# Context-derived Pseudonyms for Protection of Privacy in Transport Middleware and Applications

David Evans Alastair R. Beresford
Computer Laboratory
University of Cambridge
Cambridge, United Kingdom. CB3 0FD
{David.Evans,Alastair.Beresford}@cl.cam.ac.uk

Trevor Burbridge Andrea Soppera
British Telecommunications plc.
Adastral Park
Martlesham Heath, United Kingdom. IP5 3RE
{Trevor.Burbridge,Andrea.2.Soppera}@bt.com

## Abstract

*This paper outlines why context-aware transport applications necessarily record information about people and describes how we can use anonymity techniques to minimise the resulting invasion of privacy. In particular we: (i) describe how to generate unlinkable personal pseudonyms; (ii) describe a method of creating group pseudonyms (an opaque pseudonym representing several individuals); and (iii) describe how to store these pseudonyms safely in a database. We go on to present two sample transport applications which utilise our technique and suggest areas for future work.*

## 1 Introduction

Future transport applications will utilise technology to encourage more effective use of limited resources. One method which looks particularly promising is to build context-aware transport systems. Such systems use sensors to record pertinent information, construct from these data a model of the real-world environment, and use this model to allow applications to serve their users in an automatic or semi-automatic way. For example, a bus alert application may notify an individual when to leave work in time to catch up with a friend who is already on the bus. To make the alert message useful, the application's context may encompass information about the friend's location, position data collected from the bus, prevailing congestion information, and the juxtaposition of bus stop and office location. Effective context information, by necessity, includes information about people. This means that any transport application of non-trivial complexity will gather, collate, and distribute a tremendous quantity of personal information and could have a major impact on the privacy of individuals. Our aim is to minimise this impact whilst still allowing context-aware applications to function.

While the ability to locate friends and family is often put forward as an appealing application of ubiquitous computing systems, we believe that people are not typically interested in their friends' physical locations but rather in what their friends are doing or the stage they are at in a predefined set of activities. For example, Bob is probably less interested in Alice's current location than he is in the fact that she is no longer on the bus and has entered the pub. Bob's query could be answered if Alice's name were added to a roster of bus passengers for the duration of her journey and then placed on a list of those inside the pub. However, this clearly violates Alice's privacy without some restriction on who can retrieve this information.

In the past, we have argued that anonymisation rather than access control is the appropriate mechanism to achieve privacy for transport applications [5]. Here we make our recommendations more concrete by illustrating how pseudonyms—digital identities which offer control over their linkability to real people—can be constructed and describing architectures suitable for a pseudonym database.

Anonymising location data is not an easy task since certain regions of space and time have a strong sense of identity associated with them. We call such regions *home locations*. Example home locations include the immediate

area surrounding an office desk during the working day and the doorway to a flat or apartment in the morning or early evening. The presence of home locations in the environment means that giving each user a single static pseudonym is insufficient to protect privacy since such a pseudonym can be strongly associated with a real individual [1]. Instead our scheme generates pseudonymous identities that change with the user's context.

Our scheme naturally extends to allow pseudonyms for groups of people. Creation of individual pseudonyms is efficient, permitting an implementation on devices like mobile phones that have constrained computation and memory resources. Complex message exchanges are not required, thereby ensuring that users' natural behaviour is not impeded by time-consuming interactions. Our goal is for this pseudonym scheme to form an integral part of a transport middleware; ideally, all applications built on top of this middleware would make use of pseudonyms as a matter of course.

This paper is organised as follows. Section 2 outlines previous work on preserving privacy. Section 3 describes our threat model while Section 4 presents the pseudonym mechanism itself and describes the design options available for the pseudonym database. In Section 5 we describe how the scheme may be used to implement two applications: locating friends and reserving taxis. Section 6 outlines some possible extensions to our mechanism, while Section 7 concludes and describes our plans for future work.

## 2 Related work

The use of pseudonyms or unlinkable identifiers to protect privacy is based on the premise that anonymity protects the privacy of an individual, which Pfitzmann and Köhntopp define as "the state of being not identiable within a set of subjects, the anonymity set" [11].

