Privacy in Georeferenced Context-aware Services: A Survey

Daniele Riboni, Linda Pareschi, and Claudio Bettini

EveryWare Lab, D.I.Co., University of Milano
via Comelico 39, I-20135 Milano, Italy
{riboni,pareschi,bettini}@dico.unimi.it

Abstract. Location based services (LBS) are a specific instance of a broader class of Internet services that are predicted to become popular in a near future: context-aware services. The privacy concerns that LBS have raised are likely to become even more serious when several context data, other than location and time, are sent to service providers as part of an Internet request. This paper provides a classification and a brief survey of the privacy preservation techniques that have been proposed for this type of services. After identifying the benefits and shortcomings of each class of techniques, the paper proposes a combined approach to achieve a more comprehensive solution for privacy preservation in georeferenced context-aware services.

1 Introduction

It is widely recognized that the success of context-aware services is conditioned to the availability of effective privacy protection mechanisms (see, e.g., [1, 2]). Techniques for privacy protection have been thoroughly studied in the field of databases, in order to protect microdata released from large repositories. Recently some of these techniques have been extended and integrated with new ones to preserve the privacy of users of Location Based Services (LBS) against possibly untrusted service providers as well as against other types of adversaries [3]. The domain of service provisioning based on location and time of request introduces novel challenges with respect to traditional privacy protection in microdata release. This is mainly due to the dynamic nature of the service paradigm which requires a form of *online* privacy preservation technique as opposed to an *offline* one used, for example, in the publication of a view from a database. In the case of LBS, specific techniques are also necessary to process the spatio-temporal information describing location and time of request which is also very dynamic. On the other hand, location and time are only two of the possibly many parameters characterizing the context of an Internet service request. Indeed, context information goes far beyond location and time, including data such as personal preferences and interests, current activity, physiological and emotional status, and data collected from body-worn or environmental sensors, just to name a few. Privacy protection techniques specifically developed for LBS are often insufficient and/or inadequate when applied to generic context-aware services.

Consider, for instance, cryptographic techniques proposed for LBS (e.g., [4, 5]). These techniques provide strong privacy guarantees at the cost of high computational overhead on both the client and server side; moreover, they introduce expensive communication costs. Hence, while they may be profitably applied to simple LBS such as nearest neighbor services, it is unlikely that they would be practical for complex context-aware services. On the other hand, obfuscation techniques proposed for LBS (e.g., [6, 7]) are specifically addressed to location information; hence, those techniques cannot be straightforwardly applied to other contextual domains. With respect to techniques based on identity anonymity in LBS (e.g., [8, 9]) we point out that, since many other kinds of context data besides location may help an adversary in identifying the owner of those data, the amount of context data to be generalized in order to enforce anonymity is large. Hence, even if filtering techniques can be used for improving the service response, it could happen that in order to achieve the desired anonymity level, context data become too general to provide the service at an acceptable quality level [10]. For this reason, specific anonymity techniques for generic context-aware services are needed.

Moreover, in pervasive computing environments context-aware services can exploit data provided by sensors deployed in the environment that can constantly monitor context data. Hence, if those context sources are compromised, an adversary's inference abilities may increase taking advantage of the observation of users' behavior and of up-to-date context information. Defense techniques for privacy preservation proposed for LBS do not consider this kind of inference capabilities, since location and time are the only contextual parameters that are taken into account. As a result, protecting against the above mentioned kind of attacks requires new techniques.

In this paper we survey privacy protection techniques for georeferenced context-aware services. As depicted in Figure 1, the general privacy threat we are facing is the release of *sensitive associations* between a user's identity and the information that she considers private. The actual privacy risk certainly depends on the adversary's model; for the purpose of this survey, unless we mention specific attacks, we adopt the general assumption that an adversary may obtain service requests and responses as well as publicly available information.

We distinguish different types of defense techniques that can be used to contrast the privacy threat.

- ◦ **Network and cryptographic protocols.** These are mainly used to avoid that an adversary can access the content of a request or response while it is transmitted as well as to avoid that a network address identifies the location and/or the issuer of a request.
- ◦ **Access control mechanisms.** These are used to discriminate (possibly based on context itself) the entites that can obtain certain context information.

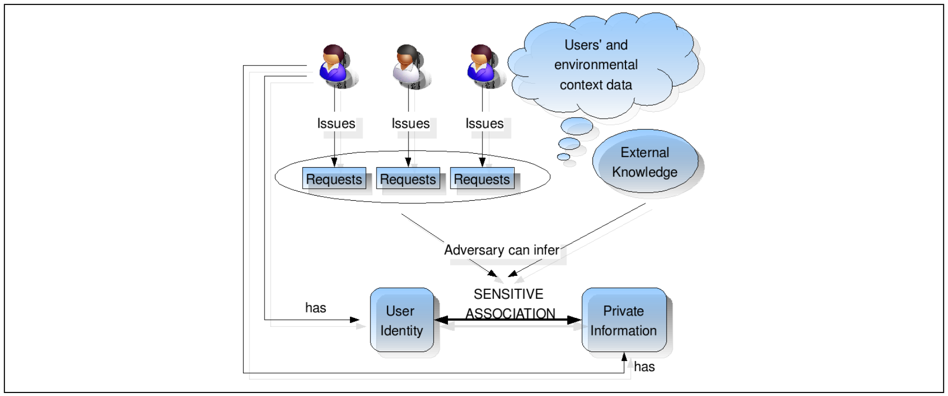

Fig. 1. The privacy threat

- **Obfuscation techniques.** Under this name we group the techniques, usually based on generalization or partial suppression, that limit the disclosure of private information contained in a request. Intuitively, they control the release of the second part of the association describing the privacy threat.
- **Identity anonymization techniques.** These are techniques that aim at avoiding the release of the first part of the association, i.e., the identity of the issuer. The goal is to make the issuer indistinguishable among a sufficiently large number of individuals.

