Technical Report RT/23/2011

Privacy in Distributed Context-Aware Systems

Pedro Alves
INESC-ID/IST/Opensoft
pedro.h.alves@ist.utl.pt

Paulo Ferreira
INESC-ID/IST
paulo.ferreira@inesc-id.pt

May 2011

Resumo

Context-aware systems aim to deliver a richer user experience by taking into account the current user context (location, time, activity, etc.), possibly captured without his intervention. For example, cellphones are now able to continuously update the user's location while, at the same time, users execute an increasing amount of activities online, where their actions may be easily captured (e.g., login in a web application) without user consent.

The potential advantages of these applications is huge, but we must not forget the important privacy issues that it creates. Particularly in distributed context-aware systems where the information that is captured may be sent to a distant server, people feel uncomfortable about exposing highly personal data, such as their location or current activity. This paper examines the current privacy management techniques and identifies four main groups: *Privacy Policies*, *Data Perturbation*, *Anonymization* and *Lookup Notification*. Each of these groups is analyzed highlighting the advantages and disadvantages of their application to Distributed Context-Aware Systems.

Privacy in distributed context-aware systems

Pedro Alves

INESC-ID/IST/Opensoft

pedro.h.alves@ist.utl.pt

Paulo Ferreira

INESC-ID/IST

paulo.ferreira@inesc-id.pt

2011-05-17

Abstract

Context-aware systems aim to deliver a richer user experience by taking into account the current user context (location, time, activity, etc.), possibly captured without his intervention. For example, cellphones are now able to continuously update the user's location while, at the same time, users execute an increasing amount of activities online, where their actions may be easily captured (e.g., login in a web application) without user consent.

The potential advantages of these applications is huge, but we must not forget the important privacy issues that it creates. Particularly in distributed context-aware systems where the information that is captured may be sent to a distant server, people feel uncomfortable about exposing highly personal data, such as their location or current activity. This paper examines the current privacy management techniques and identifies four main groups: *Privacy Policies*, *Data Perturbation*, *Anonymization* and *Lookup Notification*. Each of these groups is analyzed highlighting the advantages and disadvantages of their application to Distributed Context-Aware Systems.

1 Introduction

In recent years, we have been watching a tremendous growth of available personal sensing devices such as the iPhone and Android cellphones. These devices have come to include multiple sensors such as GPS, Wifi/3G, accelerometer and light sensor, and can run a variety of applications. In a sense, the "pervasive" world envisioned by Mark Weiser in 1995 [Weis 95], where devices integrate seamlessly into their users everyday life, is becoming a reality.

However, this seamless integration creates important privacy issues. People are sensitive about revealing their location or activities but these systems often transmit these and other types of context information[1] without requiring a specific user action, in order to increase their usability. For example, the CenceMe system [Milu 08] transmits periodically the current user's location to a group of friends. It wouldn't make sense to ask the user permission before each transmission and users are comfortable with automatic transmissions because they have previously defined with whom they are willing to share their location. In context-aware systems used by large communities, this may not be the case. Users who easily share their location among

---

[1] Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

their friends will probably reject opening up this information to the whole community, because they don't trust the recipients of that information. Still, community context-aware systems can provide immense value to their users (e.g, traffic congestion) so techniques to protect users' privacy in these systems must be employed. Clearly, the main challenge here lies on providing a fair balance between loss of privacy (what we send) and value added by the service provider (what we receive back).

Although previous research on this topic has mainly focused on the specific issues of location privacy (the ability to prevent other parties from learning one's current or past location [Bere 05]), we analyze privacy management techniques that are suitable for all kinds of context-aware systems.

