Context-Dependent Access Control for Contextual Information

Christin Groba, Stephan Groß and Thomas Springer
Technische Universität Dresden
Department of Computer Science
Institute for System Architecture
D-01062 Dresden, Germany
Email: {christin.groba, stephan.gross, thomas.springer}@tu-dresden.de

***Abstract*— Following Mark Weiser's vision of ubiquitous computing and calm technology, computer systems should run in the background, preferably without the user noticing it at all. The gathering and disclosure of contextual information on the one hand enables the improvement of system behaviour towards a more autonomous and adaptive behaviour but on the other hand raises privacy issues by disclosing personal data. Thus, a major challenge in ubiquitous computing environments is achieving a good balance between convenience and control over personal data. In this paper we describe an access control mechanism for context data that enables the user to control his personal data in a convenient and non-intrusive way. The approach is based on existing role-based access control mechanisms but extends them as follows. Firstly, our approach is owner-centric, i.e. it is under control of each user, to whom his context is propagated throughout the system. Secondly, our approach does not only control the access to context data but also utilizes context information to simplify the management of these control mechanisms to make the handling of access control more convenient to the user. And thirdly, it introduces individual roles for each user and thus replaces the centrally defined role model of common role-based access control by distinct models for each user. We have validated our approach based on an extended instant messaging system called Adaptive Multimedia Messenger, providing varying buddy information dependent on the access permission of the requesting user.**

I. Motivation

Context-Awareness is a key technology for enabling applications to be sensitive of the social and physical setting in which they operate, so that they can adapt their behaviour accordingly, tailoring their services to best fit the current situation of their users and thus improve the overall usability. However, by gathering and sharing context information in distributed systems several privacy issues arise as context data always involves personal data. Thus, to meet both user demands of convenient use as well as privacy we have to implement sound security mechanisms especially for access control in context-enhanced computing environments.

In this paper we describe an access control mechanism for context data. Our approach can be characterized by three basic properties: First, it is owner-centric, that means it is up to each user if and how his context data is propagated throughout the system enabling him to enforce his right for informational self-determination. Secondly, our approach not only controls the access to context data but also utilizes context to simplify the management of these control mechanisms. And thirdly, it improves approved role-based access control by introducing individual roles for each user and thus replacing the centrally defined role model by distinct models for each user.

The remainder of our paper is organized as follows: Before we describe our approach of an owner-centric context-dependent role based access control system for context data in more detail (section II) we further elaborate on the addressed problem by discussing a typical use case scenario (section I-A) and its security implications (section I-B). We also analyse the drawbacks of traditional role-based access control (RBAC) for our purpose and argument how it can be enhanced to satisfy our requirements (sections I-C and I-D). Section III describes the prototype implementation to validate our approach and first evaluation results. In section IV we discuss related work and finally come up with a conclusion and an outlook for future work in section V.

A. A typical use case scenario

Alice enjoys the upcoming possibilities of using context-sensitive applications. Location-based services, electronic reminders as well as intelligent housing seem to make life easier and let her concentrate on things that really matter. A distributed context service captures Alice's personal information (e.g. available devices, connectivity, activity, on-line status and location) and provides it to new-quality, context-aware applications.

Besides incorporating context into her personal applications, Alice also likes to share it with her colleagues and clients via an AMM buddy list. The Adaptive Multimedia Messenger (AMM) [1] is an instant communication system providing multi-point conferencing based on a mobile communication infrastructure seamlessly integrating heterogeneous access networks and devices. By using the AMM buddy list clients and co-workers get an understanding of Alice's current situation and may act accordingly. Although Alice has basically agreed upon capturing and sharing her contextual data, access permission should depend on her own current environment as well as on the user who requests the information.

Example 1: Working as an architect, earlier as usual Alice arrives in her office to finish an urgent project plan. Instead of going into her own office, Alice enters a special inspiration

Second International Conference on Availability, Reliability and Security (ARES'07)
0-7695-2775-2/07 $20.00 © 2007 IEEE

room for being undisturbed. As the official working hours set in, the stationary telephone in the room rings. Alice remembers that in the current situation only colleagues who are members of the project she works on have access to her location and thus available devices. All others would not even know that she is at work already.

