this attack can be found here [6]. Obfuscating the victim's location as big as the whole North America cannot protect him from absence disclosure attacks if he lives in Europe, despite the fact that the adversary cannot locate his exact location. However, using that obfuscation, if the user lives in the US, his location privacy is protected against both presence and absence disclosure attacks.

Individual vs. Mass Target — The attacks can disclose the private information of a specific user in an *individual target* attack, or it might be targeting a set of users, collectively, in a *mass target* attack where the adversary does not distinguish users in the set, for example when they belong to a community.

Tracking vs. Identification — The two main known attacks on users' location privacy, which are used usually to disclose users' presence, are *tracking* and *identification*. These two attacks are tightly related to each other, although they have different ways of obtaining users' private location information.

In *tracking* attacks, the adversary's goal is to reconstruct the users' actual trajectories (which might have been distorted by privacy preserving mechanisms) and subsequently identify the locations that the users have visited. This information can be used to predict the future locations of users. The tracking can be done in various manners depending on the adversary's goal. The adversary might want to know the trace of location instances (i.e., coordinates) visited by the users in a given time period, or the location sites (e.g., specific hospital) where they have been to, or only the type of places that the users are used to periodically visit.

In *identification* attacks, the adversary wants to discover the real identity of her targets. This can be done on small scale where the adversary is interested in de-anonymizing a specific observed event, or on a large scale where the adversary is interested in finding the identity of users from whom the adversary has observed some anonymous traces of events. The identification is done using some inference attacks based on the adversary's knowledge on the linkability of the users to sensitive areas such as their homes or work places [37, 41, 50]. Identification can also leverage on the mobility pattern of users, because users tend to visit certain places regularly [23]. It can also be done indirectly through de-anonymizing [58] the social network that is linked (e.g., friend-finder applications) to the users' observed events.

It is clear that the success of each of the two above-mentioned attacks also paves the way for the other. In the case the adversary manages to discover the actual trajectory of a user, the identification of the user is not a difficult task. Especially if the adversary has access to the information about location sites such as homes or work places of the users, which contain a lot of information about their identities. In the case that the adversary has already de-anonymized some events of a user, the recovery of the user's actual trajectory (i.e., tracking him) can be done more easily, if the adversary has access to the mobility profile of the users.

Tracking and *identification* attacks have been individually studied in the literature. However, there is not much research done on *modeling* the inter-relation between them, which the adversary is likely to make use of (a combined attack). Considering that many privacy preserving mechanisms focus on protecting users from one of these two attacks, it is of upmost importance to analyze to what extent the adversary can break the privacy of users by running unexpected attacks, e.g., de-anonymizing the traces using the social network graph available to the adversary. Lastly, some inference attacks can be developed in order to recognize the activities of the targets and track the types of places they visit (which may eventually leads to their identification) [54].

5 Location-Privacy Measurement

Every location-privacy preserving mechanism is designed based on an assumed location-privacy metric. In the literature, various metrics have been used to capture location privacy in different scenarios. Needless to say, choosing the right metric for each specific setting is of upmost importance to increase the actual users' location-privacy against possible attacks. Considering different users' privacy needs, we define location privacy on macro and micro levels. We define the notion of *macroscopic* location privacy to be a user's privacy level throughout his trajectory, whereas the *microscopic* location privacy is defined to represent a user's privacy on a small scale, for example at the time an event related to a given user is observed. These two metrics are tightly related to each other, and reflect location privacy in two different scales. The selection of the right scale highly depends on the threat model and on the specification of location-privacy requirements. Using these notions, we classify the metrics that have been used so far, and thus we can discuss their effectiveness in representing the true privacy level of users.

Microscopic Location-Privacy We expect the location privacy of a user at the micro level, depending on the user's privacy requirements, to be inversely proportional to the success of the adversary in identifying his real identity when an event is observed of a user, or to locate the user at a given time instance and find out his presence/absence at a given location. The more accurately the adversary can locate a user, the poorer the location privacy of the user will be. Depending on the threat model, the adversary might be interested to find out the coordinate of the user's location, his location site, or the type of his location.

The most popular metric for this micro location privacy is based on the *uncertainty* of the adversary. This metric was originally proposed by Diaz *et al.* [25] and Serjantov and Danezis [65] for anonymous networks (known as entropy-based or information theoretic-based metrics), and by Samarati and Sweeney [63, 67] for database privacy (known as k-anonymity metric, where, assuming maximum uncertainty for the adversary, k is equal to the effective anonymity set size [65]). These metrics were adapted to measure microscopic location privacy by Gruteser and Grunwald [38], and later used in many papers such as [7, 12, 34, 35, 48, 57, 64, 70, 72]. Virtually all of the various versions of uncertainty-based metrics for micro location-privacy, measure *only* the adversary's success in the presence-disclosure identification attacks. The metric reflects a given user's privacy, from whom an event is observed by the adversary, as the size of the effective anonymity set in which the user is hidden.

Macroscopic Location-Privacy How accurately the adversary can track a user throughout his trajectory (i.e., tracking attack), or how closely she can find out the identity of a user after observing a set of events (i.e., identification attack), is reflected by macroscopic location privacy metrics. The set of macro metrics can be divided into two main categories, based on the set of the criteria that is used in each metric: *uncertainty*-based and *error*-based metrics.

— The first set of macro metrics reflects the *uncertainty* of the adversary in tracking users. Similarly to the micro metrics, in this category, entropy-based metrics and k-anonymity are the two most popular measures. The set of observed events are linked to each other, based on the a priori knowledge of the adversary, in a way that each link shows the possibility and also probability of observing the two linked events from the same user. The entropy-based metrics exploit this data structure and compute the adversary's uncertainty at the outgoing links of each observed event. These values reflect the user's privacy throughout his trajectory, however, only at the time instances that an event is observed of the user. The overall privacy level is measured mainly as the fraction of times when the uncertainty is below a threshold [9, 10, 30, 40, 42, 44, 45, 47, 53]. The k-anonymity metric is extended also to the macro level by measuring how many trajectories at a time period are indistinguishable in an anonymity set [12, 36, 59, 69].

— The second set of macro metrics are based on the adversary's *error* in tracking/identifying users. This category is divided into multiple subclasses: (i) *clustering-error*, (ii) *probability of error*, and (iii) *distortion-based* metrics.

(i) *Clustering-error* metrics: The adversary's goal is defined to be the clustering of the observed events into partitions, based on the relation $\sim_o$, each partition for one user. Formally, the adversary is looking for $\widehat{\mathcal{R}}_u$ for all $u \in \mathcal{R}$. Two slightly different versions of this metric are used in [28, 39]. Note that both versions aim at measuring the success of adversary's tracking attack.

(ii) *Probability of error* metrics: The adversary's probability of error in finding the real identity of a user, or linking his observed events, is considered as the metric. For identification attacks, in [50] various algorithms using machine learning techniques are proposed to identify the *homes* of mobile users in the users' observed events and subsequently find their identities based on the adversary's knowledge. Similarly, in [41] an algorithm is proposed to identify users based on their home addresses. The higher the average adversary's probability of error is, the higher the users' location privacy is in their model. In [33] the probability of error is used as the metric to evaluate users' location privacy against tracking attacks in mix zones.