2 Definition of Privacy

Within the realm of distributed context-aware systems, the major privacy concerns pertain to the distribution of personal context to others. In this sense, we can turn to Westin's [West 67] definition of privacy as it is quite appropriate: "the claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others". Oyomno [Oyom 09] further categorizes this definition into three properties that are essential to achieve privacy:

- **Solitude** - Freedom from observation or surveillance, that is, the power to prevent current or past personal context from being visible to others;

- **Anonymity** - Freedom from being identified in public, that is, the power to prevent others from relating context information with the actual person involved;

- **Reservation** - Freedom to withdraw from communication, that is, the power to interrupt personal context propagation, at any time.

Note that these properties do not map into binary (*on* or *off*) behaviors - they represent a spectrum of options that are adapted to each situation. In some situations, a person might allow others to surveil his actions (e.g., cameras in a supermarket to prevent robbery) but in other situations, that would be unbearable (e.g., cameras on his bedroom).

In fact, social psychology studies emphasize the dynamic nature of privacy, which is therefore characterized as a dynamic process of negotiating the boundary between the individual and the environment [Altm 81]. These studies also provided the background that allowed Raento [Raen 05] to provide a different set of properties for privacy-enabled systems:

- **Control** - One must be able to control the type and extent of information revealed to others, but that is typically decided dynamically according to situationally arising needs and demands. These dynamics greatly complicate the development of control mechanisms. Consolvo [Cons 05] shows that although the most important factor in disclosure is the identity of the asker or observer, there are no static rules that can decide what is revealed but that is instead completely situation-dependent.

- **Accountability** - The act of disclosing information usually implies making its recipients accountable for actions that use that information. In fact, the discloser of information may perceive breaches of implicit or explicit agreements on using revealed information and holds the other person accountable for those actions. The multiple layers introduced by distributed context-aware systems [Henr 05] increase the distance between the discloser of information and its recipients, making it much more difficult to understand the implications of such disclosure.

- **Plausible deniability** - When being asked about something private, a person must be able to plausibly deny noticing or understanding the question instead of appearing to refuse to answer. Deniability is clear component of mediated disclosure. Teenagers do not always answer their mobile phones, claiming that they didn't hear the ringing or that the battery was dead. This is plausible for mobile phones due to their unreliability but it can happen in direct face-to-face communication as well, when someone pretends to misunderstand the question or answers vaguely.

- **Reciprocity** - The disclosure of personal information is normally not one-sided, but mostly symmetrical: the amount of disclosure from A to B is strongly related to the amount of disclosure from B to A. Studies have shown that reciprocality in self-disclosure between partners is necessary for building of trust and deepening of relationship [Rubi 75].

- **Utility** - On a more sociological point of view, there are important questions that must be answered, related to the utility of private data. For example, can the utility of private data be measured and traded? This is a very hard problem as the capabilities of future information systems are highly unpredictable. For example, nobody in 1981 knew that their newsgroup postings would be indexed and easily searchable at Google Groups.

Even though these properties are tied to sociological behavior, they provide interesting hints for privacy management techniques to include in context-aware systems. For example, we can predict now that *control* probably requires some kind of *privacy policy*, and that *plausible deniability* may lead to *data perturbation* techniques. In the next section, we describe these techniques in finer detail.
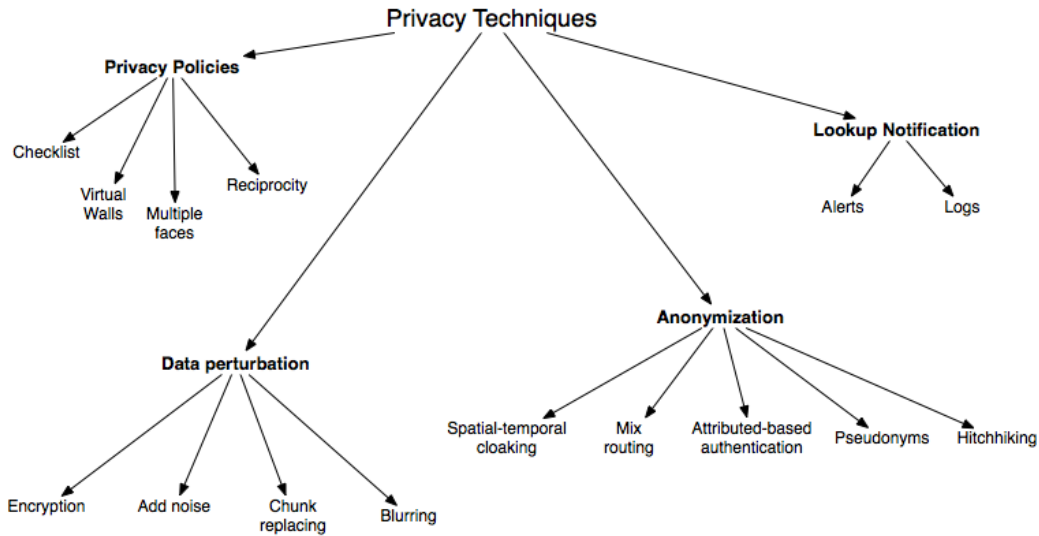
3 Privacy management techniques