Example 2: Having several appointments Alice heads from one building site to another, carrying a mobile phone and a WLAN-enabled laptop with her. Mr. Vu, one of Alice's clients, wants to talk over his restaurant plans. He consults his buddy list to find out, how to get in touch with Alice. Since he currently has only access to Alice's available devices, he does not see that she has just left his biggest competitor. Only shortly after, Mr. Vu's access rights change and he sees that Alice is located near his building site. An entry in her calendar states that she will be there for about 30 minutes. The chance for Mr. Vu to take a taxi and meet her there.

B. Some reflections about security

Looking at our just mentioned use case scenario we can make some important observations concerning its security aspects. First of all, we can distinguish several acting entities like our architect Alice, her colleagues, and her clients. All these entities have different interests and security concerns that have to be taken into account when designing such a system. For example, Alice is not willing to give full access to her personal information neither to her co-workers nor to her customers. This might also be in their interest, e.g. when dealing with competing customers as in example 2. Secondly, we have to consider the different systems such as the context service, the adaptive multimedia messenger, the actors' personal electronic devices like PDA, laptop or cell phone, a probably used sensor network infrastructure, and last but not least the utilized access network. All these systems are used to provide the user with an up-to-date buddy list showing him the location and accessibility of the persons he is interested in. In doing so, these systems have one thing in common: they are based on a distributed architecture, thus making it difficult to rely on centralized security measures. Furthermore, they have to incorporate the different actor's security needs. Having a closer look at the basic security protection goals we can draw several conclusions.

- The integrity and availability of the proposed service is a protection goal typically wanted by all participating actors. However, availability is a hard problem for mobile devices as they might vanish at any time. Thus, the underlying context service and AMM architecture has to compensate this effect, for instance by introducing redundancy using proxies. Proxies acting on behalf of a user must respect his (possibly changing) security wishes, e.g. who is allowed to access personal data like location or activity.
- Talking about confidentiality we have to distinguish between protecting message exchange from eavesdropping and the protection of a user's personal data. A basic method to protect a user's privacy is to suppress the generation of personal data. However, this contradicts the principle of a context service.

In the following we concentrate on the second aspect and propose a solution to the mentioned dilemma of enhancing a system's usability by context data on the one hand and protecting its user's right for informational self-determination on the other hand. We, therefore, propose an owner-centric dynamic role-based access control model for context data.[1]

C. Why RBAC is insufficient

Traditional RBAC [2] is designed to compensate the weakness of discretionary and mandatory access control concerning large, structured organizations of people. RBAC's role-concept meets the key requirements of such organizations:

- All forms of data belong to the organization's intellectual property rather than to its originator.
- A global security policy managing access rights applies to hundreds of users and resources.
- Due to the organization's dynamics access rights may be granted or revoked.

Defined independently from a certain user a role is a set of permissions for a job profile, responsibility, or field of activity. The main advantage of this access control model is its efficient management of users and their access rights. However, the above use case scenario differs from the characteristics of large organizations and brings up new requirements that can only be partially fullfilled by the traditional RBAC model.

First, opposing to an organization-wide security policy, the context-owner individually defines roles and assigns them to other users. Therefore, the global, centralized policy enforcement shifts towards individual, decentralized policies and role concepts.

Secondly, as context data gets involved, the access decision no longer depends on user credentials only. The context in which the access request is made such as time, place, and object content will additionally influence the access decision. Traditional RBAC neither supports time-based, content-based nor context-based access control. It lacks the capability of conditionally granting or denying access.

Thirdly, the volatile and dynamic nature of context data requires a fast and flexible access management. RBAC access management is partitioned in user-role assignment compensating an organization's dynamics and a rather static role-permission assignment. With the variability of context not just some users but entire roles may temporary loose or acquire access permission.

D. Why RBAC is still the model of choice

RBAC introduces an efficient access rights management. The role abstraction layer between a user and his access permissions allows fast and simple changes without analyzing the overall access structure. The context-owner is considered

[1] To prevent any confusion we call the user associated with context data the *context-owner*, distinguishing him from the user that actually makes a context access request.

to be his own security administrator defining individual roles. Studies like [3] prove that the mechanism of forming groups is even for non-experts an easy process. It provides sufficient flexibility without much configuration effort, because a multitude of users and permissions are mapped on just a few roles.