This classification may apply as well to defenses against LBS privacy threats, however our description of available approaches and solutions will be focused on those for more complex context-aware services. Sections 2, 3, 4, and 5 address each of the above types of defenses, respectively. Based on the weaknesses emerged from the analysis of the existing techniques, in Section 6 we advocate the use of a combined approach, present preliminary proposals, and illustrate the general characteristics that a comprehensive combined approach may have. Section 7 concludes the paper.

2 Network and cryptographic protocols

The development of context-aware services received impulse by technological progresses in the area of wireless communications, mobile devices, and sensors. The use of wireless channels, and more generally insecure channels, poses a first threat for the users' privacy since it makes easier for an adversary to acquire service requests and responses by eavesdropping the communication or analyzing traffic on the network. In the literature, several models have been proposed for privacy preservation in context-aware systems. While some of them rely on a *centralized* architecture with a single trusted entity in charge of ensuring the users' privacy, other models rely on a *decentralized* architecture in which mobile devices use direct communication channels with service providers. *decentralized* architectures in which mobile communication channels with service providers. In both cases, two natural countermeasures for privacy attacks are: a) implement secure communication channels so that no third party can obtain requests/responses while they are in transit, and b) avoid the recognition of the client's network address, even by the service provider, which may be untrusted.

In order to protect point-to-point communications, in addition to standard wireless security, different cryptographic techniques can be applied. One possibility is clearly for applications to rely on SSL to encrypt communication; an alternative (or additional) possibility is to provide authentication, authorization and channel encryption through systems like Kerberos ([11]). Kerberos is based on a centralized entity, *Key Distribution Center* (KDC), in charge of authenticating clients and servers in the network, and providing them with the keys needed for encrypting the communications. The centralized model that inspires Kerberos does not protect from attacks aimed at acquiring the control of the KDC entity. Specific solutions to communication protection also depend on the considered architecture and adversary's model, and are outside the scope of this paper.

Different approaches ([12, 13]) aim at guaranteeing a certain degree of anonymity working at the IP level. The Tarzan system ([12]) adopted a solution based on a network overlay that clusters nodes in subnetworks called *domains* on the base of their IP addresses. The IP hiding is achieved by the substitution of the sender's IP with the pseudonym corresponding to its domain. Moreover, when a node needs to send a packet, its communications are filtered by a special server called *mimic* that is in charge of *i)* substituting the IP and other information that could reveal the sender identity with the adequate pseudonym, and *ii)* of setting a virtual path (*tunnel*) that guarantees the communication encryption.

Most solutions presented in the literature apply a combination of routing protocols for IP hiding, and cryptographic techniques ([14]) to protect from eavesdropping over the communication channel. Onion Routing ([15]) implements both the features of IP hiding and message encryption. In order to preserve the sender's IP address, each message travels towards the receiver via a series of proxies, called *onion routers*, which choose the next component of the path setting an unpredictable route. Each router in the path re-encrypts the message before forwarding it to the next router. However, even these solutions suffer from attacks aimed at acquiring the control of one or more nodes of the network.

A different application of a privacy-preserving routing protocol is presented in [16]: the proposed solution has been designed for protecting the user's privacy while moving in smart environments. This solution is based on a hierarchy of trusted servers where the leaves, called *portals*, are aware of the user's location, while internal nodes are aware of services provided by the environment. The user accesses the network through a portal and, according to her privacy preferences, she is assigned to an internal node, called *lighthouse*, that has the task of filtering and encrypting all the communications between the user and the service provider. The lighthouse does not know the user's position but is aware of the next hop in the server hierarchy composing the path to the user's portal. Similarly, the portal does not know which service the user is asking for, but it is aware of the path to the chosen lighthouse. The privacy preservation is achieved decoupling position data from both the identity information and other context parameters. However, this approach requires the servers in the hierarchy to be trusted and it does not protect by privacy attacks performed acquiring the control of one of the nodes in the structure.

The use of cryptographic techniques can also be extended to hide from the service provider the exact request parameters as well as the response. This approach has been proposed in the area of LBS where location information is often considered sensitive by users. In particular, solutions based on this approach aim at retrieving the nearest neighbor (NN) point of interest (*poi*) with respect to the user position at the time of the request.

A first solution was proposed in [4]: the authors propose a form of encrypted query processing combining the use of a data structure suited for managing spatial information with a cryptographic schema for the secret sharing. On the server side, location data are handled through a *directed acyclic graph* (DAG), whose nodes correspond to Voronoi regions obtained by a tessellation of the space with respect to *poi*s stored by the service provider. The query processing is performed according to the protocol proposed in [17] that allows a client to retrieve the correct Voronoi area without communicating its precise location. The drawback of this solution is that, in order to resolve a NN query, the user needs to send a number of queries that is proportional to the depth of the DAG instead of a single request. The consequent communication overhead impacts on the network traffic and on the response time, which are commonly considered important factors in mobile computing.

Recently, a cryptographic approach inspired by the Private Information Retrieval (PIR) field was proposed in [5]. The service provider builds a Voronoi tessellation according to the stored *poi*s, and superimposes on its top a regular grid of arbitrary granularity. In order to obtain the response to a NN query the privacy preservation mechanism relies on a PIR technique that is used for encrypting the user query, and for retrieving part of the location database without revealing spatial information. Some of the strong points of this solution are that location data are never disclosed; the user's identity is confused among identities of all users; and no trusted third party is needed to protect the users' privacy. However, since mobile devices are often characterized by limited computational capability, the query encryption and the answer processing performed at the client side have a strong impact on service response time, network and power consumption. In particular, when applied to context-aware services that perform the adaptation on a wide set of heterogeneous context data, this technique may result in unacceptable computation overhead both at the client and at the server side.

3 Access control in context-aware systems

Pervasive computing environments claim for techniques to control release of data and access to resources on the basis of the context of users, environment, and hardware/software entities. In general, the problem of access control [18] consists in deciding whether to authorize or not a requesting entity (*subject*) to perform a given *action* on a given resource (*object*). Access control mechanisms have been thoroughly studied in many fields, including operating systems, databases, and distributed systems. However, the characteristic features of pervasive environments introduce novel issues that must be taken into account for devising effective access control mechanisms. In particular, differently from centralized organizational domains, pervasive environments are characterized by the intrinsic decentralization of authorization decisions, since the object owners (users, services, infrastructures) are spread through the environment, and may adopt different policies regarding disclosure of private information. Hence, specific techniques to deal with the mobility and continuously changing context of the involved entities are needed to adapt authorizations to the current situation.