(iii) *Distortion-based* metrics: Having prior knowledge about the system, and after observing a set of events, the goal of the adversary of perpetrating the tracking attack is to reconstruct the actual trajectory of the users. The distortion-based metric [66] reflects how distorted the reconstructed trajectory of each user will be for the adversary. To measure the distortion, it is enough to condition the possible actual trajectories of the targets to the observed events, and compute the expected distance of the predicted location of a user with his actual location at any time instance. It is shown that this metric is superior to the other macro metrics that focus on tracking attacks, in terms of the accuracy of the metric. A set of criteria, derived from the definition of location privacy, is also proposed to compare the effectiveness of the metrics.

Here, we discuss the next steps towards the definition of more accurate and realistic location-privacy metrics. The departure point is the threat model. Concerning presence disclosure attacks, we need metrics that represent users' location privacy in the presence of identification/tracking attacks. In identification attacks, *uncertainty*-based metrics (when we are measuring the system level privacy) and metrics based on the adversary's *probability of error* (when we measure user level privacy) are more representative of users' location privacy. Regarding tracking attacks, *distortion*-based metrics are shown [66] to be more accurate than other metrics, as the adversary's goal is to predict as closely as possible the users' actual locations over time, and the more distorted her prediction is the higher users' location privacy is.

6 Application Scenario: Location-based Services (LBSs)

Using the proposed framework and terminology, we briefly model location-privacy in a typical LBS, as a small example to show the effectiveness of the framework.

Consider a LBS in a region, by means of which users can obtain the list of their nearby public places, by sending their GPS coordinates to the server. Users might either subscribe to the server in order to rate the places they visit, or simply use the service without authenticating to the server. As users decide when to use the system and this is done infrequently, the application is *manual* and *discrete*. In this kind of LBS, users do not need to form any social network on the server to use the service. Thus, their relation is not disclosed to the service provider through this service.

The three entities (users, applications, and privacy tools) employ location-privacy preserving methods in the following way. Users connect to the server with *pseudonyms* and their real identities are hidden. A pseudonym is the concatenation of IP address, cookie id, and username in the application. Hence, the first method for *users* in **trns** function is *anonymization*. The more basic method used by users is *hiding events*, as they do not connect to the server at every time instance and are hidden most of the times. The application on the mobile devices that connects to the LBS, also unintentionally uses *obfuscation* method by perturbing the actual events' location-stamps (due to the error of GPS devices). The third entity that is active in **trns** is the set of *privacy tools* that can be implemented in both distributed or centralized architectures, by using all four methods in **trns**. However, compared to the distributed architecture, the centralized form is more powerful but less practical. Especially, the method of *adding dummy events* can highly increase users' location-privacy in centralized architecture, where the privacy tool can fully anonymize users (by using the same group pseudonym for all users).

We assume the LBS operator aims at identifying users, and hence she is the adversary in our threat model. This explains the *technology* and *access* credentials of the adversary. Her *knowledge* can be modeled as following. She knows a subset of $\mathcal{U}$ and their real identities in $\mathcal{I}$. This is because only a fraction of the users in a region are known to the adversary. Moreover, she does not certainly know $|\mathcal{U}|$. However, she can estimate how many different users actively use the system at a specific time. She also has access to the home and work addresses of some users in $\mathcal{U}$, for example those have made this information available online, or by accessing a governmental database that stores this information. She is well aware of the space in which users move (GIS information modeled as the 3-layer structure in our framework), because it is part of the provided service. She has some statistics (with some precision) about users' typical mobility in the region. Any part of her knowledge is subject to error, as the adversary has some level of uncertainty in them. This must be modeled and quantified in the computational threat model. These collectively model the adversary's *means*.

In terms of the adversary's *actions*, in our model, she can be classified as a *passive*, *global*, and *long-term* observer. Let us assume her *goal* is to perpetrate *presence disclosure* attack on *individual* targets. Then, she is able to execute both *identification* and *tracking* attacks. Here, we explain that these two attacks reinforce each other and have a tight dependency to each other in this scenario. She first *tracks* users (while they are pseudonymous) and clusters their observed events, by using her knowledge about users' pseudonyms and mobility patterns. Then, she tries to *identify* each pseudonymous user, as she has access to the information about their sensitive places (i.e., home and work). After a user is identified, his observed events are de-anonymized and thus he can be tracked more accurately. This information might even help the adversary to find out the users' locations from which they did not connect to the server. Thus, there is a strong dependency between *identification* and *tracking* attacks in this application.

The notion of location privacy that users are more likely to be concerned about is *macroscopic* location-privacy, as both of the above-mentioned attacks work at a large scale. The location-privacy metrics must capture the adversary's success in both attacks, considering their dependency. From our model of location privacy in LBSs, as discussed above, there are some open problems yet to be addressed in location privacy of LBSs: Modeling the adversary's a priori knowledge and incorporating her uncertainty in her knowledge; Modeling the interrelation of the two attacks; Capturing ultimate success of the adversary by a metric; Evaluating the users' location-privacy without employing privacy tools and depending only on the mechanisms that are (un)intentionally used by *users* and the *application* on his mobile device; and Measuring the impact of the method *adding dummy events*, especially in centralized architecture.

Other applications can similarly be modeled within our framework and interdependency between various components of location privacy can be identified. As it is shown, this results in finding the drawbacks of existing approaches and suggestions for improving them.

7 Related Work

In this section, we briefly survey the papers that formalize location privacy or give an overview of location-privacy problem.

Hong and Landay [43] introduce a basic toolkit, called Confab, for developing privacy-sensitive ubiquitous computing applications. The requirements of end-users (e.g., decentralized control, special exceptions for emergencies, and plausible deniability) and also application developers needs (e.g., support for optimistic and pessimistic applications, access control mechanisms, and logging) are considered.

As an important technique to protect users' location privacy in LBSs, Duckham and Kulik [26] propose a formal model for obfuscation mechanisms. The authors provide an algorithm to balance each user's desired quality of service against their need for location-privacy.

Bettini *et al.* [11] model the microscopic location-privacy of users in location-based services. The authors take a few different kinds of knowledge the adversary could acquire, and evaluate users' privacy using uncertainty-based metrics.

Decker [24] gives an overview of location-privacy problems in LBSs and divides them into two main classes: direct and indirect attacks. Furthermore, the technical approaches to prevent misuse of location data are classified in the following categories: policy approaches, anonymization, and deliberate impairment of locating. The role of legal regulations in protecting users' location privacy is also discussed.

Blumberg and Eckersley [13] present a list of emerging threats and opportunities of location-aware services that create digital repositories of people's movement and activities. As a way to protect people's location privacy in the short run, the authors refer to "using cryptographic tools" for building systems that blindly provide location-based services and cannot infer information about people's location. The authors believe that, in the long run, "the decision about when we retain our location privacy (and the limited circumstances under which we will surrender it) should be set by democratic action and lawmaking."