Chaum was one of the first to use anonymity to preserve the privacy of users of a communication network [2]. This has developed into an active research field, with anonymous communications solutions developed for web browsing [12], messaging [3] and even GSM mobile telephony [8]. The related field of statistical disclosure control uses anonymity to protect the privacy of personal information stored in databases [14], and in the transport arena, Grutser and Grunwald used anonymity techniques to protect the location privacy of vehicle drivers [6].

## 3 Threat model

We now outline our security goals and threat model. We consider three parties: (i) the transport middleware operator; (ii) Alice, who is willing to share some context information with Bob; and (iii) Bob, who is trying to determine the current context of Alice. We assume the use of a communication mechanism that prevents trivial linking of pseudonym transmissions with user identities. This can be done via a mix network such as Tor [4].

First, we want the privacy of the individuals who are publishing pseudonyms, in this case Alice, to be preserved. Without authorisation, the middleware operator or an eavesdropper should not be able to determine the current context of individuals. In particular, the operator should not be able to track the movements of a user through the transport network. Although route usage information may be a valuable asset for the transport operators, users should be able to control information release by selecting a pseudonym with an appropriately short-lived context. This minimises the impact of any home locations which might exist in the environment.

Second, we want to protect the privacy of recipients of pseudonym data, in this case Bob. In particular we want to prevent adversaries from tracking his movements, determining his context, and impersonating him. If contexts are short-lived, it reduces the chance that Bob repeatedly makes queries for the same pseudonym (which could be detected by analysis of query logs) and the pseudonyms will be less useful in mounting replay attacks.

Third, we want reasonable assurance that Alice does not disclose any unintended information through the conjunction of her pseudonyms and queries on those pseudonyms. For example, the transport operator may know that at a particular bus stop, a successful query is performed against a pseudonym that is stored on the approaching bus. The operator could then assume that if only one individual boards the bus, he or she made that query and the pseudonym in question relates to the individual he or she sits next to. We return to this issue later in the paper.

## 4 Pseudonym mechanism

Users construct pseudonyms for themselves at points in time where they wish to be identified by others. Each new pseudonym is then transmitted to the system which stores it in a database. Finally, other users submit queries by forming pseudonyms that they suspect exist and asking the database whether this is in fact the case. We will now describe how pseudonyms are created and how the pseudonym database supports storage and queries.

### 4.1 Pseudonym creation

In order to explain the computations that are used to construct a pseudonym, we first present some assumptions. We assume that each individual is in possession of a mobile device, such as a phone, that will manage his or

her pseudonyms. Individual $i$ has a single identity, $\mathrm{ID}_i$, that is known to everyone who will be using his or her pseudonyms, and a set of keys, $\{K_{i,1}, K_{i,2}, \dots\}$, where key $K_{i,j}$ is known only to individuals $i$ and $j$. Key exchange may be effected by, for example, a Bluetooth data exchange between mobile devices or by trading business cards showing codes that can be scanned by a device's camera. We also assume that there exists a set of contexts $C = \{C_1, C_2, \dots\}$ that can be read by the devices. The context for a particular pseudonym encompasses the knowledge about the environment that will be used by others to link the pseudonym with its corresponding user. Candidates for the context include bus route numbers, pub names, and street names; the context can also include a value that can only be determined by individuals in a certain physical location (such as a number displayed on the wall that changes once per minute). In order to avoid a particular user generating the same pseudonym upon every visit to a particular location, values such as the date can be included as part of the context.

A pseudonym for individual $i$ that provides for linkability by individual $j$ within context $n$ is denoted $P_{i,j}^{(n)}$ and is constructed as

$$P_{i,j}^{(n)} = H_{K_{i,j}}(\mathrm{ID}_i, C_n) \tag{1}$$

where $H_K(\cdot)$ is a message authentication code with key $K$. A message authentication code combines its argument and the key into a value that is dependent on each. If $y = H_K(x)$, one can deduce neither $x$ or $K$ given only $y$, nor can one determine $y$ given $x$ but not $K$. Furthermore, it is infeasible to find $K'$ and $x'$ such that $y = H_{K'}(x')$. For more details of MACs and an example implementation, see RFC 2104 [9].

A pseudonym $P$ constructed using Equation 1 has the following properties:

1. Given access to $P$, an attacker without the correct key cannot infer the ID or the context of the corresponding individual.

2. Given access to $P$ and the correct key, a user may be able to infer the context because the size of $C$ is likely to be small, the correct context can probably be guessed by brute force.

