**University of Konstanz**
**Distributed Systems Group**
**Technical Report KN-06-03-2009-DISY-01**
**March 2009**

# PathForge:: Faithful Anonymization of Movement Data

by

Sebastian Kay Belle  Marcel Waldvogel

## ABSTRACT

For most mobile networks, the provider needs the current position of their users to provide efficient service. The resulting motion data is not only an invaluable source for analyzing traffic or flow patterns, but also for tracking an individual's whereabouts, even without his knowledge. Today, many carry at least one mobile networked device (cell phone, UMTS-enabled netbook, ...) with them wherever they go, day or night. The resulting motion data can be used to reveal the most intimate details of our lives, making this information extremely privacy sensitive. Nevertheless, little is done to prevent the collection of this information, on the contrary, tracking features and data are being actively collected and even marketed. In this paper, we present *PathForge*, a lightweight solution, which not only fulfills the provider's efficiency requirement, but continues to allow flow pattern analysis, yet provides full privacy for the user when not actively involved in a call.

# 1 Introduction

Nowadays we are in the comfortable position to access the Internet from nearly anywhere as long as our mobile phone is connected to the provider's cell phone network. We can browse the Web, write emails and other types of text message, or simply make a phone call. However, most of the time we simply carry our mobile devices around waiting for incoming calls, messages, and e-mails or until we feel an urge to initiate such a connection. We are used to take our cell phone, or any other mobile device we can think of, with us all the time, though, most of us are not aware of the consequences this behaviour implies: We are continuously being tracked by our service provider.

Carriers initially did not start tracking for profiling the movement of its users, but, simply because of scalability and efficiency issues, which are the very reason our current cellular networks have been able to accomodate the increased proliferation of mobile devices by orders of magnitude. Whenever a data packet or signalling message such as call setup needs to be routed towards a user's cell phone, the carrier needs at least a very good guess about the location of the user to avoid flooding the network.

Today, operators on the lookout for additional income, are offering user tracking as a service to end users, service providers, and government agencies. Even though law enforcement agencies have been using the pretense of terrorist threats to gain more and more access to formerly private information, this has actually been going on for over a decade [NYT98]. Today we are willingly – however, mostly unaware – utilizing new techniques embedded in modern cell-phones (e.g. GPS) to alleviate continuous tracking by these official institutions [CN08]. An actual lawsuit filed with the German Constitutional Court against telecommunication data retention[Sta08] presents evidence that the impact of data retention does not result a significant increase in the prevention or solution rate of criminal activity in that governments want us to believe when giving up more and more of our privacy.

*Location-based services* (LBS)[MAHP03] disclose a multitude of possibilities to facilitate powerful applications making life easier or safer for users. However, LBS opens up threats that also endanger the privacy of people as many users unconcernedly and unaware submit their location information to untrusted services. Even though these service providers do not actively misuse these information, a malicious third party may get access to this information[USA02][FN04]. New services even promote these threats[BRN09].

Krumm and Horvitz[KH06] have shown that destinations of people can be extrapolated with a high probability given only a part of their path, even if the person has never visited the destination before. Their results show a median prediction error of a destination estimation that comes close to three kilometers after a trip fraction of 0.5 for the *open-world model*. Thus, tracking a person for about half of its trip is enough to get a good estimation about its potentially sensitive destination.

In this paper we propose a novel idea – we will refer to as *path forging*[1] – that addresses

[1]Webster's Revised Unabridged Dictionary (1913) – *Forge* (Page:585):: (2) *To form or shape out in any way; to produce; to frame; to invent*, (4) *To make falsely; to produce, as that which is untrue or not genuine; to fabricate; to counterfeit, as, a signature, or a signed document*

| | **Path Cloaking** [HGXA07] | **Personalized** [GL05] | **AnonySense** [KTC+08] | **Peer-to-Peer** [CML06] | **Privé** [GKS07] | **PathForge** |
|---|---|---|---|---|---|---|
| **Accuracy** | • | ◦ | ◦ | *na* | *na* | • |
| **Low Density** | • | – | • | • | • | • |
| **Autonomous** | • | – | • | – | – | • |
| **Service is::** | | | | | | |
| **→ centralized** | • | • | • | – | ◦ (Certificate) | – |
| **→ decentralized** | – | – | – | • | • | – |
| **→ client-based** | – | – | • | – | – | • |
| **Dynamic Adaption** | – | ◦ | ◦ | ◦ | ◦ | • |
| **A-Priori Knowledge** | *na* | *na* | **!** (history) | *na* | *na* | *na* |