The traditional constraint concept of RBAC aims at preventing role conflicts and enforcing the principles of separation of duty and least privilege. Constraints will be used to define context constraints whose purpose is to set the terms under which a role is valid for a certain user, or a permission is valid for a certain role.

As the above discussion shows, a sufficient solution for the conflict context-awareness vs. privacy will need to extend traditional RBAC to an owner-centric, flexible, context-based access control model.

II. Concept of an owner-centric context-Dependent RBAC

A. Defining individual policies

The concept of defining individual polices originates from the idea that each user should be in control over access to both his personal and his contextual data. Therefore, the context-owner is considered his own security administrator designing an individual security policy which includes:

- definition of roles
- permission assignment to each role in form of context-dependent access rules
- role assignment to each potential context-user

To integrate context data with an access decision context-dependent access rules are created. They define the condition under which a permission is valid for a certain role.

access rule	is a quadruple of a role name, access mode, access object and context constraint. *access rule:=(role, mode, object, context constraint)*
context constraint	is a conjunction of one or more conditions that bind one or more statements disjunctively. *context constraint*:= $condition_1 \cup \ldots \cup condition_n$ $condition := statement_1 \cap \ldots \cap statement_n$
statement	consists of an context attribute, an operation, and a reference value. The operation is either an element within a set of mathematical comparisons $\{\leq, \geq, \neq, =\}$ or user-defined e.g. *in, within.* *statement:=(context attribute operation reference value)*

During evaluation the current attribute value is concatenated via the operation to the reference value, so that the statement itself can be evaluated as true or false. According to the so-called whitelisting approach all access to context data is denied by default. An access rule is an explicit access permission which is valid only if the role, object, and access mode matches the requested parameters and if the context constraint is evaluated as true.

B. Finding an access decision

Once the individual policy is defined, it needs to be properly stored and enforced each time an access request is made. When access to context data is requested three players are distinguished:

- the user requesting access to an object
- the context-owner associated with the requested object
- the context source acquiring and providing context data

The context source could be located near the context-owner, e.g. the cell phone detecting the current location. On the other hand it could be a server within an infrastructure, e.g. collecting data about the noise level of a room and inferring the activity. In fact, information concerning one context-owner is distributed over many context sources while at the same time one context source may provide data to several context-owners. The challenge is to enforce the security policy of each owner in such a distributed environment.

If it was the context source to process a context request, it would need to store the policy of each context-owner whose data it collects. This might exceed the source's capacity because additionally to providing context data security policies have to be held consistent and up-to-date. Furthermore, the context-owner is forced to publish his security policy. In fact, the owner may get timid while defining the policy because it is regarded sensitive information he does not want to disclose. A further challenge is to evaluate the access rules in order to make an access decision. This requires the knowledge of the owner's individual role concept and access to other context sources since one source may not possess all necessary context attributes. But if sources had access to all other sources a policy definition would be useless and control through the context-owner would not exist.

The solution is to have the context-owner store his security policy locally and let him for evaluation purposes access all his context data even though it is distributed. As figure 1 shows, an access request is either directly addressed to the context-owner or forwarded to him via a context source. This approach has several advantages:

- The policy always remains with the context-owner. It is always up-to-date, available and does not have to be published.
- The context-owner achieves a high degree of control over external access on his personal data.
- The complete evaluation of context rules is guaranteed because an owner has always access to his own data.
- All access requests will be answered by the context-owner. Logging the request and corresponding decision enables the context-owner to adjust his role and permission concept.

The initial motivation of context sharing is that an user is aware of a context-owner's current situation and to make him

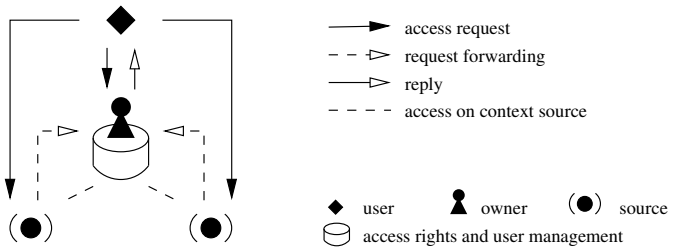

Fig. 1. Owner-centric approach: all context access requests are redirected to the context-owner. Only the context-owner has access to all his context data and therefore is able to completely evaluate all access rules.