Krumm [52] provides a literature review of computational location privacy. The authors discuss the need for sharing location information and also the value that people put on preserving location privacy. Going through a list of threats and countermeasures, the author, state that the progress in computational location privacy is dependent to the accuracy of location privacy metrics.

Shokri *et al.* [66] propose a framework for modeling and evaluating macroscopic location-privacy metrics. Within this framework, they formalize various metrics and, based on a set of criteria derived from the definition of location privacy, the authors study the effectiveness of existing metrics in reflecting the actual users' location privacy. Finally, they propose a distortion-based metric and show that it is superior to other existing macro metrics.

As discussed, all these works focus on formalizing a specific problem of location privacy, e.g., particular protection mechanisms, and therefore do not provide a generic framework that encompasses all location-privacy components.

8 Conclusion

In this paper, we propose a framework for location privacy that unifies its relevant components, considering users' actual location-privacy requirements. We identify different categories of threats, and establish a methodology for measuring location privacy in different scenarios in order to identify appropriate location-privacy metrics. The proposed framework enables us to design and build appropriate location-privacy protection mechanisms, identify the drawbacks of existing works, express different works with the same terminology, and discover new directions for research in location privacy.

Acknowledgment

The authors would like to thank Marco Gruteser, Claudia Diaz, and Carmela Troncoso for their insights and suggestions on earlier versions of this work.

References

1. http://en.wikipedia.org/wiki/Bluedating.
2. http://www.aka-aki.com.
3. http://csg.ethz.ch/research/projects/Blue_star.
4. http://www.madeinlocal.com.
5. http://www.loopt.com.
6. http://www.pleaserobme.com.
7. C. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In *Data and Applications Security XXI*, 2008.
8. B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 237–246, New York, NY, USA, 2008. ACM.
9. A. R. Beresford. *Location privacy in ubiquitous computing.* PhD thesis, University of Cambridge Computer Laboratory, 2005.
10. A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, page 127, Washington, DC, USA, 2004. IEEE Computer Society.
11. C. Bettini, S. Mascetti, X. S. 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.
12. C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *In 2nd VLDB Workshop SDM*, pages 185–199, 2005.
13. A. J. Blumberg and P. Eckersley. On locational privacy, and how to avoid losing it forever. Technical report, Electronic Frontier Foundation (EFF), 2009.
14. V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127, New York, NY, USA, 2008. ACM.
15. L. Buttyán, T. Holczer, and I. Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *ESAS*, pages 129–141, 2007.
16. G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy. Efficient and robust pseudonymous authentication in vanet. In *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 19–28, New York, NY, USA, 2007. ACM.
17. A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-centric urban sensing. In *ACM WICON*, 2006.
18. C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *GIS '06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 171–178, New York, NY, USA, 2006. ACM.

19. R. Chow and P. Golle. Faking contextual data for fun, profit, and privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 105–108, New York, NY, USA, 2009. ACM.
20. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *ACM MobiSys*, 2008.
21. C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 211–224, New York, NY, USA, 2008. ACM.
22. B. Danev and S. Capkun. Transient-based identification of wireless sensor nodes. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 25–36, Washington, DC, USA, 2009. IEEE Computer Society.
23. Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. Identification via location-profiling in gsm networks. In *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 23–32, New York, NY, USA, 2008. ACM.
24. M. Decker. Location privacy-an overview. In *ICMB '08: Proceedings of the 2008 7th International Conference on Mobile Business*, pages 221–230, Washington, DC, USA, 2008. IEEE Computer Society.
25. C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
26. M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proceedings of Pervasive Computing, Third International Conference, PERVASIVE*, Munich, Germany, 2005.
27. M. Duckham and L. Kulik. Location privacy and location-aware computing. In *Dynamic and Mobile GIS: Investigating Change in Space and Time*, 2006.
28. L. Fischer, S. Katzenbeisser, and C. Eckert. Measuring unlinkability revisited. In *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 105–110, New York, NY, USA, 2008. ACM.
29. J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX-SS'06: Proceedings of the 15th conference on USENIX Security Symposium*, Berkeley, CA, USA, 2006. USENIX Association.
30. J. Freudiger, M. H. Manshaei, J.-P. Hubaux, and D. C. Parkes. On non-cooperative location privacy: a game-theoretic analysis. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 324–337, New York, NY, USA, 2009. ACM.
31. J. Freudiger*, M. H. Manshaei*, J.-Y. Le Boudec, and J.-P. Hubaux. On the Age of Pseudonyms in Mobile Ad Hoc Networks. In *IEEE Infocom*, 2010.
32. J. Freudiger, M. Raya, and J.-P. Hubaux. Self-Organized Anonymous Authentication in Mobile Ad Hoc Networks. In *Conference on Security and Privacy in Communication Networks (Securecomm)*, pages 350–372, 2009.
33. J. Freudiger, R. Shokri, and J.-P. Hubaux. On the optimal placement of mix zones. In *PETS '09: Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*, pages 216–234, Berlin, Heidelberg, 2009. Springer-Verlag.
34. B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pages 620–629, Washington, DC, USA, 2005. IEEE Computer Society.

35. B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.
36. A. Gkoulalas-Divanis and V. S. Verykios. A free terrain model for trajectory k—anonymity. In *DEXA '08: Proceedings of the 19th international conference on Database and Expert Systems Applications*, pages 49–56, Berlin, Heidelberg, 2008. Springer-Verlag.
37. P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*, pages 390–397, Berlin, Heidelberg, 2009. Springer-Verlag.
38. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, New York, NY, USA, 2003. ACM.
39. B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.
40. B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *MobiSys '08: Proceeding of the 6th international conference on Mobile systems, applications, and services*, pages 15–28, New York, NY, USA, 2008. ACM.
41. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.
42. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171, New York, NY, USA, 2007. ACM.
43. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 177–189, New York, NY, USA, 2004. ACM.
44. L. Huang, H. Yamane, K. Matsuura, and K. Sezaki. Towards modeling wireless location privacy. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2005)*, pages 59–77, 2005.
45. L. Huang, H. Yamane, K. Matsuura, and K. Sezaki. Silent cascade: Enhancing location privacy without communication qos degradation. In *Security of Pervasive Computing (SPC)*, pages 165–180, 2006.
46. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: a distributed mobile sensor computing system. In *ACM SenSys*, 2006.
47. T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 246–257, New York, NY, USA, 2007. ACM.
48. P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719–1733, Dec. 2007.
49. H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Pervasive Services, 2005. ICPS '05. Proceedings. International Conference on*, pages 88–97, July 2005.