Figure 1: Privacy techniques

Privacy management techniques can be categorized into four main types (see Figure 1): Privacy Policies, Data perturbation, Anonymization and Lookup Notification. We start by broadly describing each one, and afterwards we delve into specific techniques for each category.

- **Privacy policies** - Applications using this technique allow the user (the context discloser) to provide rules (privacy policies) that define to whom and to what extent is his information revealed to others. This is the most common technique on both academic and industry social applications (e.g. Facebook, Twitter) and it is tied to both Raento's *control* (although not as dynamic as the original definition implies) and *accountability*. Achieving accountability is probably the reason why this technique has been widely adopted, because it clears all the responsibilities of the service supplier from a judicial view point. However, this approach has revealed multiple problems: it is cumbersome for users to specify fine-grained policies and users are not particularly good at it [Corn 07].

- **Data perturbation** - This type of technique consists on transforming or partially omitting information before being delivered to the context consumer, in such a way that it is impossible to reconstruct the original message while still keeping (some of) its usefulness. This technique may be used for effective *plausible deniability*: since the information that reaches the recipient is incomplete, the user is able provide multiple explanations to fill in the gaps without loosing credibility.

4

- **Anonymization** - Using this technique, the information is delivered intact to context consumers except for its author, which is removed or replaced with one that can not be used to infer the real author. This technique is easy to implement and it is the most effective: if there is no identity associated with the information, privacy is no longer an issue. Anonymization is widely used on many virtual communication tools (e.g., IRC, Forums, Wikis, Second Life), but it is prone to abuses (trolling) [Shir 03] and is being dismissed by recent social applications such as Facebook, Twitter and Linkedin, which would become useless if they allowed anonymous users. Also, most modern social applications include some kind of reputation management directly connected to the user's identity, that rewards those who provide more personal information, effectively reducing the anonymity factor. Nevertheless, for large community social applications which gather information with the sole purpose of providing statistics (e.g., obesity control), anonymization is still very used.

- **Lookup notification** - This technique consists of providing the user with information of who has consumed his context information and when. This can occur in real-time (the user is alerted that someone is consuming his context information) or *a posteriori*, by keeping a log of who has seen which information. Unlike the other three techniques, which are applied before data is delivered, this technique is applied after data delivery (that is, after a potential privacy breach) so it is usually combined with other techniques. However, it can still be a very powerful privacy mechanism. Firstly, it prevents repeated privacy breaches (e.g., due to privacy policy misconfiguration). Secondly, it creates a psychological barrier to potential privacy intruders, as they know their intrusion will be notified. Thirdly, it gives more confidence to the context provider, allowing him to use the application with less stress. This last point illustrates again the already mentioned (and desirable) *control* property.

We will now detail each of these categories describing their specific techniques.