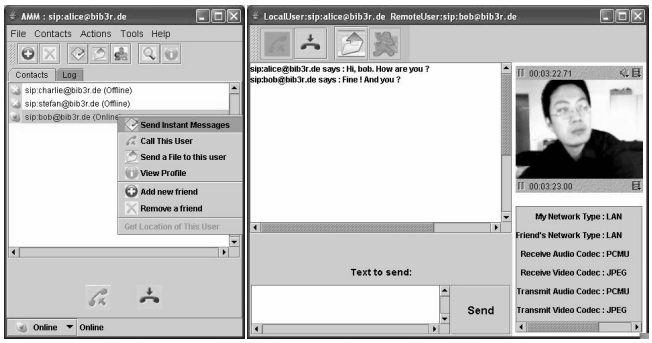

Fig. 2. Screenshot of the Adaptive Multimedia Messenger.

act accordingly. Applying the awareness principle to access control means to answer the question: How can a context-owner get aware of the user's context while making an access decision? Placing a counter-request ends up in a deadlock because both the context-owner as well as the initial user would get stuck evaluating context rules and waiting for the others reply. Hence, the user has to become an active part within the access decision process.

In an initialization phase the user sends a message to the context-owner whose reply will include all possible roles the context-owner assigned to him. On basis of this reply and according to his own current context the user may choose a role and formulate an access request with it. This solution is especially useful, if the user is assigned two exclusive roles and the context-owner would not know which one to pick. On the other hand the context-owner could allow the selection of more than one applying role but restricts a certain cardinality.

The problem that arises with the shift from central to local RBAC is the loss of a global understanding of roles. Individually defined roles may vary both in name and associated permissions. To make a profound selection, however, the user must have a certain understanding of what a role means. Publishing the set of permissions that are associated with a role is as sensitive as publishing the whole security policy and a context-owner might hesitate to do that. Therefore, a role description such as "A project member is an user working in the same project as the context-owner" needs to be exchanged in the initialization phase. For specific application domains certain role descriptions may be predefined at global scope and are only complemented by some individual refinements.

C. Enhancing availability

The downside of an owner-centric approach is the non-availability of information when the context-owner is not reachable. Because the context-owner is the only one to decide on an access request no information can be passed in his absence. There are two reasons, why a context-own might be out of reach: Firstly, he is not logged on to the system or entered it anonymously. Consequently, context data is not captured and thus not available. Secondly, a connection to the context-owner failed and neither a direct request nor its forwarding could be delivered. In practice, however, it is reasonable that some data is available despite the owner's absence. This is especially true for rather static information such as telephone numbers and email addresses. Therefore, the solution is a proxy appointed by the context-owner and authorized to pass on information. A certificate of authorization will increase the proxy's authenticity acting on behalf of the context-owner. Furthermore, the proxy has the owner's policy including the role concept and context rules and is obliged to enforce it. The context sources have to be configured the way that the context-owner as well as his proxy have full access to the context information.

III. Validation

We evaluated our concept of owner-centric context-dependent access control for context data by implementing a prototype application called Adaptive Multimedia Messenger presented in [1] and [4]. The prototype is based on a mobile communication infrastructure seamlessly integrating heterogeneous access networks (e.g. WLAN, GSM, UMTS, and Bluetooth) and devices (e.g. several types of PDAs and Laptops). On top of that infrastructure, the AMM provides multi-point conferencing adaptively using text chat, audio and video connections with different levels of quality. Because of this variety of communication options often the question arises if it is a good time to contact somebody and what communication mode is appropriate. Information like available devices, connectivity, activity and location of the potential contactee help to answer these questions. These information are made available based on an extended buddy list which not only provides the on-line state of buddies but also their current connectivity, device capabilities, location, supported codecs and communication modes. A screenshot of the AMM is depicted in figure 2. To provide access to context information we developed a distributed context service [1] which serves as basis for our prototype. Naturally, not all of the named information should be available to any buddy in the list. Thus, we have integrated our access control model into the context service.