50. J. Krumm. Inference attacks on location tracks. In *In Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), volume 4480 of LNCS*, pages 127–143. Springer-Verlag, 2007.
51. J. Krumm. Realistic driving trips for location privacy. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*, pages 25–41, Berlin, Heidelberg, 2009. Springer-Verlag.
52. J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, 2009.
53. M. Li, K. Sampigethaya, L. Huang, and R. Poovendran. Swing & swap: user-centric approaches towards maximizing location privacy. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 19–28, New York, NY, USA, 2006. ACM.
54. L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov networks. In *IJCAI'05: Proceedings of the 19th international joint conference on Artificial intelligence*, pages 773–778, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.
55. H. Lu, C. S. Jensen, and M. L. Yiu. Pad: privacy-area aware, dummy-based location privacy in mobile services. In *MobiDE '08: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 16–23, New York, NY, USA, 2008. ACM.
56. J. Meyerowitz and R. Roy Choudhury. Hiding stars with fireworks: location privacy through camouflage. In *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 345–356, New York, NY, USA, 2009. ACM.
57. M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 763–774. VLDB Endowment, 2006.
58. A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *SP '09: Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, Washington, DC, USA, 2009. IEEE Computer Society.
59. M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: a generalization-based approach. In *SPRINGL '08: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 52–61, New York, NY, USA, 2008. ACM.
60. P. Papadimitratos, M. Poturalski, P. Schaller, P. L. and D. Basin, S. Čapkun, and J.-P. Hubaux. Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking. *IEEE Communications Magazine*, 46(2):132–139, February 2008.
61. A. Pfitzmann and M. Köhntopp. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, v0.32. Accessible through http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, 2009.
62. O. Riva and C. Borcea. The urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2), 2007.
63. P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *IEEE Symposium Research in Security and Privacy*, 1998.
64. K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, , and K. Sezaki. Caravan: Providing location privacy for vanet. In *The 3rd workshop on Embedded Security in Cars (ESCAR)*, 2005.

65. A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
66. R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A distortion-based metric for location privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 21–30, New York, NY, USA, 2009. ACM.
67. L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5), 2002.
68. C. Troncoso, G. Danezis, E. Kosta, and B. Preneel. Pripayd: privacy friendly pay-as-you-drive insurance. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 99–107, New York, NY, USA, 2007. ACM.
69. T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 547–555, April 2008.
70. T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*, pages 348–357, New York, NY, USA, 2009. ACM.
71. T.-H. You, W.-C. Peng, and W.-C. Lee. Protecting moving trajectories with dummies. In *Mobile Data Management, 2007 International Conference on*, pages 278–282, May 2007.
72. G. Zhong and U. Hengartner. A distributed k-anonymity protocol for location privacy. *IEEE International Conference on Pervasive Computing and Communications*, 0:1–10, 2009.

PseudoID: Enhancing Privacy for Federated Login

Arkajit Dey[1] and Stephen Weis[2]

[1] Massachusetts Institute of Technology, Cambridge, MA, USA 02139
[2] Google Inc., Mountain View, CA, USA 94043

Abstract. PseudoID is a federated login system that protects users from disclosure of private login data held by identity providers. We offer a proof of concept implementation of PseudoID based on blind digital signatures that is backward-compatible with a popular federated login system named OpenID. We also propose several extensions and discuss some of the practical challenges that must be overcome to further protect user privacy in federated login systems.

1 Introduction

Internet users often manage login credentials for many accounts across multiple web sites. This is both an inconvenience and a potential security risk, as users often resort to reusing passwords. Users also become accustomed to typing user names and passwords in many different interfaces. This can leave users more susceptible to phishing, that is, having their credentials stolen by imposter websites.

Issues with managing web login credentials helped motivate the creation of web single sign-on (SSO) systems. One SSO model is for users to have a single *identity provider* (IDP) for all logins. Arbitrary web sites may then become *relying parties* (RPs), who delegate logins to the identity provider. The IDP handles authenticating the user and attesting an identity back to the RP.

Some proposals, such as Windows Live ID or Facebook Connect, rely on a centralized identity provider. Other systems, such as OpenID, allow users to have identities from among a federation of identity providers. Federated login systems like OpenID offer more flexibility to end users, since they are able to choose among many identity providers. Large web mail providers like Yahoo, Google, and MSN have all adopted OpenID [22, 19, 21] and are already capable of serving as identity providers for hundreds of millions of users.

While federated login systems like OpenID may streamline logins, they also create risks to user privacy. The core problem in both centralized and federated login systems is that all user logins to relying party web sites must flow through an identity provider. A user's identity provider can easily link together the various websites that the user visits. An identity provider could, for example, release data about which sites users visited without user consent.

In a federated system with many providers to choose from, users could avoid identity providers that abused privacy and use reputable firms. Unfortunately, honest identity providers may still be compromised and leak logs, or otherwise be compelled to reveal logs. Besides simply revealing which sites a user visits, identity providers often reveal personal information about users through extensions like OpenID Attribute Exchange (AX) [16] or Simple Registration (SREG) [17]. The goal of this exchange is typically to pass information like an email address, real name, or birth date from an identity provider to a web site. Automatically obtaining these data can greatly streamline the user sign-up process for relying parties.

Although most identity providers will prompt users asking whether they want to reveal this information, identity providers could reveal whichever data they want to a relying party. Thus, there is no way for a user to *selectively disclose* certain properties (e.g. age, gender, etc.) about themselves to a relying party. Much work has gone into developing cryptographic schemes for selective disclosure [4, 5, 3, 2], but these have yet to be adopted in practice.

In this paper, we outline a privacy-preserving federated login system called PseudoID and offer a proof of concept implementation as a pseudonymous OpenID provider located at `http://pseudoid.net`. The system utilizes blind signatures [7] as part of a blind signature service. This service allows users to generate a pseudonym that can be used to login to relying parties, but cannot be linked to their true identity. We also propose extensions based on zero-knowledge proofs [11] to support selective disclosure of user properties.

2 Federated Login Overview

Web users who want to use a particular website most often authenticate themselves directly to the site by entering a user name and password as in Figure 1. Maintaining many sets of user credentials across different sites creates a burden for the user and can lead to password reuse. Websites, in turn, are burdened with the unwanted responsibility of creating accounts, storing credentials, and authenticating users. Account creation, or on-boarding, is often a large barrier to signing up new users. It is not uncommon for over half of sign-up attempts to be abandoned.

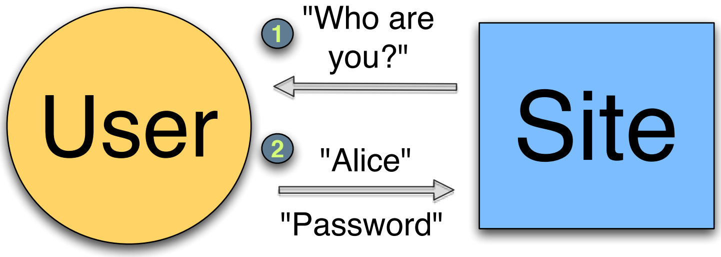


Fig. 1. A typical web login system where users log into websites by entering site-specific credentials.

Federated login systems, on the other hand, extract authentication as a service in its own right. Just as websites rely on third-party services for traffic analysis, CAPTCHA verification, or file hosting, they can also rely on separate services for authentication.

Federated login adds a third party to the interaction between the user and the website: the *identity provider* (IDP). Instead of authenticating herself to the website directly, the user authenticates herself to the IDP. The IDP then returns a user identifier to the website. Thus, the website is often referred to as the *relying party* (RP) since it relies on the identity provider for authentication.

Federated login alleviates the need for websites to store user credentials, making them less desirable targets for attackers who want to hijack user accounts. The user benefits from federated login too. Instead of managing separate login credentials for every website he wants to use, the user can just log into a single identity provider.