**Legend**:: • Available; ◦ Partially; – Not Available; ! Required; *na* Not Applicable

Table 1: Spatial cloaking techniques utilizing $k$-anonymity compared to PathForge

the following issues:

- How to obtain accurate motion data, e.g. for traffic and flow analyses, whilst impeding malicious parties from inferring sensitive information from this data.
- Minimizing impact on the operator's infrastructure or efficiency.
- How to embed this into the carriers' existing infrastructure while leaving control and implementation of the privacy-preserving system to the client systems.

# 2 Related Work

Most of the techniques proposed to protect user privacy in terms of her location are based on $k$-anonymity [SS98, Swe02] that protects sensitive data from inference by generalizing a dataset so that the individual entries cannot be distinguished from at least $k-1$ other entries in the dataset. In the context of location hiding $k$-anonymity guarantees a defined degree of anonymity for the users in the system by spatial cloaking algorithms. In general, spatial cloaking algorithms can be divided into (1) centralized spatial cloaking systems and (2) decentralized spatial cloaking systems.

## 2.1 Centralized Spatial Cloaking

Gedik and Liu[GL05] propose the *Clique-Cloak* algorithm. *Clique-Cloak* takes the anonymity level ($k$-level), the spatial tolerance, and the temporal tolerance – as chosen by the user – as input to alter the query submitted to an LBS provider. The queries are altered such that several queries can be grouped together depending on the parameters passed. The queries in these groups are indistinguishable from each other in terms of the possibility to infer sensitive information about the users. This technique protects a user's privacy, however, it

relies on a complex infrastructure has a high computation overhead and does not guarantee high accuracy.

Hoh and Gruteser [HG05] introduced privacy by *path confusion* based on $k$-anonymity. They achieve good accuracy in terms of the tracked location while maintaining user privacy. The problem with path confusion is that anonymity is not guaranteed for sparsely populated areas. Therefore, Hoh et al. [HGXA07] extend the work on path confusion by introducing a *privacy metric* that reflects the probability that an adversary can correctly follow a trace. Based on this metric Hoh et al. remove points from the set of tracked users. Thereby, tracking information in sparsely populated areas will be removed as long as single individuals can be identified and tracked. This leads to good accuracy while preserving the privacy of the user even in sparsely populated areas.

Kapadia et al. [KTC+08] propose an interesting enhancement to centralized spatial cloaking. They divide their system into two layers: (1) They apply *statistical k-anonymity*, similar to local blurring, at the user level before any report is sent to the system, thus, already protecting user privacy from the system itself. *Statistical k-anonymity* is based on the Voronoi tessellation of local regions that are aggregated subsequently according to the history of users that visited the specific area. However, this requires the system to have *a priori* knowledge about visiting devices. (2) In the second level the system creates $l$-anonymous reports by aggregating $l$ different reports into a report group before submitting them to the application that performs the data analysis.

We agree with of Kapadia et al. to apply spatial cloaking at user level, however, in contrast to them, our approach does not need to include a (centralized) system layer as our system design is able to use the established infrastructure of a carrier – with only minor modifications to the standard protocols as discussed in section 3 – to facilitate easy assimilation of the system.

The latter have the requirement that you need to trust your every movement to the central authority, which is what *PathForge* clearly tries to avoid. This drawback also led others to the creation of decentralized spatial cloaking systems, as discussed next.