To this aim various techniques for context-aware access control have been recently proposed. Context-aware access control strategies fall in two main categories. The first category is the one of techniques aimed at granting or denying access to resources considering the context of the requesting user and of the resource (see, e.g., [19–21]). The second category is the one of techniques aimed at controlling the release of user's context data on the basis of the context of the requesting entity and of the user herself. In this section we concentrate on techniques belonging to the latter category. On the contrary, techniques belonging to the former category are outside the scope of this paper, and will not be reviewed; however, we point out that, since those techniques imply the release of users' context data to the access control mechanism, generally they also adopt strategies to enforce users' privacy policies.

Proposed context-aware access control mechanisms can be roughly classified in those that derive from *discretionary (DAC)* [22] and those that derive from *role-based (RBAC)* [23] access control. In DAC systems, the owner of each object is in charge of stating policies to determine the access privileges on the basis of the subject identity. These techniques are well suited to domains in which subjects do not belong to a structured organization (e.g., they are well suited to generic Internet services), since they are released from the burden of managing groups or roles of subjects. On the other hand, techniques based on RBAC (in which the access privileges depend on the subject role) are well suited to structured organization domains (like, e.g., hospitals, companies), since the definition of functional roles simplifies the management of access control policies.

Other techniques related to access-control in context-aware systems include the use of access-rights graphs and hidden constraints (e.g., [24]) as well as *zero-knowledge proof theory* [25] (e.g., [26]). These are called *secret authorization* mechanisms, since they allow an entity to certify to a verifier the possession of private information (e.g., context data) revealing neither the authorization policies nor the secret data.

In the following we briefly describe the access control techniques for context-awareness derived from DAC and RBAC models, respectively.

Techniques derived from DAC. Even early approaches to discretionary access control allowed the expression of conditions to constrain permissions on the basis of the spatial and temporal characterization of the subject. For instance, in a bank setting, access to customer accounts could be acknowledged to authorized personnel only during working hours and from machines located within the bank. More recently, access control techniques specifically addressed to the protection of location information (e.g., [27]) have been proposed. However, the richness and dynamics of contextual situations that may occur in pervasive and mobile computing environments claim for the definition of formal languages to express complex conditions on a multitude of context data, as well as sufficiently expressive languages to represent the context itself. To this aim, *Houdini* [28] provides a comprehensive formal framework to represent dynamic context data, integrate them from heterogeneous sources, and share context information on the basis of users' privacy policies. In particular, privacy policies can be expressed considering the context of the data owner (i.e., the user) and the context of the subject. As an example, a user of a service for locating friends could state a policy to disclose her current location to her friends only if her mood is *good* and her current activity is not *working*. Privacy policies in *Houdini* are expressed in a restricted logic programming language supporting rule chaining but no cycles. Rules preconditions express conditions on context data, while postconditions express permissions to access contextual information; reasoning with the resulting language has low computational complexity. Policy conflict resolution is based on explicit rule priorities.

Another relevant proposal, specifically addressed to the preservation of mobile customers privacy, can be found in [29]. That work proposes an access control system aimed at controlling the release of private data based on time, location, and customer's preferences. For instance, a user could state a policy to disclose her location and profile information only during the weekend and if she is in a mall, and only in exchange for a discount coupon on items in her shopping list. The proposed solution is based on an intermediary infrastructure in charge of managing location and profiles of mobile users and to enforce their privacy policies. A specific index structure as well as algorithms are presented to efficiently enforce the proposed techniques.

Techniques derived from RBAC. Many other existing approaches to context-aware access control are based on an extension of the RBAC model. As anticipated before, RBAC systems are well-suited to structured organization domains. However, the baseline RBAC model is not adequate to pervasive and mobile computing domains, which are characterized by the dynamics of situations that may determine the role played by a given entity in a given context. For this reason, various proposals have been made to extend RBAC policies with contextual conditions (see, e.g., [19]), and in particular with spatio-temporal constraints (e.g., [30]). More recently, this approach has been applied to the privacy protection of personal context data. A proposal in this sense is provided by the *UbiCOSM* middleware [31], which tackles the comprehensive issue with mechanisms to secure the access not only to services provided by ubiquitous infrastructures, but also to users' context data, based on contextual conditions and roles. The context model of UbiCOSM distinguishes between the *physical* dimension, which describes the spatial characterization of the user, and the *logical* dimension, which describes other data such as the user's current activity and device capabilities. For instance, the context *TouristAtMuseum* is composed by the physical context *AtMuseum* (characterized by the presence of the user within the physical boundaries of a museum) and by the logical context *Tourist* (which defines the user's role as the one of a tourist). Users can declare a policy to control the release of a personal context data as the association between a permission and a context in which the permission applies. Simple context descriptions can be composed in more complex ones by means of logical operators, and may involve the situation of multiple entities. For instance, in order to find other tourists that share her same interests, a user could state a policy to disclose her cultural preferences to a person only if their current context is *TouristAtMuseum* and they are both co-located with a person that is a friend of them both.

Another worth-mentioning system is *CoPS* [32], which provides fine-grained mechanisms to control the release of personal context data, as well as techniques to identify misuse of the provided information. In particular, policies in CoPS are organized in a hierarchical manner, on the basis of the priority level of the policy (i.e., organization-level, user-level, default). Permissions depend on the context and the role of the subject. CoPS supports both administrator and user-defined roles. While the former reflect the hierarchical structure of the organization, the latter can be used to categorize entities in groups, in order to simplify the policy management by users. The system adopts a conflict resolution mechanism based on priorities and on the specificity of access control rules. Moreover, a trigger mechanism can be set up to control the release of particular context data against the frequency of the updates; this technique can be used, for instance, to notify the user in the case someone tries to track her movements by continuously polling her location.

