Chapter 4
Privacy

Abstract The issue of privacy is well known and, for obvious reasons, distributed context-aware systems are not an exception. As a matter of fact, people are sensitive about revealing their location or activities (and other types of context information) which are often transmitted by such context-aware systems. In addition, such transmission is often done without requiring a specific user action, in order to increase the usability of such systems. In this chapter we define what privacy means within the realm of distributed context-aware systems; then, we describe the several privacy management techniques available: privacy policies, data perturbation, anonymization, and lookup notification.

4.1 Introduction

As already mentioned, in recent years, we have been watching a tremendous growth of available personal sensing devices such as the iPhone and Android mobile phones. 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 [96], 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 such context-aware systems often transmit these and other types of context information without requiring a specific user action, in order to increase their usability. For example, CenceMe [59] transmits periodically the current user's location to a group of friends. It would not 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 their friends will probably reject opening up this information to the whole community, because they do not trust the recipients of that information. Still,

P. Ferreira and P. Alves, *Distributed Context-Aware Systems*, SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-04882-6_4, © The Author(s) 2014

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 [10, 90]), we analyze privacy management techniques that are suitable for all kinds of context-aware systems.

4.2 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 [98] 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 [63] 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 [5]. These studies also provided the background that allowed Raento [72] 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 [28] showed 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 [45] 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 a clear component of mediated disclosure. For example, teenagers do not always answer their mobile phones, claiming that they did not 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 reciprocity in self-disclosure between partners is necessary for building of trust and deepening of relationship [78].
- **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.

4.3 Privacy Management Techniques

Privacy management techniques can be categorized into four main types (see Fig. 4.1): 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

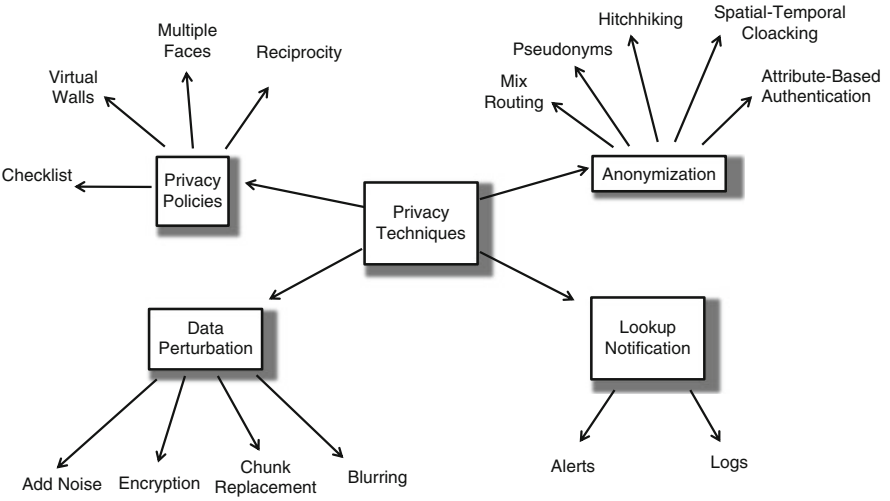

Fig. 4.1 Privacy management techniques

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 viewpoint. 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 [30].

- **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 to provide multiple explanations to fill in the gaps without losing credibility.
- **Anonymization**—Using this technique, the information is delivered intact to context consumers except for its author, which is removed or replaced with one that cannot be used to infer the real author. This technique is very 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) [89] and is being dismissed by recent social applications such as Facebook, Twitter, and LinkedIn. 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 (i.e., 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 now detail each of these categories describing their specific techniques.

4.3.1 Privacy Policies

Privacy policies consist of 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. There are several approaches as follows (see Fig. 4.1): checklist, virtual walls, multiple faces, and reciprocity.

4.3.1.1 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 neither 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 [42] found that only 1.2 % of Facebook users changed the default privacy settings for profile searchability.

4.3.1.2 Virtual Walls

Kapadia [51] proposes this technique as a solution to user difficulties in defining fine-grained privacy policies. The idea is to set up user-defined policies based on 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*. Sensor 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). Then, a simple matrix is used to match these two dimensions: transparent walls propagate all footprints, translucent walls propagate only general footprints, and opaque walls do not 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.