## 2.2 Decentralized Spatial Cloaking

Chow et al. [CML06] address the security threat introduced by a centralized service utilizing peers in the vicinity to form local, $k$-anonymous groups that query the LBS as one. Chow et al. require the user to specify his intended privacy level by two parameters, $k$ and $A_{\min}$, where $k$ is the intended anonymity level and $A_{\min}$ the minimal resolution of the area to look up for peers to form a group. However, this technique aims towards privacy preserving LBS queries, simplified, Chow et al. use spatial cloaking while the user submits a query to the database, though, not in the meantime when the user's location is tracked by her carrier.

Ghinita et al. [GKS07] introduce the idea of utilizing space-filling curves to guarantee query anonymity. They define an infrastructure utilizing a centralized certification server to certify the integrity of clients in the system. Every certified client defines his intended level of anonymity, similar to the approach of Chow et al., to form $k$-anonymous groups.

Subsequently, a query to an LBS uses the minimal bounding box of a $k$-anonymous group to locate the point of interest, passing a set of answers to the initiating user. Again, this system does only guarantee to protect the location of the user at the time of submitting the query, but, does not make any guarantees about the movement data of a user in general. Furthermore, the system of Ghinita et al. requires the LBS to support region queries instead of simple point queries.

Though decentralized spatial cloaking systems omit the single point of failure inherent to centralized systems, their main focus lies on protecting user privacy during query processing. Furthermore, a malicious node can easily infiltrate the system and infer knowledge about the users in a $k$-anonymous group as protecting the privacy within such a group remains somewhat fuzzy in the discussed systems.

Table 1 illustrates the capabilities of existing spatial cloaking systems and the abilities of our proposed *PathForge* system design. The table is divided into the most important aspects of spatial cloaking systems, namely: (1) How accurate the continuous tracking data is while preserving user privacy, (2) if the system can preserve user privacy even in sparsely populated ares, (3) if the system needs user interaction or is autonomous[2], (4) which system architecture is used – either centralized, decentralized, or if the system is running on the user's device –, (5) if the system adapts dynamically to its environment, and (6) if the system needs *a priori* knowledge (e.g., some kind of history or premonition).

In general, all spatial cloaking systems need a more or less complex infrastructure that facilitates the proposed privacy preserving spatial cloaking techniques. In contrast to the aforementioned methods we leverage the design of a simple systems that does neither rely on $k$-Anonymity, nor on a (complex) infrastructure to be built up before the system can be used. Only some minor changes in the protocol of the existing infrastructure would be necessary to adapt our system design. At the same time, it avoids the blurring required by $k$-anonymity.

# 3 Forge Your Path

In this section we develop the idea of the *PathForge* system, where the user should be able to shape her path. As we want to separate user location from user identification – instead of aggregating $k$ users into a $k$-anonymous groups – while retaining an existing infrastructure.

## 3.1 Cellular Telephony

First, we introduce a simplified view on cellular telephony:: A provider's network is organised in cells, each covered by an access point (AP). Whenever a phone enters a new cell, it registers itself with its AP, so that it can be reached by the network for e.g. incoming call signalling messages or other data.

[2]We agree with Kapadia et al. [KTC+08] that a privacy-enhancing system should not require any user interaction.

This registration information is potentially highly sensitive information, especially as linking of the subscriber ID (IMSI) to a phone number and other customer data is mostly trivial. Thus, we can identify three different states:

1. The device registers itself with the appropriate AP for *reachability*,
2. the device starts an *outgoing* phone call (or some other kind of potentially billable data transmission within the provider's network) and needs to identify itself, and
3. the network is signalling an *incoming* call to our user using the reachability information.

The former state is the most frequent, at least for those people that do not constantly talk, and is thus the one with the highest privacy-invading potential; however, it is required for the network to work; take note that there is little need to provide high-assurance identification, as long as a mechanism ensures incoming calls to be routed properly and the phone ownership is ascertained before allowing it to originate or answer calls.

## 3.2 ID-less Tracking

If no IDs should be tracked, there are two options:

First, pure broadcast system. In such a system – such as legacy one-way pagers – the message is sent over every AP. This would not scale to the billions of users we have today, not even if the caller were to provide an approximate location of the callee.