Open issues and remarks. As emerged from the above analysis of the state-of-the-art, the main strong point of techniques derived from DAC consists in the efficiency of the reasoning procedures they employ to evaluate at run-time the access privileges of the requesting entity. This characteristic makes them very well suited to application domains characterized by strict real-time requirements, like telecommunication and Internet services. On the other hand, the roles abstraction adopted by techniques derived from RBAC can be profitably exploited not only in structured organizational domains but also in open environments (like ambient intelligence systems), since heterogeneous entities can be automatically mapped to predefined roles on the basis of the contextual situation to determine their access privileges.

Nevertheless, some open issues about context-aware access control systems are worth to be considered. In particular, like in generic access control systems, a formal model to represent policies and automatically recognize inconsistencies (especially in systems supporting the definition of negative authorizations) is needed; however, only part of the techniques proposed for context-aware computing face this issue. This problem is further complicated by the fact that the privacy policy of a subject may conflict with the privacy policy of an object owner. Proposed solutions for this issue include the use of techniques for secret authorization, like proposed in [24]. Moreover, an evident weakness of these systems consists in their rigidity: if strictly applied, an access control policy either grants or denies access to a given object. This weakness is alleviated by the use of obfuscation techniques (reported in Section 4) to disclose the required data at different levels of accuracy on the basis of the current situation.

A further critical issue for context-aware access control systems consists in devising techniques to support end users in self-defining privacy policies. Indeed, manual policy definition by users is an error-prone and tedious task. For this reason, straightforward techniques to support users' policy definition consists in making use of user friendly interfaces and default policies, like in Houdini and in CoPS, respectively. However, a more sophisticated strategy to address this problem consists in the adoption of statistical techniques to automatically learn privacy policies on the basis of the past decisions of the user. To this aim, [33] propose the application of rough set theory to extract access control policies based on the observation of the user's interaction with context-aware applications during a training period.

As a final remark, we point out that context-aware access control systems do not protect privacy in the case the access to a service is considered a private information by itself (e.g., because it reveals particular interests or habits about the user). To address this issue, techniques aimed at enforcing anonymity exist and are reviewed in Section 5.

4 Obfuscation of context data

In some cases, the strict application of access control mechanisms (i.e., either deny or allow access to a given context data in a given situation) may be a too rigid strategy. For instance, consider the user of a service that redirects incoming calls and messages on the basis of the current activity. Suppose that the service is not completely trusted by the user; hence, since she considers her current activity (e.g., *MeetingCustomers*) a sensitive information, whether to allow or deny the access to her precise current activity may be unsatisfactory. Indeed, denying access to that data would determine the impossibility to take advantage of that service, while allowing access could result in a privacy violation. In this case, a more flexible solution is to *obfuscate* [34] the private data before communicating it to the service provider in order to decrease the sensitivity level of the data. For instance, the precise current activity *MeetingCustomers* could be obfuscated to the more generic activity *BusinessMeeting*. This solution is based on the intuition that each private data is associated to a given sensitivity level, which depends on the precision of the data itself; generally, the lesser the data is precise, the lesser it is sensitive. Obfuscation techniques have been applied to the protection of microdata released from databases (e.g., [35]).

Several techniques based on obfuscation have also been proposed to preserve the privacy of users of context-aware services. These techniques are generally coupled with an access control mechanism to tailor the obfuscation level to be enforced according to the trustiness of the subject and to the contextual situation. However, in this section we concentrate on works that specifically address context data obfuscation. The main research issue in this field is to devise techniques to provide adequate privacy preservation while retaining the usefulness of the data to context-awareness purposes. We point out that, differently from techniques based on anonymity (reviewed in Section 5), techniques considered in this section do not protect against the disclosure of the user's identity.

Various obfuscation-based techniques to control the release of location information have been recently proposed (see, e.g., [36, 6, 7]), based on generalization or perturbation of the precise user's position. One of the first attempts to support privacy in generic context aware systems through obfuscation mechanisms is *semantic eWallet* [37], an architecture to support context-awareness by means of techniques to retrieve users' context data while enforcing their privacy preferences. Users of the semantic eWallet may express their preferences about the accuracy level of their context data based on the requester's identity and on the context of the request. That system supports both *abstraction* and *falsification* of context information. By abstraction, the user can decide to generalize the provided data, or to omit some details about it. For instance, a user involved in a *BusinessMeeting* could decide to disclose her precise activity to a colleague only during working hours and if they both are located within a company building; activity should be generalized to *Meeting* in the other cases. On the other hand, by falsification the user can decide to deliberately provide false information in order to mask her precise current context in certain situations. For instance, a CEO could reveal to her secretary that she is currently *AtTheDentist*, while telling to the other employees that she is involved in a *BusinessMeeting*. In the semantic eWallet, context data are represented by means of ontologies. Obfuscation preferences are encoded as rules whose preconditions include a precise context data and conditions for obfuscation, and postconditions express the obfuscated context data to be disclosed if the preconditions hold.

While in the semantic eWallet the mapping between precise and obfuscated information must be explicitly stated case-by-case, a more scalable approach to the definition of obfuscation preferences is proposed in [38]. That work copes with the multi-party ownership of context information in pervasive environments by proposing a framework to retrieve context information and distributing it on the basis of the obfuscation preferences stated by the data owner. It is worth to note that in the proposed framework the owner of the data is not necessarily the actual proprietary of the context source; instead, the data owner is the person whom the data refers to. For instance, the owner of data provided by a server-side positioning system is the user, not the manager of the positioning infrastructure; hence, the definition of obfuscation preferences about personal location is left to the user. Obfuscation preferences are expressed by conditions on the current context, by specific context data, and by a maximum detail level at which that data can be disclosed in that context. The level of detail of a context data refers to the specificity of that data according to a predefined *obfuscation ontology*. Context data in an obfuscation ontology are organized as nodes into a hierarchy, such that parent nodes represent more general concepts with respect to their children; e.g., the activity *MeetingCustomers* has parent activity *BusinessMeeting*, which in turn has parent activity *Working*. For instance, an obfuscation preference could state to disclose the user's current activity with a *level 2* specificity in the case the requester is *Bob* and the request is made during *working hours*. In the case those conditions hold, the released data is calculated by generalizing the exact current activity up to the second level of the *Activity* obfuscation ontology (i.e., up to the level of the grandchildren of the root node), or to a lower level if the available information is less specific than that stated by the preference. Since manually organizing context data in an obfuscation ontology could be unpractical, a technique to automatically discover reasoning modules able to derive the data at the required specificity level is also presented.