Systems like Facebook Connect and OpenID 2.0 are able to offer one-click logins for relying parties, which greatly simplifies the login process. For example, Plaxo, a social networking and address book site, performed a two-click OpenID login experiment where 92% of users successfully completed registration after starting the sign-up process [13]. In contrast, on-boarding abandonment rates of 50% are common for many websites.

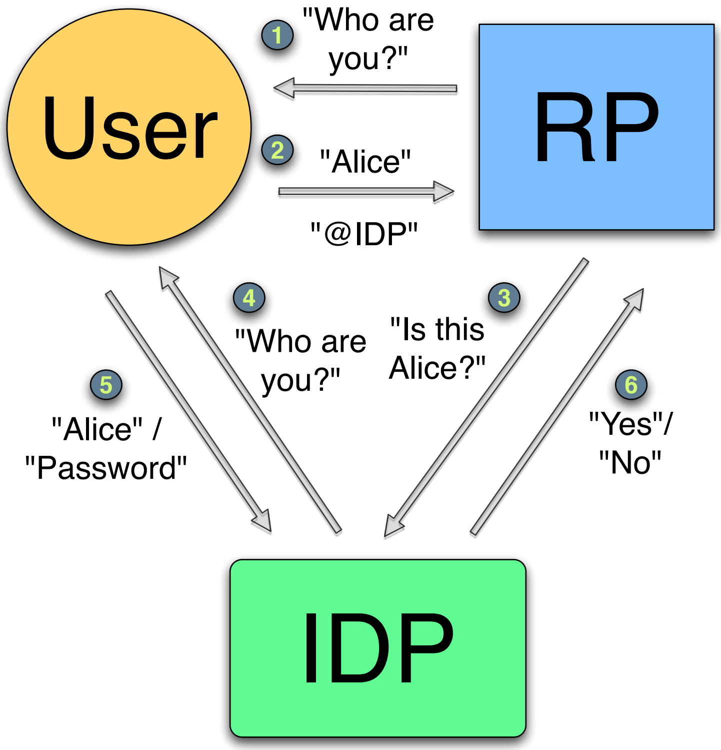


Fig. 2. A federated login system: (1) The RP prompts the user for an identity. (2) The user enters an identifier that specifies an IDP. (3) The RP requests that the IDP authorize the user. (4) The IDP prompts the user for her credentials. (5) The user presents her credentials to the IDP. (6) The IDP returns the result of the authorization to the RP.

Accordingly, federated login systems are being adopted by a growing number of Internet sites, particularly by large web mail providers and social networks. Several different federated login technologies have arisen over the years, such as Microsoft Passport (now Windows Live ID), OpenID, Facebook Connect, and SAML.

However, popular federated login systems have generally been designed without privacy as a primary concern. Subsequently, current widely-used federated login systems could put sensitive user data at risk. The problem of user privacy is indeed magnified in federated login systems since identity providers act as stewards of user data for multiple websites. This not only makes identity providers more appealing targets to attackers, but also more likely to be subpoenaed for user records.

2.1 OpenID: A web-based federated login system

OpenID is a popular federated login system that we focus on for a proof of concept implementation. In OpenID, users can claim identifiers in the form of URIs. To login to a website that supports OpenID, the user enters his OpenID URI and the relying party redirects him to his identity provider's page. The identity provider authenticates the user through its choice of authentication system (e.g. passwords, smart cards, etc.) and then returns the user to the relying party with either a positive or negative assertion that the user owns the claimed identifier. If the relying party receives a positive assertion from the identity provider, it may allow the user to enter the site under the name of the claimed identifier.

With the advent of OpenID 2.0 [15], the protocol also began to support the concept of *directed identity* [6] or *private digital addresses* through a new feature called *identifier select* [18]. This allows the user to just specify the URI of his identity provider instead of claiming a personal identifier when logging into a website. The site then redirects the user to the identity provider as before, but the identity provider now has the opportunity to select an identifier for the user. Upon successfully authenticating the user, the identity provider returns the selected identifier to the site.

This allows the identity provider more flexibility in selecting identifiers for its users. For example, the identity provider may decide to return a different identifier for the same user for different relying parties in order to implement true directed identity as defined in Kim Cameron's Laws of Identity [6]. Indeed, some OpenID providers like Google do return a per site unique identifier rather than a globally unique identifier for its users.

2.2 Privacy Concerns in Federated Login

In federated login systems, users entrust identity providers to manage their identity, so privacy concerns may seem relatively minor. After all, in OpenID and most other major single sign-on systems, a malicious identity provider could easily impersonate users to relying parties. However, even if an identity provider is not corrupt, there are privacy concerns for *honest, but retentive* providers who may reveal user data due to a security breach or a legal subpoena.

The core privacy issue with widely-deployed federated login systems is that a user's identity can be correlated with the sites she logs into. For example, OpenID identity providers will authenticate users, then redirect them back to a relying party. This makes it trivial for an identity provider to know all the web sites a user logs into. The same is true for Live ID or Facebook Connect.

One might develop a different federated login flow where a user acted as an intermediary between relying parties and identity providers. The user could avoid passing any information about the specific relying party to the identity provider. In this case, the identity provider might return an anonymized identifier via the user to the relying party. However, if the provider colluded with the relying party, they could link the user's real identity with the account on the relying party.

Alternatively, an identity provider could abstain from logging, could try to anonymize or delete identifying information, or could simply destroy logs completely. Logs are retained for many valid reasons including analytics, diagnostics, and security auditing. In practice, abstaining from logging is often not a viable option.

Removing or anonymizing identifying information ex post facto is one option, but has proved difficult in practice. Supposedly anonymized logs released by AOL [1] and Netflix [14] were both de-anonymized to some extent. An identity provider would need to be vigilant and thoroughly scrub logs to remove identifying data. They would also need to ensure that identifying data was not being unintentionally logged by an unrelated service or a different layer of the stack.

Another issue is that both logs anonymization and destruction may be subject to data retention laws that specify minimum retention periods [10]. Identity providers may be legally compelled to collect identifying data and retain it for a minimum period. In some jurisdictions, authorities have broad powers to rapidly seize these data, often without user knowledge.

Given these considerations, we will focus on privacy in a setting where: (1) identity providers are unable to assure that logs are not retained, and (2) identity providers may be compelled to reveal logs at some time.

There are several real-world risks where *honest, but retentive* identity providers may threaten user privacy. One risk is simply if the provider were compromised and logs were leaked to an attacker. Another risk is for providers operating in jurisdictions where logs may be seized without due legal process.

Identity providers may have an interest in *not* being able to link a particular user's identity to logins on a relying party. An identity provider may want to provably show that they cannot link logins on a relying party to a particular user. With this in mind, we informally define what it means for an identity provider to be *private* in Section 3. Section 4 proposes a practical system that meets this definition.

3 Properties of Private Federated Login

For the scope of this paper, we are going to focus on privacy in federated login systems with an OpenID-like login flow illustrated in Figure 2. The relevancy to other types of federated login systems may vary.