Second, break the tie between the user's location and its identification. The most simple approach would be to simply remove the device ID from the periodic registration process. Thus, the system could still track devices but would have no clue about the device ID. Simplified, the system would know the location of the $k$ devices in the network, ending up in a $k$-anonymous location database, for a huge $k$. Obviously, this approach *as is* has the same drawbacks as the pure broadcast system. Nevertheless, this scenario, appropriately refined, provides the basis for *PathForge*. We will fix the problems one by one.

## 3.3 1 × ID Switching

Consider two users in the network, Alice and Bob as well as the service provider Susan. Whenever Alice an Bob meet, they swap their phones. Each path of the phones retains full fidelity, but the coupling to their users' IDs is broken (Fig. 1). When Alice receives a call destined for Bob, she just informs the caller to dial the other phone's number.

Both Alice and Bob have a secret key $K$ from their network operator, e.g. stored on their Subscriber Identity Module (SIM) and an associated ID based on their key $K$ that is computed as e.g. $I_{\text{proxy}} = h(K \parallel 1)$, where $\parallel$ denotes concatenation and $h()$ a cryptographic one-way hash function, e.g. from the *Secure Hash Algorithm* (SHA) family.[3]

[3]Please note that we use symmetric keys for all operations, even though using asymmetric keys would be possible. As the network provider has manufactured the SIM card and has full control over the network, the added complexity and slower speed of asymmetric keys is unlikely to be compensated by any benefit.

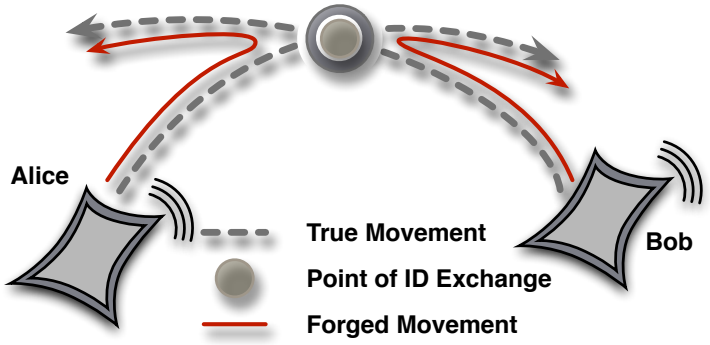



Figure 1: Switching the user's *Proxy ID* $I_{\text{proxy}}$

Alice and Bob use $I_{\text{proxy}}$ to register with an AP and thus the network's location database. To prevent the location from being helpful, Alice and Bob switch their IDs whenever they come in "direct" contact[4] (Fig. 1). From the point of exchange Alice will identify herself as Bob, using Bob's ID as her Proxy ID, and vice versa, thus, forging their motion data by utilizing the motion of each other.

Delivering incoming data (e.g. a phone call) to the intended person is straightforward for Susan. Whenever data needs to be transmitted to Alice, Susan will initially contact Bob, which is currently posing as Alice. Thus, Bob tells Susan to forward the data to $I_{\text{proxy}}^{\text{Bob}}$. To verify that the connection is made to the authorized phone, subsequently, Susan requests Alice to authenticate using their knowledge of $K$. Therefore, Susan sends Alice a nonce $N$ and Alice (Bob) compute an authentication ID as $I_{\text{auth}} = h(K \parallel 2 \parallel N)$, prompting, Alice to send back $I_{\text{auth}}$ for authentication.

Similarly, to initiate data transmission over the service provider's network, Alice needs to authenticate to be able to transmit the data. Note, whenever Alice or Bob initiate a connection or accept incoming data, their true identity will be revealed, and both will set back their false IDs to their initial values as computed, thus, both need to find a new partner for ID switching afterward. We will discuss the revocation of proxy IDs in detail in section 3.4.3 Due to the added overhead of transmitting calls and data messages in multiple cells, we need the users to show their real position for such high-volume transfers.

Fig. 2 shows the (simplified) sequence diagram that highlights the changes that are necessary to the standard protocol used nowadays by service providers to establish a data connection to a client. Basically, only the authentication step (shown as thick red arrows) needs to be doubled as the client that is wrongly connected due to the changed IDs will return the look-up ID $I_{\text{proxy}}$ to use by the service provider to identify the intended receiver instead of $I_{\text{auth}}$. Thus, if the service provider cannot verify the passed ID as $I_{\text{auth}}$ it will use the passed ID to look up the intended receiver.