3. Given $P_{i,j}^{(n)}$ and $P_{i,j}^{(m)}$, individual $j$ can link the movements of user $i$ as he or she moves from context $n$ to context $m$.

4. Given access to $P = P_{i,j}^{(n)}$, $P' = P_{i,j}^{(m)}$, but not $K_{i,j}$, an attacker cannot determine that $P$ and $P'$ represent the same individual.

5. Unrestricted distribution of the pseudonym does not compromise real-world identity.

These properties mean that linkability is controlled via the distribution of $K_{i,j}$. Furthermore, Property 5 implies that when pseudonyms are placed in a database, the database can answer queries from anyone without weakening the linkability control vested in the users themselves. The database does not have to be trusted to provide access control.

In this model, because each key is known to only two people, a user wishing to share pseudonyms with $M$ friends must construct $M$ pseudonyms. In general, we expect $M$ to be small; for example, one study has found that American teenagers have an average of about two dozen friends [10, page 12], while another found that only 9% of respondents regularly converse with more than 10 people using instant messaging services [13, page 7]. However, the mechanism does not strictly require this approach to key sharing. Key $K_{i,j}$ can be re-interpreted as being generated by user $i$ and shared with those users in the set $j$. In this case, all users within $j$ will have equivalent ability to link the pseudonyms constructed using $K_{i,j}$. Furthermore, these users will share the ability to generate pseudonyms under key $K_{i,j}$ and, for any particular pseudonym, it will be impossible to determine the user that created it.

Before introducing our scheme for group pseudonyms, we outline the privacy preserving set representation due to Hohenberger and Weis [7], hereafter referred to as HW, that is its basis. Consider a set of integers $A = \{a_1, a_2, \dots, a_{|A|}\}$. Let $K = (G, n, p)$ where $G$ is a carefully-selected multiplicative group of composite order $n = pq$ where $p$ and $q$ are large primes. HW specifies how one can transform $A$ into $A_K = \{\delta_1, \delta_2, \dots, \delta_m\}$ where $m \geq |A|$ can be selected to be as large as is desired. We denote this procedure $\langle\cdot\rangle_K$, meaning we can write $A_K = \langle A \rangle_K$.

By providing someone with $A_K$ and $K$, he or she can test whether any arbitrary integer $b$ is in the set $A$ but obtaining other knowledge about the membership of $A$ is impossible. Furthermore, $A_K$ is randomised: multiple runs of the algorithm with inputs $A$, $K$, and $m$, will produce different $A_K$s, each of which represents $A$.

For a set of pseudonyms $G = \{P_1, P_2, \dots\}$, we use HW directly to form the group pseudonym for $G$, known as $P_G$, *i.e.*,

$$P_G = \langle \{P_1, P_2, \dots\} \rangle_K$$

for some key $K$. Such group pseudonyms have the following properties.

1. Given access to $P_G$, an attacker without the correct key cannot infer any of the pseudonyms within the group $G$.

2. Given access to $P_G$ and $K$, a user can check whether specific pseudonyms are in $G$. However, given that the pseudonyms must be known, the user in question must be in possession of the appropriate individual

pseudonym keys. (Obviously the user must also either know the appropriate context or be able to find it via brute force.)

3. Given $P_G$ and $K$, a user knowing the keys to some of the pseudonyms in $G$ learns nothing about the status of pseudonyms whose keys are unknown.

4. Given $P_G$, it is not possible for a user to directly determine $|G|$. All that is available is $m$, an upper bound.

5. Suppose that $P_G$ and $P'_G$ are two pseudonyms corresponding to $G$, generated using the randomisation feature of HW. Even if they share the same key, examination will not reveal that they both represent $G$.

6. Unrestricted distribution of the pseudonym, once again, does not compromise the real-world identity of anyone in the group.

This scheme has the caveats that the size of $P_G$ is $O(m)$ and the time required for a membership test is also $O(m)$. Thus there is a drawback to using a large value of $m$ to conceal $|G|$.

### 4.2 Pseudonym database

After creation by a user, a pseudonym is sent to a database for storage. Care must be taken that such publication does not reveal the individual's identity to the database operator; this may be possible if the operator can observe the individual's change in context as publication takes place. Publication in advance of or following the context change can mitigate this attack, as can ensuring that a particular individual is one of many publishing pseudonyms at any given moment.