We assume that identity providers have a set of users that can be thought of as "real" identities. Users may possess credentials which are presented to identity providers for authentication. If the credentials are valid, identity providers will return some identifier to a relying party. This identifier may be of any form, i.e. a "real" user name, a pseudonym, a value derived from the credentials, or even a random value. In order to contrast these different behaviors, we will first state several informal properties.

Definition 1 (One-wayness). *An identity provider is one-way if given a specific identifier, attackers have no significant ability to cause identity providers to return that value.*

One-wayness means that users actually "own" their identities and people cannot imitate them on relying parties. For example, one cannot typically login with a specific user name without knowing that user's password. A trivial example of an identity provider that is *not* one-way is one where an identity provider will assert any identity without authorizing the user. We'll refer to this as the "Yes IDP". Users of a Yes IDP could log into relying parties with arbitrary identities, but would not be able to prevent other people from using the same identities.

Definition 2 (Consistency). *An identity provider is consistent if users may present credentials that will return the same identifier over multiple sessions.*

The consistency property means that users can have long-lived identities on relying parties. That is, users can log in as the same identity to a relying party any number of times. A trivial inconsistent identity provider would be one which returned a random value for each login, or a “Random IDP”. Users of a Random IDP would be anonymous on relying parties and could not be linked to their real identities, but would not be able to establish long-lived accounts on relying parties.

Definition 3 (Unlinkability). *An identity provider is unlinkable if given a transcript of an authentication event and a set of users, an attacker has no significant advantage in distinguishing the user being authenticated.*

Unlinkability is intended to capture the notion that an attacker who obtains access logs from an identity provider and relying party should not be able to tell which “real” user was logging in. OpenID identity providers are generally linkable in practice, although there are exceptions. An attacker obtaining a user’s credentials or identity provider access logs would be able to trivially see which user was associated with a particular identity on a relying party.

Note that this property is not specific to OpenID or a flaw in the OpenID protocol. Instead, it’s an artifact of how real world identity providers typically authenticate users: with user names and passwords. For example, an attacker may observe “Alice” authenticate herself to an identity provider and the identity “Bob” returned to a relying party; linking her real identity to her pseudonym.

Thus, an unlinkable relying party must not require any identifying information about their real users during the federated login protocol. The Yes and Random IDPs mentioned before are in fact unlinkable, but are not practical in many use cases since they are respectively not one-way or consistent. A practical unlinkable system must be both one-way and consistent. We will present such a system in Section 4.

4 PseudoID: A privacy-preserving federated login system

PseudoID is designed to be a one-way, consistent, and unlinkable federated login system. It consists of a token service used during setup, and a private identity provider used for sign-ons. The user has an account with the token service, which may be a persistent, “real” identity like an email address. During setup, the user logs on to the token service using a familiar authentication scheme, such as entering a user name and password.

The user then requests an access token from the token service that is bound to a desired pseudonym. When logging into a relying party, the user presents this token to an identity provider. The identity provider will verify the authenticity of the token and return the user’s pseudonym to the relying party.

To be unlinkable, the access tokens must be generated such that even if both the token service and identity provider are compromised, the user’s “real” identity with the token service cannot be linked to their pseudonyms on different relying parties. PseudoID achieves this property by employing *blind signatures*.

4.1 Blind Signatures

Traditional public key digital signature schemes [9] consist of a private signing function S known only to the signer and a public verifying predicate V. Then for any message m that is provided to the signer to be signed, a verifier can check that $V(m, S(m))$ is true. It is infeasible to produce the signature $S(m)$ without knowledge of the signing function S.

Blind signature systems [7] augment this traditional scheme with a blinding function B and its inverse unblinding function B^{-1}, such that $B^{-1}(S(B(m)) = S(m)$ and both functions are known only to a user getting a message signed.

In a blind signature scheme, the user wishes to obtain a signature $S(m)$ on some message m without revealing the contents of m to the signer. To do so, the user sends the blinded message, $B(m)$, to the signer that leaks no information about m. The signer then signs the blinded message and returns $S(B(m))$ to the user. Finally, the user unblinds this signed message to obtain

$$B^{-1}(S(B(m))) = S(m),$$

a valid signature on m that can be publicly verified.

One example of a blind signature system is Chaum's RSA blind signatures. In a standard RSA digital signature system, the public parameters are a modulus n and an exponent e. Only the signer knows the private exponent d. To blind a message m prior to sending it to the signer, the user multiplies it by a random blinding factor r to produce $B(m) = mr^e$. The signer signs $B(m)$ to produce

$$m^d r^{ed} \equiv m^d r \pmod{n}$$

by Euler's theorem. Since the user can compute r^{-1}, he can unblind the returned signature to obtain

$$m^d \pmod{n},$$

a valid signature on the original message m.

4.2 Blind Token Service

PseudoID employs a blind signature service (BSS) or *blind signer* that generates blinded access tokens. These tokens are redeemed with an identity provider and used to derive identifiers that are returned to relying parties. This setup phase is outlined in Figure 3.

During a setup phase, the user will visit the blind signer and login to an existing account. The user then selects a pseudonym that they want to use on a relying party and a secret value. This pseudonym and random secret value are bundled into an access token that the blind signer will sign. That is, the user will prepare a token $T = (pseudonym, secret)$.

To prevent the signer from being able to link a user with her pseudonym, the user first blinds the token $B(T)$ before sending it to the blind signer. The blind signer will sign this token without knowing its contents and return it to the user as $S(B(T))$. Upon receiving the singed token back from the service, the user unblinds it to obtain a signed token $S(T)$ that contains the user's chosen pseudonym and secret value. Note that the blind signer will not see the user's pseudonym or secret value in the clear; it will only see the blinded token.

4.3 Private Identity Provider

The identity provider relies on the blindly signed tokens to be able to authenticate users without forcing them to reveal their identity. When a user is redirected to her identity provider by a relying party, the provider checks whether the user has an access token that has been signed by the blind signer.

The signature on the token may be either publicly verifiable or privately verifiable. In the former case, the identity provider can verify the signature on the access token using the blind signer's public key. In the latter case, the identity provider could send the token to the blind signer and ask them whether they signed it. The sign-on process using an access token in the publicly verifiable case is illustrated in Figure 4. If the

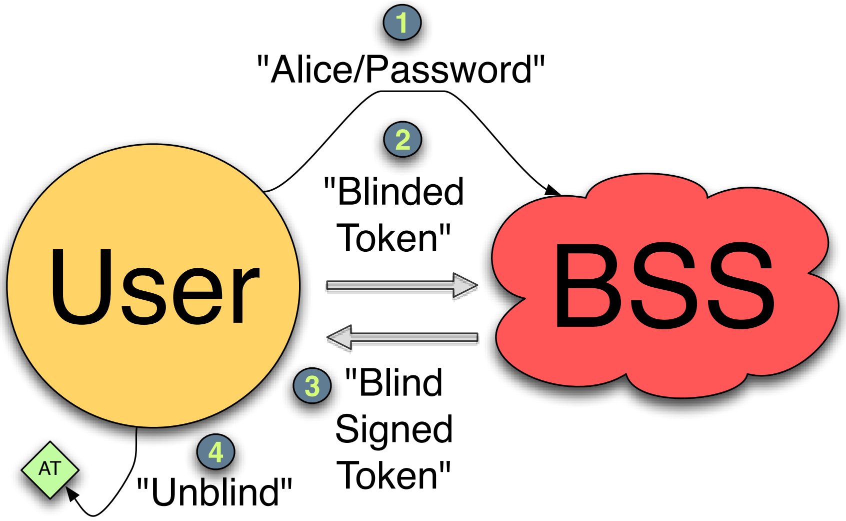


Fig. 3. Blind Signer Setup: (1) User first authenticates herself to the BSS normally. (2) Then the user sends the BSS a blind token to sign. (3) The BSS signs the token and returns it. (4) The user unblinds the blind signed token to obtain a valid, untraceable access token (AT).