Based on the consideration that the quality of a context information (*QoC*, intended as its closeness to the physical reality it describes) is a strong indicator of privacy sensitiveness, Sheikh et al. propose the use of QoC to enforce users' privacy preferences [39]. In that work, the actual quality of the disclosed context data is negotiated between service providers and users. When a service provider needs a data regarding a user's context, it specifies the QoC that it needs for that data in order to provide the service. On the other hand, the user specifies the maximum QoC she is willing to disclose for that data in order to take advantage of the service. Service requirements and user's privacy preferences are communicated to a middleware that is in charge of verifying if they are incompatible (i.e., if the service requires a data to a quality the user is not willing to provide). If this is not the case, obfuscation mechanisms are applied on that data in order to reach the quality level required by the service provider. QoC is specified on the basis of five indicators, i.e., precision, freshness, spatial and temporal resolution, and probability of correctness. Each context data is associated with five numerical values that express the quality of the data with respect to each of the five indicators. Given a particular context situation, a user can specify her privacy preferences for a context data by defining the maximum quality level for each of the five indicators that she is willing to disclose in that situation. For instance, the user of a remote health monitoring service could state to disclose vague context information to the caregivers when in a non-emergency context, while providing accurate data in the case of emergency.

One inherent weakness of obfuscation techniques for privacy in context-awareness is evident: if the service provider requires a context data to a quality that the user is not willing to disclose, access to that service is not possible. In order to overcome this issue, anonymization techniques (presented in Section 5) have been proposed, which protect from the disclosure of the user's identity, while possibly providing accurate context information.

5 Identity anonymization techniques

While obfuscation techniques aim at protecting the right-hand side of the sensitive association (SA) (see Figure 1), the goal of techniques for identity anonymization is to protect the left-hand side of the SA in order to avoid that an adversary re-identifies the issuer of a request.

In the area of database systems, the notion of k-anonymity has been introduced [40] to formally define when, upon release of a certain database view containing records about individuals, for any specific sensitive set of data in the view, the corresponding individual can be considered indistinguishable among at least k individuals. In order to enforce anonymity it is necessary to determine which attributes in a table play the role of *quasi-identifiers* (*qi*), i.e., data that joined with external knowledge may help the adversary to restrict the set of candidate individuals. Techniques for database anonymization adopt generalization of *qi* values and/or suppression of records in order to guarantee that the set of released records can be partitioned in groups of at least k records having the same value for *qi* attributes (called *qi-groups*). Since each individual is assumed to be the respondent of a single record, this implies that there are at least k candidate respondents for each released record.

The idea of k-anonymity has also been applied to define a privacy metric in location based services, as a specific kind of context-aware services [8]. In this case, the information being released is considered the information in the service request. In particular, the information about the user's location may be used by an adversary to re-identify the issuer of the request if the adversary has access to external information about users' location. Attacks and defense techniques in this context have been investigated in several papers, among which [8, 9]. Moreover, a formal framework for the categorization of defense techniques with respect to the adversary's knowledge assumptions has been proposed in [3]. According to that categorization, when the adversary performs his attack using information contained in a single request the attack is said to be *single-issuer*; otherwise, when the adversary may compare information included in requests by multiple users, the attack is said to be *multiple-issuers*. Moreover, cases in which the adversary can acquire information only during a single time granule are called static (or *snapshot*), while contexts in which the adversary may observe multiple requests issued by the same users in different time granules are called dynamic (or *historical*). A possible technique to enforce anonymity in LBS is to generalize precise location data in a request to an area including a set (called *anonymity set* [41]) of other potential issuers. An important difference between the anonymity set in service requests and the *qi-group* in databases is that while the *qi-group* includes only identities actually associated to a record in the table, the anonymity set includes also users that did not issue any request but that are potential issuers with respect to the adversary's external knowledge.

With respect to identity anonymization in generic context-aware systems, it is evident that many other kinds of context data besides location may be considered *qi*. Hence, a large amount of context data must be generalized in order to enforce anonymity. As a consequence, the granularity of generalized context data released to the service provider could be too coarse to provide the service at an acceptable quality level. In order to limit the information loss due to the generalization of context data, four different personalized anonymization models are proposed in [42]. These models allow a user to constrain the maximum level of location and profile generalization still guaranteeing the desired level of anonymity. For instance, a user could decide to constrain the maximum level of location generalization to an area of $1\,km^2$, while imposing no constraints on the level of generalization of her profile.

As outlined in the introduction, sensing technologies deployed in pervasive environments can be exploited by adversaries to constantly monitor the users' behavior, thus exposing the user to novel kinds of privacy attacks, like the one presented in [43]. In that work it is shown that even enforcing k-anonymity, in particular cases the attacker may recognize the actual issuer of a service request by monitoring the behavior of the potential issuers with respect to service responses. For example, consider a pervasive system of a gym, suggesting exercises on the basis of gender, age, and physiological data retrieved from body-worn sensors. Even if users are anonymous in a set of k potential issuers, the attacker can easily recognize the issuer of a particular request if she starts to use in a reasonable lapse of time a machine the system suggested to her, which was not suggested to any other potential issuer. The proposed solution relies on an intermediary entity that filters all the communications between users and service providers, calculates the privacy threats corresponding to possible alternatives suggested by the service (e.g., the next exercise to perform), and automatically filters unsafe alternatives.

A further issue to be considered is the defense against the well-known problem of *homogeneity* [44] identified in the field of databases. Homogeneity attacks can be performed if all the records belonging to a *qi*-group have the same value of sensitive information. In this case it is clear that the adversary may easily violate the users' privacy despite anonymity is formally enforced. The same problem may arise as well in context-aware services in the case an adversary recognizes that all the users in an anonymity set actually issued a request with the same value of private information. To our knowledge, a first effort to defend against such attacks in context-aware systems has been presented in [45]. That proposal aims at protecting from multiple-issuers historical attacks by applying a bounded generalization of both context data and service parameters.