There are two options for the architectural design of the database: a single database can be responsible for all pseudonyms within the system, or many small databases can manage those pseudonyms having specific contexts. In the single database model, users submit their pseudonyms to a well-known database that serves the entire transport middleware system. Traditional methods for achieving scalability can be applied to ensure that the response time for pseudonym storage and queries is satisfactory and this may lead to a distributed database implementation. However, as far as users are concerned, there is only a single pseudonym repository. This approach has the disadvantage that the database is well-positioned to perform traffic analysis on requests for storage and retrieval. Although this should not compromise the identities of the pseudonym holders, it may permit reconstruction of the social links between them and those who are looking for them. Nevertheless, this approach has the advantage of simplicity: no mechanism is required to determine where a pseudonym might be stored and maintenance efforts can be concentrated.

As opposed to constructing a single database for the entire system, we can use multiple disjoint databases, each responsible for pseudonyms having contexts with a certain degree of commonality. For example, a database on a bus could store pseudonyms of the passengers currently on the bus. This decentralised approach has the advantage that collusion is required to infer connections between pseudonyms within sufficiently different contexts. However, there is the complication that users (whether storing a pseudonym or querying for one) must determine the correct database to use. Furthermore, the fact that multiple databases are required means that the complexity of maintenance may be unattractive and the most natural place to locate a particular database may not be feasible; for example, transit vehicles operating in harsh environments may be unable to support communication with their passengers.

Users make queries by forming a suspected pseudonym and forwarding it to the database. The database's response indicates whether or not the pseudonym is currently present. This means that the database operator knows the outcome of the users' queries and, in the context of our threat model described in Section 3, a small loss of privacy will result. Finally, the database will, upon request, remove a specific pseudonym. Access control is not required for this operation because we assume that all parties able to form a particular pseudonym have the rights to revoke it.

## 5 Sample applications

We now describe two transport applications that can be implemented using our pseudonym scheme.

### 5.1 Friend finder

Suppose that Alice and her friend Bob are travelling to a common destination using public transport. They live on the same transit route and a bus following this route will first pick up Alice and then, some time later, Bob. Bob would like to travel on the same bus as Alice. To do this, when Bob arrives at his transit stop, he needs to know whether Alice is on the bus that is approaching.

Suppose that each bus has its own pseudonym database. (This application can also be implemented using a single database.) Let the number plate for bus $i$ be $C_i$. Furthermore, assume that Alice and Bob share key $K_{A,B}$ and that it is known only to them. Let Alice's ID be $\mathrm{ID}_A$. As Alice boards a particular bus, say bus $i$, she uses Equation 1 to form the pseudonym $P^{(i)}_{A,B}$. This pseudonym is then stored in the bus' database; upon alighting, Alice removes it.

When Bob sees bus $j$ approaching, he forms the pseudonym $P^{(j)}_{A,B}$, again using Equation 1. This would be

Alice's pseudonym were she on bus $j$. Bob then asks the bus whether its database contains $P^{(j)}_{A,B}$; it will only if $i = j$. In this case, $P^{(j)}_{A,B}$ must have been placed there by Alice because she is the only other party in possession of $K_{A,B}$. Bob can conclude that Alice is on bus $j$.

### 5.2 Group taxi reservation

We next outline a simple protocol to implement a group taxi reservation service. Suppose that the organiser of an event such as a conference has arranged special taxi pricing for conference attendees. In our protocol, the organiser first forms a pseudonym for each attendee. We assume that there is a single key $K_{O,T}$ that is shared between the organiser and the taxis. Attendee $i$'s pseudonym $P_i$ is computed as $P_i = H_{K_{O,T}}(\text{ID}_i, \text{event ID})$, where $\text{ID}_i$ is individual $i$'s identity within the event (such as a badge number) and "event ID" is an identification value unique to the event. The organiser then generates a group pseudonym key $K_1$, constructs $P_A = \langle\{P_1, P_2, \dots\}\rangle_{K_1}$, and transmits $K_1$ and $P_A$ to each of the taxis. Similarly, for each taxi $j$, the organiser generates the pseudonym $R_j = H_{K_{O,A}}(\text{ID}_j, \text{event ID})$, where $\text{ID}_j$ is the taxi's number plate and $K_{O,A}$ is a key shared between the organiser and the attendees. After generating group pseudonym key $K_2$, the organiser forms $P_T = \langle\{R_1, R_2, \dots\}\rangle_{K_2}$ and transmits it and $K_2$ to each attendee.