4.3.1.3 Multiple Faces

Users predefine a small set of disclosure policies, thinking of each one as a different public "face" they might wear [56]. This technique simplifies the definition of multiple permutations of privacy preferences using the metaphor of faces to represent different behaviors during the course of a user's everyday life. As the user encounters a new situation, he 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.

4.3.1.4 Reciprocity

The strongest advantage of this technique is its simplicity [72]—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 [78]. However, it is not easy to implement a completely fair reciprocal system. For example, in the instant messaging XMPP protocol [79], you are allowed to see if a user is online *only* if that user also can see that you are online. However, if a user is much more active than the other one, the relation is no longer reciprocal. Another example is the NESSIE system [67], which claims to be using this technique by disclosing the users who have registered interest to the user producing events—"when you see me I see you." Again, this may not be considered reciprocity—knowing that someone is watching the other does not imply that the latter can see his actions. In fact, it would be more appropriate to consider it as a form of *lookup notification*.

4.3.2 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 [41]. There are several approaches as follows (see Fig. 4.1): add noise, encryption, chunk replacement, and blurring.

4.3.2.1 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, aggregated statistical data is still meaningful and correct. One option to implement this technique is to add random noise drawn independently either from a known distribution [2] or from a rotation scheme [61], 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 [53]. 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 [41] 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.

4.3.2.2 Encryption

This technique works by encrypting all personal context information (e.g., with a symmetric key) transmitted through a secure channel to everyone that is allowed to consume the information [70]. To proof that personal information was indeed produced by a given person, information may be digitally signed (e.g., using a private asymmetrical key). Symmetric cryptography inherits all the traditional shortcomings associated to the use 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); and (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 be 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-constrained devices such as mobile phones.

4.3.2.3 Chunk Replacement

Mun[60] proposes to replace chunks of data with synthetic but realistic samples that have a limited impact on the quality of the aggregated analysis. Replacing data is a reasonable alternative to *selective hiding* approaches that are prone to raising suspicion and thus losing credibility [55]. For example, if someone is continuously transmitting his location and then 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 data.

4.3.2.4 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 [48]. 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 users, who prefer not to disclose any information instead of blurring it [28]. Still, it is a valid technique derived from the already mentioned "plausible deniability" property.

4.3.3 Anonymization

Anonymization consists of 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-known metric is *k-anonymity* [92]. It is based on the idea of generalizing a data record until it is indistinguishable from the records of at least $k - 1$ other

individuals. This algorithm applied to location information can be implemented with the following steps [43]: 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 during the time period described by the tuple. Generally speaking, the larger the anonymity set k is, the higher the degree of anonymity is.

The techniques for anonymizing context information usually work by either (see Fig. 4.1) (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.

4.3.3.1 Mix Routing

Originally proposed by [21] 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[29], Mist [3]). 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.

4.3.3.2 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 pseudonyms, where an individual is anonymous, but maintains a persistent identity (a pseudonym) [35]. To reduce the chance of a privacy attack based on the victim's history, Beresford [10] proposes frequently changing pseudonyms.

4.3.3.3 Hitchhiking

Proposed by Tang [93], 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 does not send his location but rather information that he 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.

4.3.3.4 Spatial-Temporal Cloaking

Gruteser [43] 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 number of subjects in the area falls below a certain threshold k (thus achieving k-anonymity), using quad-tree 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.

4.3.3.5 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 [52]. For example, Alice might reveal that she is a "student at Dartmouth" without disclosing her identity.

4.3.4 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. There are basically two approaches (see Fig. 4.1): alerts and logs.

4.3.4.1 Alerts

Applications using this technique provide the discloser of information with immediate visual or audio feedback when someone is consuming that information [28]. For example, consider a mobile phone social application which shows an alert message every time a friend queries the owner's location.

4.3.4.2 Logs

Since the abovementioned *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.

4.4 Summary

This chapter starts with some considerations regarding the definition of privacy in distributed context-aware systems, emphasizing the properties that must be considered: solitude, anonymity, and reservation. In addition, when taking into account a psychological view of the issue, we see that other important properties appear: control, accountability, plausible deniability, reciprocity, and utility. These lead us to propose a taxonomy based on the following main techniques: privacy policies, data perturbation, anonymization, and lookup notification. For each one of these techniques we described the specific solutions that can be found in the literature.