The problem we face with $1\times$*ID switching* is that Mallory, a malicious entity, can easily

[4]For instance, when the devices of Alice and Bob are within Bluetooth range.

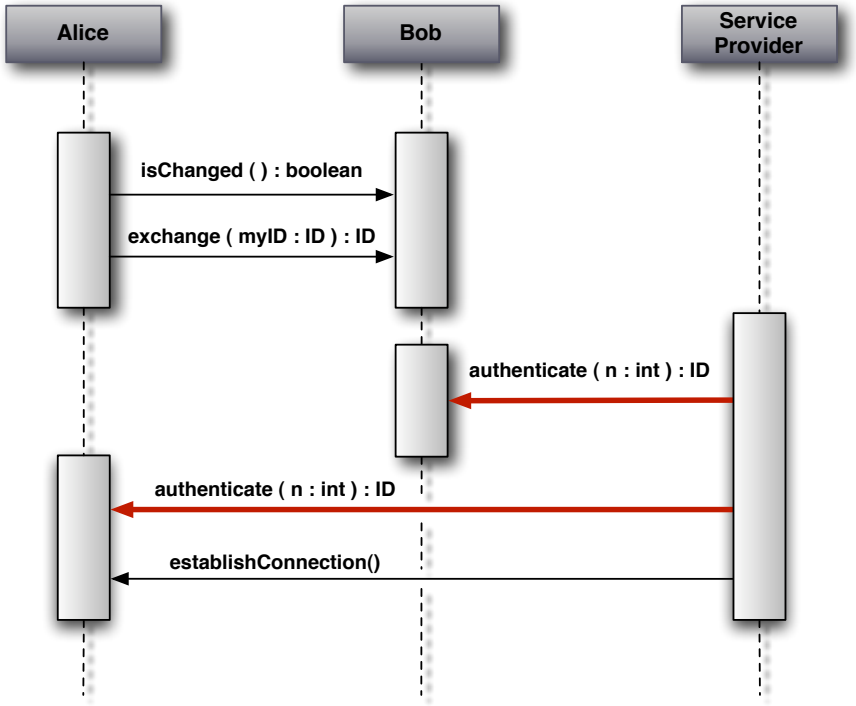



Figure 2: Protocol to look-up and authenticate the intended user after $1\times$*ID switching*.

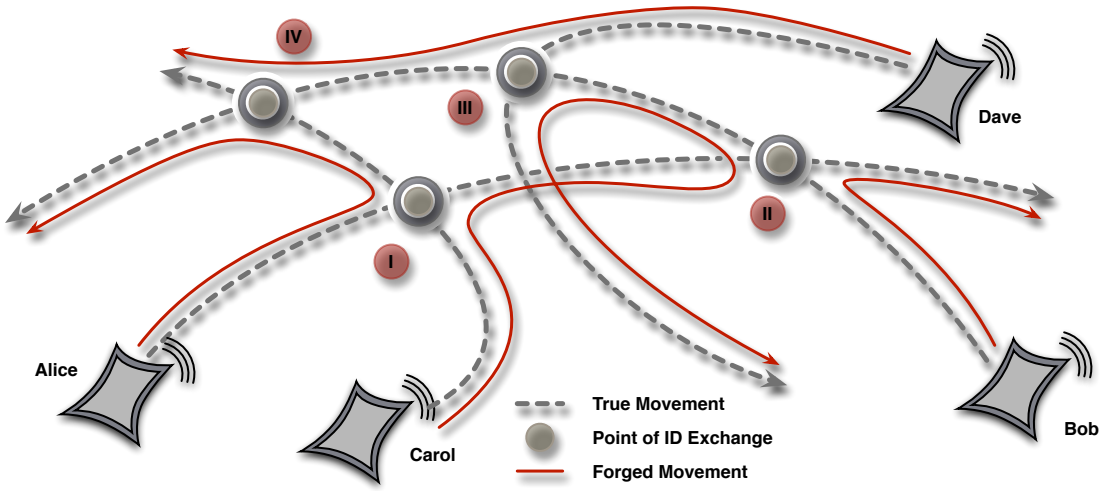
infer the true location of either Alice or Bob through backtracing. To reveal the true location of either one of the two, Mallory only has to backtrack the path of one of the IDs and check where Alice and Bob where close to each other.