A. Prototype implementation

The context-owner is represented by a personal terminal device. Responsible for access control it stores the individual

role concept, the user-role assignment as well as the role-permission assignment in form of access rules. We assume that each context-owner uses only a single mobile communication device at once. Otherwise synchronization would have been necessary which was out of scope of our work. An autonomous context service component installed on the device allows the context management including the configuration of local and remote sources. Through personalization the component distinguishes context access requests made by local applications benefiting the context-owner from requests remotely made by other users. An access request is answered according to the security policy stored on the device. For the design of individual access policies and their enforcement XACML [5] was used. The SecurityManager, as a part of the context service component, protects all access objects. It processes an access request by evaluating the request against applicable access rules and makes an access decision. Therefore, the SecurityManager covers the task of the Policy Enforcement Point and Policy Decision Point specified by XACML.

B. Discussion

The access concept was developed to support the context-owner's right for self-determination by allowing only him or his authorized proxy to decide on access requests concerning personal information. This concept is limited in that a context-owner has no influence on how a user will process the provided information as soon as he gets access to it. Therefore, the owner has to trust the user which is achieved by mutual authentication. As a pre-requisite for authorisation it is assumed that the communicating partners verify their identities in order to map them to individual roles. Authentication mechanisms can be integrated independently from the access control concept. Its validity is not affected.

For a more flexible access control traditional RBAC constraints are extended with context constraints. However, the choice of context data on which a access decision will depend on is limited. The concept works accurate as long as the context-owner integrates only his own context data to create access rules. If he uses context information he does not own, it is likely that rules cannot be completely evaluated due to an access deny or a request deadlock. However, this confinement is not very limiting. As studies [3], [6], [7] show, the user's role, the object content as well as the context-owner's situation are most important for an access decision.

The owner-centric approach can also be applied to non-personal data. For information e.g. concerning a certain room a security administrator will be considered the context-owner. In this case an access decision is independent of the owner's context, only the user's role, the content and context data associated with the room have an influence. To avoid evaluation failures or deadlocks the context constraints or administrator rights have to be appropriately selected.

IV. Related work

Access control models probably belong to the best analyzed areas of security research. First works [8], [9] were mostly motivated by military scenarios concerning confidentiality and integrity in computer systems. Later on, commercial applications introduced discretionary access control (DAC) policies letting the users have a fairly free choice how they want to protect their objects. Early publications only considered static access control, thus neglecting the mechanisms of changing access rules. The HRU model [10] was the first one to overcome this drawback by defining a simple language to express changes of access control policies.

However, by empowering the user to restrict the access to his objects one has to implement efficient procedures to manage these rights. In our opinion the most promising approach for this are role-based access control (RBAC) models. Based on [11], several approaches have been made to extend this concept by integrating context information. For securing context-aware applications [12] and [13] extend the traditional subject-centered role concept to *environment roles*. Activated by defined conditions they represent environment and system states influencing the access control decision. The context-sensitive RBAC model [14] composes so-called *context filter* from security-relevant information about the subject and target object to limit the applicability of permissions at runtime. Another context-related authorization and access control method based on RBAC is presented in [15] as part of a case study from the health care domain. The authors clearly distinguish between static role assignment to users and dynamic allocation of roles at session time and also describe how to integrate the RBAC functions in a trust infrastructure including smart cards. In [16] roles and permissions are modeled as role state and permission state machines. Current context is captured to initiate state transition events that dynamically change active roles and their permissions. The concept of context constraints proposed in [17] does not aim at RBAC core elements instead it re-evaluates a positive access decision by applying context conditions. The authors distinguish between authorization constraints and assignment constraints. While their concept is classified to be the former, the concept presented here belongs to the latter. A more general overview on existing access control models for collaborative systems discussing both the benefits as well as the weaknesses of existing models is given in [18].

All approaches mentioned above base on a central RBAC system where context data is assumed to be globally accessible and available. Target objects usually include files and devices. Our concept, however, aims at a local owner-centric access control for context data – an idea that also motivated the development of an experimental privacy-enhancing framework in [19]. Similarly a user is represented by a ContextManager which stores and protects personal data using RBAC. Our work differs in two ways. First, instead of using traditional RBAC, context information is integrated in the access decision process. Secondly, additionally to the context-owner's context and user role the user's context may actively influence the access decision.