$P_T$ represents the set of taxis assigned to the attendees while $P_A$ represents the set of attendees that should receive the discount. Upon sighting a taxi, an attendee can form the taxi's pseudonym using $K_{O,A}$ and the taxi's number plate; this can be then be tested against $P_T$ to determine whether the taxi is one arranged for by the organisers. Likewise, taxis upon seeing a passenger can, using the passenger's ID, form the appropriate pseudonym and test for its membership in $P_A$.

## 6 Extensions

We now introduce some extensions to our pseudonym generation scheme that enhance its utility in certain situations.

**Pseudonym pseudo-revocation** While there may be straightforward mechanisms to place pseudonyms in the database at appropriate times, it may be difficult to later remove them. For example, for the "friend finder" application, it is natural for passengers to produce their pseudonyms as part of presenting transit payment credentials, but there may be physical constraints making it difficult for them to remove the pseudonyms upon alighting. To solve this problem, the user can augment a submitted pseudonym $P_i$ with a "qualifier" $Q_i = E_K(P_i, \text{context restriction})$, where $E_K(\cdot)$ encrypts its argument under key $K$ and the context restriction describes limitations on the validity of the pseudonym. Restriction might comprise a range of bus stops, the duration of travel in time, *etc.* When retrieving the pseudonym, $Q_i$ is also returned and someone in possession of $K$ can decrypt it to check for validity. (Note that instead of using $E_K(\cdot)$, public key cryptography could be used, yielding asymmetric operation of $Q_i$.)

**Signed pseudonyms** From time to time it may be useful to determine that a pseudonym was constructed by a given party, which may or may not be the individual referred to by the pseudonym. To this end the pseudonym may be signed using an appropriate private/public key scheme, or using a blind signature by a trusted third party having a well-known public key.

## 7 Conclusions and future work

In this paper we have made concrete our suggestion that pseudonyms are an effective means of controlling access to personal information in pervasive transport applications. We have illustrated how these pseudonyms can be constructed, using users' contextual information and controlled key distribution to effect limited linkability, and have outlined the principal two architectures available to pseudonym database designers. We have also described how two transport-related applications can be implemented.

Our goals for the future begin with allowing non-exact context matches, thereby reducing the number of explicit pseudonyms that must be stored in order to thoroughly describe a user's current location and activities. As a part of this process, we intend to develop guidelines for selecting the most useful context to include in pseudonyms, and explore ways of acquiring this context (via, for example, automatic number plate recognition software controlled by mobile phones). Furthermore, we intend to build this system, thereby gaining concrete knowledge of database scalability, viable communications mechanisms, and usability issues.

## Acknowledgements

This research is supported by EPSRC grant EP/C547632/1 and by British Telecommunications plc.

## References

[1] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 3(1):46–55, 2003.

[2] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[3] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[4] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of 13th USENIX Security Symposium*, pages 303–320, August 2004.

[5] D. Evans and A. R. Beresford. Pseudonymous context-aware transport applications. In *Proceedings of the UK-Ubinet Workshop*, July 2006.

[6] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the ACM/USENIX International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[7] S. Hohenberger and S. A. Weis. Honest-verifier private disjointness testing without random oracles. In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies*, pages 265–284, June 2006.

[8] D. Kesdogan, H. Federrath, A. Jerichow, and A. Pfitzmann. Location management strategies increasing privacy in mobile communication systems. *Information systems security: facing the information society of the 21st century*, pages 39–48, May 1996.

[9] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. RFC 2104, 1997.

[10] A. Lenhart, M. Madden, and P. Hitlin. Teens and technology. Pew Internet and American Life Project, 2005.

[11] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability and pseudonymity—a proposal for terminology. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9, 2000.

[12] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions Information Systems Security*, 1(1):66–92, 1998.

[13] E. Shiu and A. Lenhart. How Americans use instant messaging. Pew Internet and American Life Project, 2004.

[14] L. Willenborg and T. de Waal. *Statistical Disclosure Control*. Springer, 2001.