## 3.4 $N \times$ ID Switching

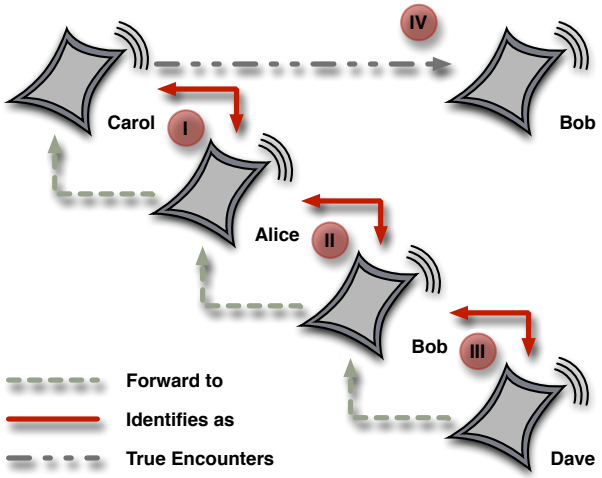
Judging from the backtracing vulnerability described above, $1\times$*ID switching* is not enough to prevent a malicious party to infer the location of a user. Therefore, we need to prevent the ID swapping location from being identified. This can be achieved by swapping IDs multiple times with different users along the way, shuffling identities like a deck of cards.

Consider four users in the network, Alice, Bob, Carol, and Dave. All of them can exchange their IDs multiple times with each other. . Thus, whenever a users gets in contact with another user their IDs will switch. Therefore, Mallory will now have an upper bound of $O(n^n)$ paths to infer the true location of Alice – again, $n$ is the number of encounters, though, in this case not only the direct encounters of Alice but the encounters on the path forged by $I^{\text{Alice}}_{\text{proxy}}$. As the upper bound is exponential large this should be sufficient to effectively prevent Mallory to infer the true location of Alice.

After having solved the backtracing problem, we are facing yet another puzzle: How to efficiently forward incoming data to Alice. Using the look-up approach discussed in Sec. 3.3, Fig. 2 we end up in a cascade of false users, and every user would ask the service provider




(a)




(b)

Figure 3: Switching IDs multiple times. The Roman numerals in (a) correspond to the numbers in (b).

Susan to forward the incoming data to the next user in the cascade. For instance, consider Carol in Fig. 3a. At point $I$ she switches her ID with Alice. At point $II$ Alice, who identifies herself as Carol at this time, switches IDs with Bob. Fig. 3b depicts the cascade that was built up to point $III$. Thus, at point $III$ Dave actually identifies himself as Carol, Bob identifies himself as Dave, Alice identifies herself as Bob, and Carol identifies herself as Alice.

Note, at point *IV* Carol actually identifies as $I_{\text{proxy}}^{\text{Dave}}$, thus, the authorization request would luckily succeed at this point and Carol could accept the incoming data. However, we can identify three distinct problems with $N \times ID$ *switching* so far:: (1) In the worst case Susan must contact every user in the cascade to identify the intended receiver , (2) every user in the cascade and the network in general is a potential point of failure, for instance, when a user switched off his cell phone or is out of reach of any AP, and (3) every participating user in the cascade must be notified that their false IDs must be revoked after any user in the cascade revealed her true ID.