3.1 Privacy policies

Privacy policies consist on a set of rules defined by the information discloser that control who is allowed to consume the information and when. Since setting rules is usually a cumbersome task, most techniques focus on providing simpler mechanisms for doing so, increasing usability and user satisfaction.

- *Checklist* - This is, by far, the most used technique for implementing privacy policies. It presents the user with a checklist of types of information (e.g., personal bio, photos, location), asking the user to choose who is allowed to see that information. This technique does not scale well with increasing types of information nor with a large number of potential consumers. For example, on popular social networking application Facebook, this checklist includes more than 20 topics which users have to decide with whom to share. This leads users to not changing the default privacy settings, oblivious to consequent privacy loss implications. In fact, Gross [Gros 05] found that only 1.2% of Facebook users changed the default privacy settings for profile searchability.

- *Virtual Walls* - Kapadia [Kapa 07] proposes this technique as a solution to users difficulties in defining fine-grained privacy policies. The idea is to setup user-defined policies based on the the concept of walls around physical places where sensors are deployed. These walls can be configured using a GUI and feature a three-level permission scheme: *transparent*, *translucent* and *opaque*. Sensors readings are named "footprints" and can be personal (when they identify the person, like an image or speech recording) or general (related to the environment, like temperature readings). Than, a simple matrix is used to match these two dimensions. Transparent walls propagate all footprints, translucent walls propagate only general footprints and opaque walls don't propagate anything. By using a policy language based on the metaphor of physical walls, the authors expect that users will find this abstraction to be an intuitive way to control access to their digital footprints.

- *Multiple faces* - Users pre-define a small set of disclosure policies, thinking of each one as a different public "face" they might wear [Lede 02]. This technique simplifies the definition of multiple permutations of privacy preferences using the metaphor of faces to represent different behaviors during the course of her everyday life. As she encounters a new situation, the user assumes the appropriate face (e.g., secure shopper, cocktail party, hanging out with friends, anonymous, family outings, traveling abroad, etc.). Users can concern themselves primarily with their collection of faces, and less so with the underlying preferences they abstract.

- *Reciprocity* - The strongest advantage of this technique is its simplicity [Raen 05] - user A reveals as much of himself to user B as user B reveals to user A. In spite of its simplicity, it can be surprisingly effective because it mimics a common (most of the times unconsciously) behavior in the real world, when dealing with privacy issues. In fact, it has been shown in several studies that reciprocity is the basis for building trust between two partners [Rubi 75]. However, it is not easy to implement a completely fair reciprocal system. For example, in the Instant Messaging XMPP protocol [Sain 05] you are allowed to see if an user is online *only* if that user also can see that you are online. However, if an user is much more active than the other one, the relation is no longer reciprocal. Another example is the NESSIE system [Prin 99], which claims to be using this technique by disclosing the users who have registered interest to the users producing events - "when you see me I see you". Again, this may not be considered reciprocity - knowing that someone is watching the other doesn't imply that the latter can see his actions. In fact, it would be more appropriate to consider it as a form of *Lookup Notification*.

3.2 Data perturbation

Data perturbation is the most effective technique for community context-aware applications whose primary purpose is gathering statistical information. The idea is to perturb a user's sequence of data values such that (i) the individual data items and their trend (i.e., their changes with time) cannot be estimated without large error, while (ii) the distribution of community data at any point in time, as well as the average community data trend can be estimated with high accuracy [Gant 08].

- *Add noise* - This technique perturbs data by adding noise (useless data) before sending it to the server. This must be done in such a way that it prevents an attacker from breaching the privacy of single users while at the same time, aggregate statistical data is still meaningful and correct. One option to

implement this technique is to add random noise drawn independently either from a know distribution [Agra 00] or from a rotation scheme [Oliv 04], after which a reconstruction algorithm is used to estimate the distribution of the original data. However, randomness may not necessarily imply uncertainty. It has been repeatedly shown that adding random noise to data does not protect privacy [Karg 03]. It is generally easy to reconstruct data from noisy measurements, unless noise is so large that utility cannot be attained from sharing the noisy data. Ganti [Gant 08] tries to overcome this problem by using well-known models from which it derives random data. For example, for a traffic reporting application, a model of vehicular traffic can be used to generate the noise distribution. Since the noise is very similar to the data itself, it is much harder to reconstruct personal data thus defeating privacy attacks.