6 Towards a comprehensive framework for privacy protection in context-aware systems

Based on the weaknesses emerged from the analysis of the proposed techniques, in this section we advocate the use of a combined approach to address the comprehensive issue of privacy in context awareness; we present existing proposals, and we illustrate the logical design of a framework intended to solve most of the identified problems.

On the need for a combined approach The analysis of the state-of-the-art reported in the previous sections has shown that each of the proposed approaches, even if effective in a particular scenario and under particular assumptions, fails in providing a solution to the general problem. In particular:

- cryptographic techniques for private information retrieval presented up to the time of writing are unfeasible to complex context-aware services, due to problems of bandwidth and computational resources consumption;
- protecting communication privacy between the context source and the context data consumer (e.g., the service provider) is useless in the case the context data consumer is untrusted;
- access control techniques (possibly coupled with obfuscation) are ineffective in the case the access to a service is a sensitive information by itself, since they do not protect from the disclosure of the user's identity. Moreover, they do not prevent a malicious subject to adopt reasoning techniques in order to derive new sensitive information based on data it is authorized to access;
- techniques for identity anonymity rely on the exact knowledge about the external information available to an adversary. However, especially in pervasive and mobile computing scenarios, such knowledge is very hard to obtain, and adopting worst-case assumptions about the external information leads to a significant degradation of the quality of released context data.

These observations claim for the combination of different approaches in order to protect against the different kind of attacks that can be posed to the privacy of users taking advantage of context-aware services.

Proposed techniques Proposals to combine different approaches in a common framework have been recently presented.

In [46], an architecture for privacy-conscious context aggregation and reasoning is illustrated. The proposed solution adopts client-side reasoning modules to abstract raw context data into significant descriptions of the user's situation (e.g., current activity and stereotype) that can be useful for adaptation. Release of private context information is controlled by context-aware access control policies, and the access to context information by service providers is mediated by a trusted intermediary infrastructure in charge of enforcing anonymity. Moreover, cryptographic techniques are used to protect communications inside the user trusted domain.

Papadopoulou et al. present in [47] a practical solution to enforce anonymity. In that work, no assumptions about the external knowledge available to an adversary are made; hence, the proposed technique does not formally guarantee a given anonymity level. For this reason, the anonymization technique is coupled with access control and obfuscation mechanisms in order to protect privacy in the case an adversary is able to discover the user's identity. That technique is applied using the *virtual identity* metaphor. A virtual identity is essentially the subset of context data that a user is willing to share with a third party in a given situation; in addition, since anonymity is not formally guaranteed, part

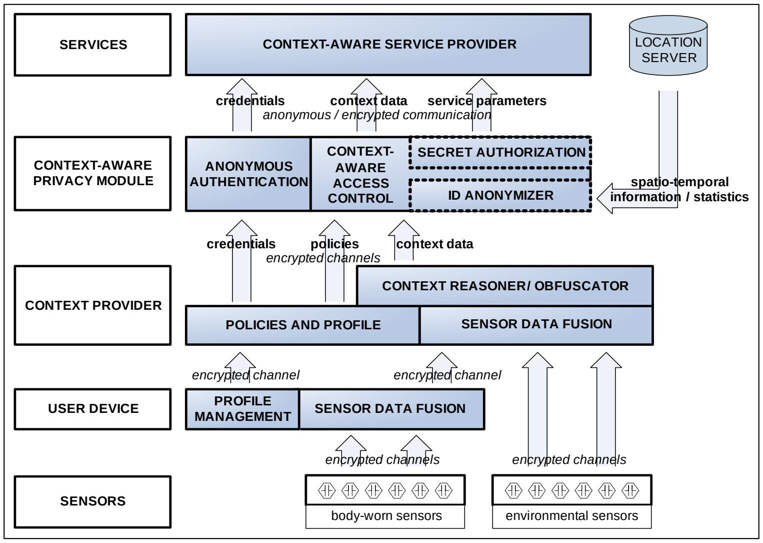

Fig. 2. The envisioned framework

of the shared context data can be obfuscated on the basis of privacy policies in order to hide some sensible details. For instance, a person could decide to share her preferences regarding shopping items and leisure activities, as well as her obfuscated location, when she is on vacation (using a *tourist* virtual identity), while hiding those information when she is traveling for work (using a *worker* virtual identity). With respect to the problem introduced by multiple requests issued by the same user, specific techniques are presented to avoid that different virtual identities can be linked to the same (anonymous) user by an adversary.

While the above mentioned works try to protect the privacy of users accessing a remote service, the *AnonySense* system [48] is aimed at supporting privacy in opportunistic sensing applications, i.e., applications that leverage opportunistic networks formed by mobile devices to acquire aggregated context data in a particular region. To reach this goal, the geographic area is logically partitioned into tiles large enough to probabilistically gain k-anonymity; i.e., regions visited with high probability by more than k persons during a given time granule. Measurements of context data are reported by mobile nodes specifying the tile they refer to and the time interval during which they were acquired. Moreover, in order to provide a second layer of privacy protection, obfuscation is applied on the sensed data by fusing the values provided by at least l nodes ($l \leq k$) before communicating the aggregated data to the application. Cryptographic techniques are used to enforce anonymous authentication by users of the system.