### 3.4.1 Remove Look-Up Cascade

To enable Susan to look up the intended receiver of the data to transmit without checking every user in the cascade, we reveal information about the ID swaps, and thus reintroduce a (this time more complex) backtracing leak, we introduce a second ID $I_{\text{lup}}$ for every user. Similarly to $I_{\text{proxy}}$ we define $I_{\text{lup}} = h(K \parallel 2)$. Anytime two users switch their ID, both will also exchange $I_{\text{lup}}$ along with an agreed seed $s$[5]. The seed $s$ is used to initialize a *pseudo-random number generator* (PRNG) on both client devices. We use the PRNG to compute pseudorandom time intervals, starting from the point in time the ID exchange took place. These time intervals are subsequently used to register with the carrier. Therefore, we compute $I_{\text{treg}} = h(I_{\text{lup}} \parallel r)$, where $r$ is a pseudorandom number obtained from the PRNG. Anytime a user comes into a new cell in the network or $I_{\text{treg}}$ was re-generated due to the expiration of the time interval, the client will register with::

1. $I_{\text{treg}}$ to be reachable as a destination whose identity will only be revealed at the receipt of a call or the generation of a billable event, and

2. $I_{\text{proxy}}$ to provide a forwarding point to the real identity.

Note, as $I_{\text{treg}}$ changes regularly it cannot be associated with $I_{\text{proxy}}$. Thus, $I_{\text{treg}}$ cannot be used to infer any movement data and does not pose a privacy risk. Subsequently, delivering incoming data to the intended receiver will now involve $I_{\text{treg}}$ instead of $I_{\text{proxy}}$. Again, consider our users Alice, Bob, Carol and Dave that exchanged their IDs according to Fig. 3a. To initiate a connection with Carol, Susan will initially contact Dave as he is posing as Carol, requesting for authorization. As Dave has no knowledge about $K^{\text{Carol}}$ he cannot pass the authentication, but, returns $I_{\text{treg}}^{\text{Carol}}$ (instead of $I_{\text{proxy}}^{\text{Dave}}$ as described in Sec. 3.3) to Susan. Therefore, Carol, who is periodically identifying herself using $I_{\text{treg}}^{\text{Carol}}$, can be identified easily by Susan. We end up with two authentication requests by Susan, the first request submitted to Dave who is posing as Carol, and the second to Carol herself – basically, the same two-pass look-up protocol as depicted in Fig.2.

Thus, we solved two of the three previously identified problems, namely (1) contacting every user in the cascade, and (2) there is only a single point of failure left (Dave) instead of having $u$ possible points of failure, where $u$ is the number of participants in the cascade.

[5] Note, $s$ will change for every ID exchange.

### 3.4.2 Tolerate Failures

As discussed in the preceding section we can reduce the look-up process to a two-pass look-up to identify the intended receiver. Thus, we only have a single point of failure left, the user that poses to be the intended receiver at this time. However, we can even do better. Consider that every user in the network can identify oneself with more than one fake ID at once. Basically, we maintain an array of length $l$ of proxy IDs per user. In this case Susan has the alternative to choose from $l$ different users in the network. Let $p_{\text{fail}} \in [0 \dots 1]$ be the probability that a user that poses to be the intended receiver of the data fails to be contacted. Obviously, this would reduce the probability of failing to contact the intended receiver to $\prod_{i=0}^{l} p_{\text{fail}}^{i}$ where $p_{\text{fail}}^{i}$ is the probability of failure for the user with index $i$ in the array of fake users.

### 3.4.3 Revoke Invalidated Proxy IDs

The last puzzle to solve is how to revoke IDs that were rendered invalid due to a user revealing his true identity. Revoking invalid proxy IDs in the system is straightforward. Along with $I_{\text{treg}}$, every user in the network identifies with a tuple $(I_{\text{proxy}}, c)$ instead of $I_{\text{proxy}}$, where $c$ is a counter that is incremented after the user has revealed his true identity to the system. Again, consider the example in Fig. 3a. In the case Carol reveals her identity to the system she will inform Susan that the tuple $(I_{\text{proxy}}^{\text{Carol}}, c^{\text{Carol}})$ she passed to Alice at point $I$ – and that was subsequently passed to Bob and Dave – needs to be revoked. Therefore, Carol sends a revoke request to Susan passing over the invalid tuple. As soon as another user tries to identify herself using the revoked tuple she will get a notification that the tuple is invalid. Thus, if Carol reveals her identity at point $III$ and revokes her tuple, Dave will get notified about the invalidated tuple and will himself notify Susan to invalidate his former tuple – this results in a cascading revocation of all tuples in the cascade depicted in Fig.3b.