- *Encryption* - This technique works by encrypting all personal context information with a symmetric key transmitted through a secure channel to everyone that is allowed to consume the information [Putt 10]. To proof that personal information was indeed produced by a given person, information may be digitally signed (e.g. using a private asymmetrical key). This technique inherits all the traditional shortcomings of symmetric key encryption: (i) it is hard to establish a secure channel to transmit the keys; (ii) compromising just one of the consuming devices automatically compromises all the personal information of every person who allowed it to consume their information (since their symmetric keys are stored in the device); (iii) revoking access to someone implies issuing a new symmetric key and retransmitting it to everyone. Encrypting personal context using asymmetrical cryptography would solve some of those problems but would create others. In particular, it would pose scalability problems since personal information would have to encrypted with the public key of every consumer (unlike symmetric cryptography, where you only need to encrypt once irrespective of the number of consumers). If the discloser is willing to share information with a large number of consumers and/or produces a steady stream of events, the continuous encryption would put a high burden on the system, with a negative impact on its performance especially on resource-restrained devices such as cellphones.

- *Chunk replacement* - Mun [Mun 09] proposes to replace chunks of data with synthetic but realistic samples that have a limited impact on the quality of the aggregate analysis. Replacing data is a reasonable alternative to *selective hiding* approaches, that are prone to raising suspicion and thus loosing credibility [Lane 10]. For example, if someone is continuously transmitting his location and than suddenly stops, this can lead others to think that person entered a sensitive location (e.g., his home). Mun solves this problem by replacing chunks instead of hiding them. For example, this technique could be used to replace a location trace segment with another closely related to the original in terms of model output equivalency, based on historical information of the user's likely movements. To be effective and believable, the substitute trace must be credible to the people with whom the user shares his/her data.

- *Blurring* - Blurring refers to disclosing something true but not specific enough to reveal sensitive information and it is a well known human behavior to control privacy [Iach 05]. For example, this happens when someone answers a question with "vague" or "not specific" information to protect his privacy. However, some studies with context-aware applications show that blurring is not commonly used by the participants, which prefer not to disclose any information instead of blurring it [Cons 05]. Still, it is a valid technique derived from the already mentioned "plausible deniability" property.

3.3 Anonymization

Anonymization consists on concealing the real identity of the person associated with some context information. To properly evaluate the effectiveness of an anonymization technique we need an objective metric of a person's anonymity. The most well-know metric is *k-Anonymity* [Swee 02]. It is based on the idea of generalizing a data record until it is undistinguishable from the records of at least k - 1 other individuals. This algorithm applied to location information can be implemented with the following steps [Grut 03]: location information is represented by a tuple containing three intervals *([x1, x2], [y1, y2], [t1, t2])*. The intervals *[x1, x2]* and *[y1, y2]* describe a two dimensional area where the subject is located. *[t1, t2]* describes a time period during which the subject was present in the area. Thus, a location tuple for a subject is *k-anonymous*, when it describes not only the location of the subject, but also the locations of k - 1 other subjects. In other words, k - 1 other subjects also must have been present in the area and the time period described by the tuple. Generally speaking, the larger the anonymity set k is, the higher is the degree of anonymity.

The techniques for anonymizing context information usually work by either: (i) providing a complex multi-hop path between the discloser and the consumer of the information, (ii) replacing the real identity with an incomplete or fake one or (iii) mixing information from multiple related people.