Towards a comprehensive framework We now illustrate how existing techniques can be extended and combined in a logical multilayer framework, which is graphically depicted in Figure 2. This framework is partially derived from the preliminary architecture described in [46]. However, the model presented here is intended to provide a more comprehensive privacy solution, addressing problems regarding sensor and profile data aggregation and reasoning (including obfuscation), context-aware access control and secret authorization, anonymous authentication, identity anonymity, and anonymous/encrypted communication. Clearly, the actual techniques to be applied for protecting privacy depend on the current context (users' situation, available services, network and environmental conditions). However, we believe that this framework is flexible enough to provide effective privacy protection in most pervasive and mobile computing scenarios. The framework is composed of the following layers:

- ***Sensors*** **layer:** This layer includes body-worn and environmental sensors that communicate context data to the upper layers through encrypted channels using energy-efficient cryptographic protocols (e.g., those based on elliptic curves [49] like in Sun SPOT sensors [50]). We assume that this layer is within the trusted domain of the user (i.e., sensors do not deliberately provide false information).
- ***User device*** **layer:** This layer is in charge of managing the user's profile information (i.e., context data that are almost static, like personal information, interests and preferences) and privacy policies. Upon update of this information by the user, the new information is communicated to the upper layer. Moreover, this layer is in charge of fusing context data provided by body-worn sensors and to communicate them in an aggregated form to the upper layer on a *per-request* basis (e.g., when those data are required by a service for performing adaptation). This layer is deployed on the user's device, which is assumed to be trusted (traditional security issues are not addressed here); communications with the upper layer are performed through encrypted channels.
- ***Context provider*** **layer:** This layer is in charge of fusing sensor data provided by the lower layers, including those provided by sensors that are not directly under the communication range of the user device. Moreover, according to the user's policies, it performs context reasoning and obfuscation for privacy and adaptation purposes, as described in [46]. It communicates user's credentials, privacy policies, and context data to the upper layer on a *per-request* basis through encrypted channels. This layer belongs to the user's trusted domain; depending on the device capabilities, it can be deployed on the user's device itself, or on another trusted machine.
- ***Context-aware privacy module*** **layer:** This layer is in charge of anonymously authenticating the user on the upper layer, and to enforce her context-aware access control policies, possibly after a phase of secret negotiation with the third party. Moreover, depending on the user's policies, it can possibly anonymize the user's identity on the basis of (either precise or statistical) trusted information received from the upper layer (e.g., spatio-temporal information about users received from a trusted location server). Protocols for anonymous/encrypted communication are adopted to provide credentials,

context data and service parameters to the upper layer. This layer belongs to the user's trusted domain. Depending on device capabilities and on characteristics of the actual algorithms it adopts (e.g., to enforce anonymity), this layer can be implemented on the user's device, on another trusted machine, or on the infrastructure of a trusted entity (e.g., the network operator).

- ***Services*** **layer:** This layer is composed of context-aware service providers and other infrastructural services (e.g., location servers). Typically, this layer is assumed not to belong to the user's trusted domain, even if particular services can be trusted by the user (e.g., a network operator location server).

7 Conclusions

Through a classification into four main categories of techniques, we have described the state of the art of privacy preservation for georeferenced context-aware services. While previous work has also proposed the combination of techniques from two or more categories, we claim that a deeper integration is needed and we propose an architecture for a comprehensive framework towards this goal. Clearly, there is still a long way to go in order to refine the architecture, work out the details of its components, implement and integrate the actual techniques, and test the framework on real applications. Moreover, there are still several other aspects, not considered in our paper, that deserve investigation. For example, since there are well-known techniques for context reasoning, they may have to be taken into account, since released context data may determine the disclosure of other context data, possibly leading to privacy leaks that were previously unidentified. Furthermore, computationally expensive techniques (e.g., those making use of ontological reasoning or complex cryptographic algorithms) pose serious scalability issues that may limit their applicability in real-world scenarios. Finally, since the access to context data of real users is generally unavailable for privacy reasons, sophisticated simulation environments are needed to evaluate the actual effectiveness of privacy preservation mechanisms in realistic situations.

Acknowledgments

This work was partially supported by National Science Foundation (NSF) under grants N. CNS-0716567 and N. IIS-0430402, and by Italian MIUR under grant InterLink II04C0EC1D.

References

1. Palen, L., Dourish, P.: Unpacking "privacy" for a networked world. In: Proceedings of the 2003 Conference on Human Factors in Computing Systems (CHI 2003), ACM (2003) 129–136
2. Lederer, S., Hong, J.I., Dey, A.K., Landay, J.A.: Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing **8**(6) (2004) 440–454

3. Bettini, C., Mascetti, S., Wang, X.S.: Privacy Protection through Anonymity in Location-based Services. Handbook of Database Security: Applications and Trends (2008) 509–530
4. Atallah, M.J., Frikken, K.B.: Privacy-Preserving Location-Dependent Query Processing. In: ICPS '04: Proceedings of the The IEEE/ACS International Conference on Pervasive Services, IEEE Computer Society (2004) 9–17
5. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), ACM (2008) 121–132
6. Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: Location Privacy Protection Through Obfuscation-Based Techniques. In: Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec'07). Volume 4602 of Lecture Notes in Computer Science., Springer (2007) 47–60
7. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In: Proceedings of the 24th International Conference on Data Engineering (ICDE 2008), IEEE Computer Society (2008) 366–375
8. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proc. of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys), USENIX Association (2003) 31–42
9. Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing **7**(1) (2008) 1–18
10. Aggarwal, C.C.: On k-Anonymity and the Curse of Dimensionality. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB), ACM (2005) 901–909
11. Neuman, B., Ts'o, T.: Kerberos: an authentication service for computer networks. Communications Magazine, IEEE **32**(9) (Sep 1994) 33–38
12. Freedman, M.J., Morris, R.: Tarzan: a peer-to-peer anonymizing network layer. In: CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, ACM (2002) 193–206
13. Reiter, M.K., Rubin, A.D.: Anonymous web transactions with crowds. Commun. ACM **42**(2) (1999) 32–48
14. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium, USENIX Association (2004) 21–21
15. Goldschlag, D., Reed, M., Syverson, P.: Onion routing. Commun. ACM **42**(2) (1999) 39–41
16. Al-Muhtadi, J., Campbell, R., Kapadia, A., Mickunas, M.D., Yi, S.: Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02), IEEE Computer Society (2002) 74
17. Atallah, M.J., Du, W.: Secure multi-party computational geometry. In: WADS '01: Proceedings of the 7th International Workshop on Algorithms and Data Structures, Springer-Verlag (2001) 165–179
18. Samarati, P., De Capitani di Vimercati, S.: Access Control: Policies, Models, and Mechanisms. In: Foundations of Security Analysis and Design, Tutorial Lectures. Volume 2171 of Lecture Notes in Computer Science., Springer (2001) 137–196