# 4 Conclusion

In this paper we presented the basic concept of a *movement forging system* that can be easily adapted to a service provider's existing infrastructure. Our main concern is to prevent malicious parties to infer the true location of the users in our system while maintaining the accuracy of the motion data for traffic and flow analyses. Anonymization in *PathForge* is solely based on switching user IDs, thus, we only work on user level and do not need any centralized system. Nevertheless, the design of such a system is a process of several obstacles, all of which can be successfully overcome. This results in the necessary techniques to enable a service provider to look up the intended destination of client efficiently without introducing a computational overhead and that only requires minor changes on the side of the service provider. Additionally, even in the case that a user reveals his true identity due to incoming or outgoing data, the revealed location information will not be enough to infer either the path the user has taken nor the destination of the user in the future as the

revealed locations are only samples of the true movement pattern. Simplified, even when the user reveals his location from time to time the future location of the user cannot be easily predicted (c.f. Krumm and Horvitz [KH06]). A problem that still remains is the topic of anonymous data transmission and reception, removing the need for revealing a position ever. For VoIP applications as well as other packet based traffic we consider to attach our system to the *Tor*[6] onion-routing system. For traditional cell-phone telephony we will investigate the applicability of $k$-anonymity techniques similar to the techniques proposed in [CML06] and [GKS07].

# References

[BRN09] Brand-Republic-News: Google's mobile phone tracking service under fire from privacy critics, February 2009.

[CML06] Chow, C.-Y.; Mokbel, M. F.; Liu, X.: 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*, S. 171–178. ACM, New York, NY, USA, 2006.

[CN08] CNET-News: Exclusive: Widespread cell phone location snooping by NSA?, September 2008.

[FN04] Fox-News: Man accused of stalking ex-girlfriend with GPS, September 2004.

[GKS07] Ghinita, G.; Kalnis, P.; Skiadopoulos, S.: PRIVE: Anonymous location-based queries in distributed mobile systems. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, S. 371–380. ACM, New York, NY, USA, 2007.

[GL05] Gedik, B.; Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, S. 620–629. IEEE Computer Society, Washington, DC, USA, 2005.

[HG05] Hoh, B.; Gruteser, M.: 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*, S. 194–205. IEEE Computer Society, Washington, DC, USA, 2005.

[HGXA07] Hoh, B.; Gruteser, M.; Xiong, H.; Alrabady, A.: Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, S. 161–171. ACM, New York, NY, USA, 2007.

[6] http://www.torproject.org

[KH06] Krumm, J.; Horvitz, E.: Predestination: Inferring destinations from partial trajectories. In *Ubicomp*, S. 243–260, 2006.

[KTC+08] Kapadia, A.; Triandopoulos, N.; Cornelius, C.; Peebles, D.; Kotz, D.: AnonySense: Opportunistic and privacy-preserving context collection. In Jadwiga Indulska, T. R., D. J. P.; Ott, M. (Hrsg.): *Pervasive Computing*. LNCS, Springer-Verlag, 2008.

[MAHP03] Mokbel, M. F.; Aref, W. G.; Hambrusch, S. E.; Prabhakar, S.: Towards scalable location-aware services: requirements and research issues. In *GIS '03: Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, S. 110–117. ACM, New York, NY, USA, 2003.

[NYT98] New-York-Times: F.B.I. seeks access to mobile phone locations, July 1998.

[SS98] Samarati, P.; Sweeney, L.: Protecting privacy when disclosing information: $k$-Anonymity and its enforcement through generalization and suppression. Technischer Bericht, 1998.

[Sta08] Starostik, M.: Verfassungsbeschwerde: Vorratsdatenspeicherung. Lawsuit filed at the German Constitutional Court, September 2008. In German.

[Swe02] Sweeney, L.: $k$-Anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, Band 10, Nr. 5, S. 557–570, 2002.

[USA02] USAToday: GPS systems used to stalk woman, December 2002.