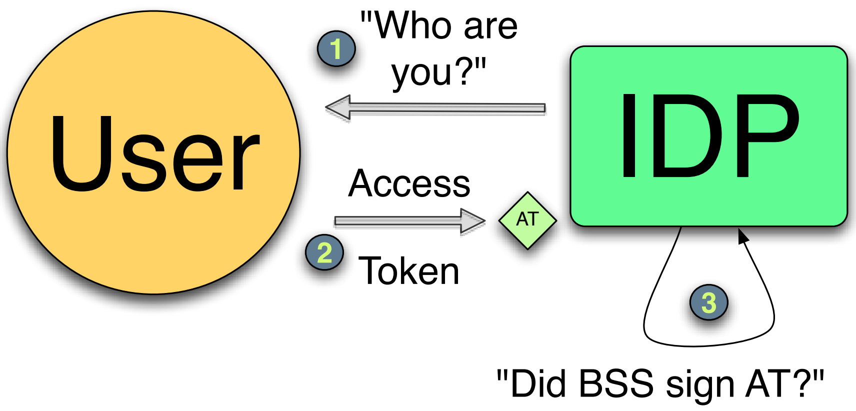


Fig. 4. Identity Provider Sign-on with Blind Signed Access Token: (1) IDP asks users to authenticate (2) User supplies access token rather than true identity or credentials (3) IDP verifies whether BSS signed the token using BSS's public key.

access token is valid, the provider is only assured that the user has been authenticated by the blind signer. Thus the provider knows that the user is a valid user of the blind signer, but does not know which user.

Given a valid access token from the user, the identity provider will compute a one-way, collision-resistant function F on its value and return that as part of an identifier to the relying party. For instance, if the access token contains $T = (pseudonym, secret)$ the identity provider could assert $(pseudonym, F(T))$ as the user's identifier to the relying party. Given the properties of F, attackers will not be able to invert the value $F(T)$ to obtain T or find a value T' such that $F(T) = F(T')$.

4.4 Properties of PseudoID

We informally argue that PseudoID can meet the properties described in Section 3.

Claim (1). **PseudoID is one-way** The access token T that is presented by the user to the IDP will contain a random secret value. The IDP will then compute a one-way, collision-resistant function of that token $F(T)$ that is presented as part of the user's identity. An attacker will only be able to cause an IDP to return that same identity if they are either able to learn the secret, invert $F(T)$, or find a colliding token value T' such that $F(T) = F(T')$. Thus, attackers will have no significant ability to cause the IDP to return a given identity.

Claim (2). **PseudoID is consistent** The private identity provider will always return the same identifier for a given access token. Its output is completely deterministic, and thus consistent.

Claim (3). **PseudoID is unlinkable** Suppose an attacker has all the logs of the blind signing service, all blinded access tokens, and all unblinded access tokens used to authenticate to the IDP. The attacker will win if they are able to link a specific blinded access token to a specific unblinded access token, i.e. revealing a user's identity. However, this is precisely what blinding prevents. PsuedoID uses blinding in exactly the same fashion as Chaum's untraceable payment scheme [7]. Thus, PsuedoID is unlinkable based on the properties of the blind signature scheme.

4.5 OpenID with PseudoID

PseudoID is practical to implement in a web setting, such as for OpenID. We have implemented a proof of concept blind signing service and identity provider available at `http://pseudoid.net`. The proof of concept blind signer is implemented as a web service. Users visit the blind signer and prepare a blinded token signed with JavaScript (see Section 5 for a discussion of some security caveats).

The blind signer will blindly sign this value and return it to the user. This value is unblinded and stored as a cookie in the user's browser. This cookie will be set on the identity provider's domain. The identity provider itself may be slightly modified from existing OpenID providers. It simply needs to read an access token from a cookie on the user's browser, verify a signature on it, and return the pseudonym it contains to a relying party. From the user's perspective, this eliminates the need to retype a user name and password on an identity provider.

PseudoID identity providers are fully compatible with existing OpenID relying parties. Existing relying parties do not have to change anything about their current OpenID flow in order to be able to accept users from private identity providers. From the perspective of the relying party, a private identity provider is indistinguishable from a regular provider. A private provider simply uses a different authentication mechanism than most other identity providers, but it still participates in the same federated login flow outlined in Figure 2.

5 Extensions and Future Work

Our goal in creating a proof of concept implementation of PseudoID has been to demonstrate the feasibility of creating and using pseudonyms within the framework of existing federated login technologies. We've strived to keep our prototype as minimal as possible and accordingly opted for simplicity over optimality in several important, but tangential, design choices. In this section, we acknowledge some of the limitations of these choices and the extensions we can make to overcome them and expand on our simple prototype. Additionally, we discuss some remaining privacy issues that plague federated login systems in general.

5.1 Token Storage

In PseudoID, a user's access tokens may be thought of as the keys to their identity. This key can be used to unlock or divulge aspects of a person's real identity (e.g. whether they are over 18, are male, have a valid U.S. address, etc...) without revealing their complete identity (i.e. this is John Doe with Social Security number 123 who lives in CA). The key is instead associated with the person's pseudonym or chosen persona.

Determining how to effectively store these tokens is an important prerequisite to deploying a full implementation of a private federated login system such as PseudoID. But it is still secondary to the question of how to create and use such tokens within existing technologies which is what we have chosen to focus on instead. Thus, in the interest of simplicity, we have deferred answering the question of token storage to future research.

As such, our proof of concept implementations uses a non-ideal, but passable, medium for token storage: browser cookies. Some of the drawbacks of this approach are that some users may clear cookies frequently or use a private browsing mode which prevents their creation. Additionally, tokens stored as cookies are not easily transferable to a user's other machines. This makes it harder to extend a user's pseudonym across multiple machines.

We should note, however, that the issue of transferability is not simply an artifact of our implementation, but rather an unavoidable side effect of our goal of unlinkability. If we truly do not want to have pieces of our identity be linkable via a central repository, we will need to handle the distribution of the tokens by some other mechanism such as by individually installing the tokens on each device we wish to use it on.

5.2 Token Accessibility

There is also the issue of who should be able to view a user's tokens. If the user just hands them out to any party that wants to authenticate him, he enables them to impersonate him with the tokens. For our proof-of-concept implementation, we adopted the simplest stance that the user only wants to show his tokens to his private identity provider. In federated login systems, the user is already trusting his identity provider so this adds no extra layer of trust.