- *Mix routing* - Originally proposed by [Chau 81] to guarantee the anonymity of participants in a electronic mail system, it has since then been applied to various context-aware systems (e.g., AnonySense[Corn 08], Mist [Al M 02]). A mix is a message router that forwards messages in such a way that an adversary cannot match incoming messages to outgoing messages. This is accomplished by numerous techniques such as padding all messages to the same size, encrypting incoming and outcoming messages with different keys or reordering messages. Obviously, this technique implies that an adequate network infrastructure is in place, with multiple nodes between the sender and the receiver, which may not be practical in many applications.

- *Pseudonyms* - Anonymity concerns the dissociation of information about an individual, such as location, from that individual's actual identity. A special type of anonymity is pseudonymity, where an individual is anonymous, but maintains a persistent identity (a pseudonym) [Duck 06]. To reduce the chance of a privacy attack based on the victim's history, [Bere 05] proposes frequently changing pseudonyms.

- *Hitchhiking* - Proposed by [Tang 06], this algorithm is suitable for applications that use location data collected from multiple people to infer statistical information about a given place, such as the number of seats available in a coffee-shop or the number of cars in a bridge. The key idea is that a person doesn't send his location but rather information that she collected at a given location. For example, in a coffee-shop, an application installed in the user's computer continuously scans the Wifi network to determine how many other computers are present. At regular intervals, it reports this count to the server.

- *Temporal-spatial cloaking* - [Grut 03] proposes an implementation of k-Anonymity based on two variables: location and time. Starting with location, it subdivides the area around the subject's position until the numbers of subjects in the area falls below a certain threshold k (thus achieving k-Anonymity), using quadtree algorithms. This is called spatial cloaking, because it reduces spatial

accuracy. It is also possible to maintain spatial accuracy at the expense of reducing temporal accuracy, by delaying the information availability until k subjects have been in a certain area. By combining both dimensions, it is possible to achieve a high level of anonymity without a big loss of precision. However, it is not suitable to situations where accuracy and timeliness are important.

- *Attribute-based authentication* - Some applications associate data with some non-identifiable attributes of the user (e.g., age, gender) instead of the user himself, thus guaranteeing anonymity [Kapa 09]. For example Alice might reveal that she is a "student at Dartmouth" without disclosing her identity.

3.4 Lookup notification

Lookup notification provides awareness of potential privacy breaches by showing who consumed or is consuming context information of a given person. Since it acts *a posteriori* (after a privacy breach) it is usually used to complement the other three types of techniques, detecting misconfigurations or just giving users more confidence with their privacy settings.

- *Alerts* - Applications using this technique provide the discloser of information with immediate visual or audio feedback when someone is consuming that information [Cons 05]. For example, consider a cellphone social application which shows an alert message every time a friend queries the owner's location.

- *Logs* - Since the above mentioned *alerts* are shown in real-time, they can become a source of interruptions leading users to turn them off, defeating its purpose. In that sense, logs represent a less intrusive technique, since they just register all consumptions of personal context information in a log which can be consulted by the information discloser at any time, when he wishes so. On the other hand, if we are in presence of a privacy breach, we will want to reduce the lag between the time of the (undesirable) information consumption and the time of log checking, to prevent further private information leakage.

4 Conclusions

Privacy management is a complex yet critical issue on distributed context-aware systems. Although there has been extensive research on this subject, with the proposal of multiple techniques as shown on this article, we are still far from reaching a complete solution that covers the privacy properties described by Raento [Raen 05], derived from decades of social psychology studies: *Control*, *Accountability*, *Plausible Deniability*, *Reciprocity* and *Utility*. For example, *Privacy Policies* give people the power to control who can consume their personal information, but still in a very rigid immutable way, when Consolvo's studies [Cons 05] show that privacy settings are very dynamic and completely situation-dependent. *Data Perturbation* techniques are, likewise, applied regardless of the situation - encryption or chunk replacement do not take into account the relation between the discloser and the consumer or other environmental variables.