19. Kumar, A., Karnik, N.M., Chafle, G.: Context sensitivity in role-based access control. Operating Systems Review **36**(3) (2002) 53–66
20. Covington, M.J., Fogla, P., Zhan, Z., Ahamad, M.: A Context-Aware Security Architecture for Emerging Applications. In: Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002), IEEE Computer Society (2002) 249–260
21. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: Proteus: A Semantic Context-Aware Adaptive Policy Model. In: Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks(POLICY 2007), IEEE Computer Society (2007) 129–140
22. Sandhu, R., Samarati, P.: Access Control: Principles and Practice. IEEE Communications **32**(9) (1994) 40–48
23. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE Computer **29**(2) (1996) 38–47
24. Hengartner, U., Steenkiste, P.: Avoiding Privacy Violations Caused by Context-Sensitive Services. Pervasive and Mobile Computing **2**(3) (2006) 427–452
25. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)
26. Wang, C.D., Feng, L.C., Wang, Q.: Zero-Knowledge-Based User Authentication Technique in Context-aware System. Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on (April 2007) 874–879
27. Hengartner, U., Steenkiste, P.: Access control to people location information. ACM Trans. Inf. Syst. Secur. **8**(4) (2005) 424–456
28. Hull, R., Kumar, B., Lieuwen, D., Patel-Schneider, P., Sahuguet, A., Varadarajan, S., Vyas, A.: Enabling Context-Aware and Privacy-Conscious User Data Sharing. In: Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04), IEEE Computer Society (2004) 187–198
29. Atluri, V., Shin, H.: Efficient Security Policy Enforcement in a Location Based Service Environment. In: Proceedings of Data and Applications Security XXI, 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Volume 4602 of Lecture Notes in Computer Science., Springer (2007) 61–76
30. Atluri, V., Chun, S.A.: A geotemporal role-based authorisation system. International Journal of Information and Computer Security **1**(1–2) (2007) 143–168
31. Corradi, A., Montanari, R., Tibaldi, D.: Context-Based Access Control Management in Ubiquitous Environments. In: Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA 2004), IEEE Computer Society (2004) 253–260
32. Sacramento, V., Endler, M., Nascimento, F.N.: A Privacy Service for Context-aware Mobile Computing. In: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05), IEEE Computer Society (2005) 182–193
33. Zhang, Q., Qi, Y., Zhao, J., Hou, D., Zhao, T., Liu, L.: A Study on Context-aware Privacy Protection for Personal Information. In: Proceedings of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN 2007), IEEE Computer Society (2007) 1351–1358
34. Bakken, D.E., Parameswaran, R., Blough, D.M., Franz, A.A., Palmer, T.J.: Data Obfuscation: Anonymity and Desensitization of Usable Data Sets. IEEE Security & Privacy **2**(6) (2004) 34–41
35. Xiao, X., Tao, Y.: Personalized privacy preservation. In: SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, ACM Press (2006) 229–240

36. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE 2005). Volume 3468 of Lecture Notes in Computer Science., Springer (2005) 152–170
37. Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. J. Web Sem. **1**(3) (2004) 241–260
38. Wishart, R., Henricksen, K., Indulska, J.: Context Privacy and Obfuscation Supported by Dynamic Context Source Discovery and Processing in a Context Management System. In: Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing (UIC 2007). Volume 4611 of Lecture Notes in Computer Science., Springer (2007) 929–940
39. Sheikh, K., Wegdam, M., van Sinderen, M.: Quality-of-Context and its use for Protecting Privacy in Context Aware Systems. Journal of Software **3**(3) (2008) 83–93
40. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Trans. on Knowledge and Data Engineering **13**(6) (2001) 1010–1027
41. Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability. Volume 2009 of LNCS., Springer (July 2000) 1–9
42. Shin, H., Atluri, V., Vaidya, J.: A Profile Anonymization Model for Privacy in a Personalized Location Based Service Environment. Proceedings of the 9th International Conference on Mobile Data Management (MDM'08) (2008) 73–80
43. Riboni, D., Pareschi, L., Bettini, C.: Shadow Attacks to Users' Anonymity in Pervasive Computing Environments. Journal of Pervasive and Mobile Computing (To appear) DOI: 10.1016/j.pmcj.2008.04.008.
44. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: Privacy Beyond k-Anonymity. In: Proceedings of ICDE 2006, IEEE Computer Society (2006)
45. Riboni, D., Pareschi, L., Bettini, C., Jajodia, S.: Preserving Privacy in LBS against Attacks based on Concurrent Requests. Technical Report TR 24-07, University of Milan (2007)
46. Pareschi, L., Riboni, D., Agostini, A., Bettini, C.: Composition and Generalization of Context Data for Privacy Preservation. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), Proceedings of the Workshops, IEEE Computer Society (2008) 429–433
47. Papadopoulou, E., McBurney, S., Taylor, N., Williams, M.H., Dolinar, K., Neubauer, M.: Using User Preferences to Enhance Privacy in Pervasive Systems. In: Proceedings of the Third International Conference on Systems (ICONS 2008), IEEE Computer Society (2008) 271–276
48. Kapadia, A., Triandopoulos, N., Cornelius, C., Peebles, D., Kotz, D.: AnonySense: Opportunistic and Privacy-Preserving Context Collection. In: Proceedings of the 6th International Conference on Pervasive Computing (Pervasive 2008). Volume 5013 of Lecture Notes in Computer Science., Springer (2008) 280–297
49. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Proceedings of Advances in Cryptology (CRYPTO '85). Volume 218 of Lecture Notes in Computer Science., Springer (1986) 417–426
50. Simon, D., Cifuentes, C., Cleal, D., Daniels, J., White, D.: Java$^{\mathrm{TM}}$ on the bare metal of wireless sensor devices: the squawk Java virtual machine. In: Proceedings of the 2nd International Conference on Virtual Execution Environments (VEE 2006), ACM (2006) 78–88