Since we are storing tokens in cookies, we run up against the security issue with setting a cookie on the identity provider's domain so that only they can read it. By the *same-origin policy*, JavaScript executing on one domain cannot set a cookie on another.

The proof of concept implementation must make a call to the identity provider to set an unblinded cookie during the setup phase. This means that access logs on the blind signer and the identity provider could be joined to correlate the user's login with the value that was set as a cookie.

To rectify this situation, PseudoID could benefit from a more flexible browser storage model than cookies, and the means to pass messages from one domain to another using the browser as an intermediary. Several features proposed in HTML 5 may help facilitate this [12].

Alternatively, we can side-step the cross-domain communication issue entirely by accepting some additional complexity such as encrypting our tokens with our provider's public key or enabling the user to prove his possession of the token via zero-knowledge proofs.

5.3 Simplified Cryptography in the Browser

Modern browsers are equipped with support for a broad range of cryptographic functionality to support SSL/TLS. Yet, it is difficult for a typical web application to make use of it. In the case of PseudoID, server-side JavaScript was the most convenient method to blind and unblind tokens. But using JavaScript is both inefficient and insecure. Basic cryptographic functionality has to be reimplemented in JavaScript and interpreted, rather than using the native cryptographic libraries already available in the browser.

There is also a question of where the JavaScript code comes from. If it is hosted on a server, it may later be substituted with malicious code without the user's knowledge. For example, if the host of the JavaScript code were compromised, an attacker could inject code to leak the user's identity.

Browser cryptographic support could be made available through a browser plug-in or extension, but this is a barrier to adoption and difficult to support on multiple platforms. PseudoID and many other applications could benefit from a simple, standardized, and cross-platform API to client-side cryptographic services.

5.4 Orthogonal Privacy Risks

Even if logins are private, users can still be tracked in other ways, by IP addresses or cookies. Anonymization on the network level is an independent risk and may be mitigated by the use of web proxies or anonymous browsing technology like Tor [20].

But there is still the risk in the proof of concept system that a malicious blind signer or identity provider may try to set tracking cookies on the user's browser while they are logged in with their real identity. To mitigate this risk, a user would need to scrub all cookies except their access token from their browser which would be impractical from a usability standpoint.

But, realistically, the user is already accepting this risk by trusting their identity provider as they do whenever they use any federated login system. If his identity provider is malicious, the user has larger concerns such as whether the provider is impersonating them to relying parties. Thus, as we have before, we will continue to assume an honest provider which renders this concern moot.

5.5 Selective Disclosure

In the current version, PseudoID access tokens contain a user-selected pseudonym and a random nonce. Tokens do not contain any meaningful semantics nor any properties of the user's real identity. By using zero-knowledge proofs, one may extend the blind signer to support selective disclosure. There is a broad range of literature on this topic [8, 4, 5, 3, 2].

The basic idea is that users will engage in a zero-knowledge proof with the blind signer. They will prove that the contents of blindly signed messages convey some meaningful data or have a proper semantic form. For example, the user may prove that a blindly signed message contains a bit value representing "Is this user over 18 years of age?" that is true for their real identity, without revealing any other information about the message. Another example use is obtaining a token with an expiration time in it. The user would prove to the blind signer that a blinded message contains a valid expiration time, without revealing any other knowledge of the message.

By allowing tokens to have these types of semantics, identity providers will be able to offer more fine-grained access policies. In the simple PseudoID system, identity providers can only verify a signature on a token – all they learn is that a blind signer signed it at some point in time. If tokens had semantics, they could, for instance, only allow access to users with tokens that were issued within some time period.

From a web-based implementation standpoint, performing zero-knowledge proofs in the browser requires better support for both cryptography and storing persistent values. While it is possible to implement zero-knowledge proof systems with a JavaScript and cookie approach, this would have the same limitations that the proof of concept PseudoID implementation has.

6 Conclusion

We have identified a key privacy concern with federated login systems, that they allow identity providers to link a user's identity and track them across visits to multiple websites. To address this concern, we have proposed a model of a private federated login system that preserves the convenience of existing federated login systems while adding the property of unlinkability.

PseudoID is a proof-of-concept implementation of this model that demonstrates the feasibility of acheiving pseudonymity using blind signatures within existing federated login technologies like OpenID. It is a basic, minimal prototype that serves as the foundation for future work in establishing true selective disclosure in federated login. It also identifies several technology improvements needed to make private federated login systems more practical: better browser support for cryptography and persistent storage.

References

1. Barbaro, M., Zeller Jr., T.: A face is exposed for aol searcher no. 4417749. New York Times (August 9, 2006), http://www.nytimes.com/2006/08/09/technology/09aol.html
2. Camenisch, J., Groß, T.: Efficient attributes for anonymous credentials. In: Proceedings of the 15th ACM conference on Computer and communications security. pp. 345–356 (2008)
3. Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact e-cash. In: EUROCRYPT. pp. 302–321 (2005)
4. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT. vol. 2045, pp. 93–118 (2001)
5. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: CRYPTO. pp. 56–72 (2004)
6. Cameron, K.: Laws of identity, http://www.identityblog.com/?p=354
7. Chaum, D.: Blind signatures for untraceable payments. In: Advances in Cryptology: Proceedings of CRYPTO. pp. 23–25 (1982)
8. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. Communications of the ACM 28(10), 1030–1044 (1985)
9. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory IT-22(6), 644–654 (November 1976)
10. Eu data directive (1995), http://www.cdt.org/privacy/eudirective/EU_Directive_.html
11. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal of Computing 18(1), 186–208 (February 1989)
12. Hickson, I., Hyatt, D.: Html5 (December 21, 2009), http://dev.w3.org/html5/spec/Overview.html
13. Kirkpatrick, M.: Comcast property sees 92% success rate with new openid method. ReadWriteWeb Article (February 10, 2009), http://www.readwriteweb.com/archives/comcast_property_sees_92_success_rate_openid.php
14. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy. pp. 111–125 (2008)
15. Openid authentication 2.0, http://openid.net/specs/openid-authentication-2_0.html
16. Openid attribute exchange 1.0, http://openid.net/specs/openid-attribute-exchange-1_0.html
17. Openid simple registration extension 1.0, http://openid.net/specs/openid-simple-registration-extension-1_0.html

18. Recordon, D., Reed, D.: Openid 2.0: A platform for user-centric identity management. In: Proceeedings of the second ACM workshop on Digital identity management. pp. 11–16 (2006)
19. Sachs, E.: Google moves toward single sign-on with openid. Blog (October 29, 2008), http://googlecode.blogspot.com/2008/10/google-moves-towards-single-sign-on.html
20. TOR: The onion router project, http://www.torproject.org
21. Windows live id becomes an openid provider. Blog (October 27, 2008), http://winliveid.spaces.live.com/Blog/cns!AEE1BB0D86E23AAC!1745.entry
22. Yahoo! announces support for openid. Press Release (January 17, 2008), http://yhoo.client.shareholder.com/press/releasedetail.cfm?releaseid=287698