In fact, the rigidity of most techniques is their bigger shortcoming for wide adoption, since they do not mimic human highly dynamic behaviors when it comes to privacy preservation. Still, some techniques manage to get closer to those behaviors as is the case of techniques that try to achieve *k-Anonymity* such as *Spatial-temporal cloaking*, whose dynamics are based on the level of entropy that gives the user sufficient confidence on the preservation of his private data. We increase entropy by aggregating related data, being it the real-word example of only allowing video-vigilance on public (typically crowded) places or location-based applications that only transmit their users location where there are enough users in the vicinity also transmitting their location. Another advantage of techniques that dynamically increase entropy is their implicit support for *Plausible Deniability*, since the interpretation gaps that arise from (incomplete) aggregated information are left to the discloser to fill in with the most appropriate information within the current situation.

In addition, although *Lookup Notification* can not avoid privacy breaches (since they act *a posteriori*), it should not be dismissed as a valid privacy management technique, since it gives users confidence that the other mechanisms in place are in fact protecting their privacy.

Finally, it is noteworthy that highly successful commercial applications such as Facebook and Twitter are aware of the privacy issues that can arise from misuse of their systems but still provide very limited privacy management mechanisms resorting only to cumbersome and error-prone checklists, which are a complete departure from deeply rooted human behaviors, as formed and refined during decades of social adaptation.

References

[Agra 00]   R. Agrawal and R. Srikant. "Privacy-preserving data mining". *ACM Sigmod Record*, Vol. 29, No. 2, pp. 439–450, 2000.

[Al M 02]   J. Al-Muhtadi, R. Campbell, A. Kapadia, and M. Mickunas. "Routing through the mist: privacy preserving communication in ubiquitous computing environments". In: *Proceedings 22nd International Conference on Distributed Computing Systems*, pp. 74–83, IEEE Comput. Soc, 2002.

[Altm 81]   I. Altman, A. Vinsel, and B. Brown. "Dialectic conceptions in social psychology: An application to social penetration and privacy regulation". *Advances in experimental social psychology*, Vol. 14, pp. 107–160, 1981.

[Bere 05]   A. Beresford and F. Stajano. "Location privacy in pervasive computing". *Pervasive Computing, IEEE*, 2005.

[Chau 81]   D. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms". *Communications of the ACM*, Vol. 24, No. 2, pp. 84–90, 1981.

[Cons 05]   S. Consolvo, I. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge. "Location disclosure to social relations: why, when, & what people want to share". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 81–90, ACM, 2005.

[Corn 07] J. Cornwell, I. Fette, G. Hsieh, M. Prabaker, J. Rao, K. Tang, K. Vaniea, L. Bauer, L. Cranor, J. Hong, B. McLaren, M. Reiter, and N. Sadeh. "User-Controllable Security and Privacy for Pervasive Computing". *Eighth IEEE Workshop on Mobile Computing Systems and Applications*, pp. 14–19, March 2007.

[Corn 08] C. Cornelius, A. Kapadia, D. Kotz, and D. Peebles. "AnonySense: Privacy-aware people-centric sensing". *Proceeding of the 6th*, 2008.

[Duck 06] M. Duckham and L. Kulik. "Location privacy and location-aware computing". *English*, 2006.

[Gant 08] R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher. "PoolView: stream privacy for grassroots participatory sensing". In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 281–294, ACM, 2008.

[Gros 05] R. Gross and A. Acquisti. "Information Revelation and Privacy in Online Social Networks". In: *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 71–80, ACM, 2005.

[Grut 03] M. Gruteser and D. Grunwald. "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking". *Proceedings of the 1st international conference on Mobile systems, applications and services - MobiSys '03*, pp. 31–42, 2003.

[Henr 05] K. Henricksen, J. Indulska, and T. McFadden. "Middleware for distributed context-aware systems". *Internet Systems 2005*, pp. 846–863, 2005.

[Iach 05] G. Iachello, I. Smith, S. Consolvo, M. Chen, and G. Abowd. "Developing privacy guidelines for social location disclosure applications and services". In: *Proceedings of the 2005 symposium on Usable privacy and security*, pp. 65–76, ACM, 2005.

[Kapa 07] A. Kapadia, T. Henderson, J. Fielding, and D. Kotz. "Virtual walls: Protecting digital privacy in pervasive environments". *Pervasive Computing*, pp. 1–18, 2007.

[Kapa 09] A. Kapadia, D. Kotz, and N. Triandopoulos. "Opportunistic sensing: Security challenges for the new paradigm". *2009 First International Communication Systems and Networks and Workshops*, pp. 1–10, Jan. 2009.

[Karg 03] H. Kargupta, S. Datta, and Q. Wang. "On the privacy preserving properties of random data perturbation techniques". *Data Mining, 2003.*, 2003.

[Lane 10] N. Lane, E. Miluzzo, H. Lu, D. Peebles, and T. "A Survey of Mobile Phone Sensing". *IEEE Communications Magazine*, No. September, pp. 140–150, 2010.

[Lede 02] S. Lederer, A. Dey, and J. Mankoff. "A Conceptual Model and a Metaphor of Everyday Privacy in Ubiquitous Computing Environments". *Computer*, No. June, 2002.

[Milu 08] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, and H. "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application". In: *6th ACM conference on Embedded network sensor systems*, p. 337, ACM Press, New York, New York, USA, 2008.

11

[Mun 09] M. Mun, P. Boda, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, and R. West. "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research". *Proceedings of the 7th international conference on Mobile systems, applications, and services - Mobisys '09*, p. 55, 2009.

[Oliv 04] S. Oliveira and O. Zaiane. "Achieving privacy preservation when sharing data for clustering". *Secure Data Management*, pp. 67–82, 2004.

[Oyom 09] W. Oyomno, P. Jappinen, and E. Kerttula. "Privacy implications of context-aware services". In: *Proceedings of the Fourth International ICST Conference on COMmunication System softWAre and middlewaRE*, pp. 1–9, ACM, 2009.

[Prin 99] W. Prinz. "NESSIE: an awareness environment for cooperative settings". *ECSCW'99*, pp. 391–410, 1999.

[Putt 10] K. P. N. Puttaswamy and B. Y. Zhao. "Preserving privacy in location-based mobile social applications". *Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications - HotMobile '10*, p. 1, 2010.

[Raen 05] M. Raento and A. Oulasvirta. "Privacy management for social awareness applications". *Proceedings of 1st Workshop on Context Awareness for Proactive Systems (CAPS 2005)*, Vol. 1, pp. 105–114, July 2005.

[Rubi 75] Z. Rubin. "Disclosing oneself to a stranger: Reciprocity and its limits1". *Journal of Experimental Social Psychology*, Vol. 11, No. 3, pp. 233–260, 1975.

[Sain 05] P. Saint-andre and R. Meijer. "Streaming XML with Jabber/XMPP". *IEEE Internet Computing*, Vol. 9, No. 5, pp. 82–89, Sep. 2005.

[Shir 03] C. Shirky. "A group is its worst enemy". `http://www.shirky.com/writings/group\_enemy.html`, 2003.

[Swee 02] L. Sweeney. "k-ANONYMITY: A model for protecting privacy". *World*, Vol. 10, No. 5, pp. 557–570, 2002.

[Tang 06] K. Tang, P. Keyani, J. Fogarty, and J. Hong. "Putting people in their place: An anonymous and privacy-sensitive approach to collecting sensed data in location-based applications". *Proceedings of the SIGCHI*, 2006.

[Weis 95] M. Weiser. "The computer for the 21st century". *Scientific American*, 1995.

[West 67] A. Westin. *Privacy and freedom.* Vol. 97, New York Atheneum, 1967.