For protecting personal user data, especially context information, several approaches have been made in the field of privacy.

[20] revives the idea of the "Platform for Privacy Preferences Project" (P3P). To decide if an user should use a context-aware application his preferences are compared with the application's declaration of intent. However, this declaration is no guarantee that it is actually enforced. Only after data misuse is detected, sanction can be put to action. [21], [22] discuss fundamental principles for privacy in ubiquitous computing environments. Their implementation in the privacy awareness system allows data collectors both to announce and implement data usage policies as well as to provide the possibility to keep track of personal information as it is stored, used, and removed from the system. Out of [22] privacy design principles, the authors of [23] view notice, choice and consent as the most useful control element for users. Therefore, a combination of user preferences for different situations is modelled as so-called *faces* to limit the collection of context data. A theoretical model to control privacy in context-aware systems is proposed by [24]. Access objects are members of defined *information spaces* limited by *boundaries*. On crossing boundaries defined actions such as alarms will be issued. Opposing a binary access decision (grant or denial) [25] creates policies to blur the detail of information depending on the requesting user. In fact, the user is always allowed to access data, however, its content is adjusted. To complement access control [26] calculates the level of exposure (LoE) when a legitimate, authorized user accesses data. If a high degree is determined, protecting mechanisms such as anonymisation, encryption or content blur should decrease the security risk.

V. Conclusions and future work

Enabling the user to control if and with whom he shares what kind of personal data is a key factor for the acceptance of context-aware systems and applications. But it places also an additional burden on the user because the definition of appropriate access rules can be a complex task. Therefore, we have introduced an owner-centric, context-dependent approach for role based access control. Our approach gives each user full control over his personal data on the one hand and utilizes context information on the other hand to ease the task of defining access rules.

Our approach is based on existing role-based access control mechanisms but extends them to reach a good balance between convenience and control over personal data. Based on the implementation of a distributed Context Service and the Adaptive Multimedia Messenger we have validated the feasibility of our approach.

In the future we plan to address the challenging issue of access control enforcement. Efficiency and scalability of an owner-centric access control will be of high interest especially in resource-limited, low-bandwidth mobile scenarios. Furthermore, a user study will be organized to validate the performance of our owner-centric approach and to examine its impact on the usability of context-aware systems.

References

[1] Thomas Springer and Kay Kadner and Frank Steuer and Ming Yin, "Middleware Support for Context-Awareness in 4G Environments," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 06)*. IEEE, 2006.

[2] D. Ferraiolo and D. Kuhn, "Role Based Access Control," in *15th National Computer Security Conference*, 1992.

[3] S. Patil and J. Lai, "Who gets to know what when: configuring privacy permissions in an awareness application," in *CHI*, G. C. van der Veer and C. Gale, Eds. ACM, 2005, pp. 101–110. [Online]. Available: http://doi.acm.org/10.1145/1054972.1054987

[4] Ming Yin and Joachim Fuchs and Thomas Springer and Frank Steuer, "An OSGi-based mobile service middleware for 4G adaptive multimedia services," in *ASWN 2006*, 2006.

[5] "OASIS eXtensible Access Control Markup Language (XACML)," http://oasis-open.org/committees/xacml, 2005.

[6] S. Consolvo, Smith, I, an E., T. Matthews, A. LaMarca, J. Tabert, and P. Powledge, "Location disclosure to social relations: why, when, & what people want to share," in *Proceedings of ACM CHI 2005 Conference on Human Factors in Computing Systems*, ser. Privacy 1, vol. 1, 2005, pp. 81–90. [Online]. Available: http://doi.acm.org/10.1145/1054972.1054985

[7] J. Olson, J. Grudin, and E. Horvitz, "Toward Understanding Preferences for Sharing and Privacy," Microsoft Research (MSR), Tech. Rep. MSR-TR-2004-138, Dec. 2004. [Online]. Available: ftp://ftp.research.microsoft.com/pub/tr/TR-2004-138.pdf

[8] D. Bell and L. LaPadula, "Secure computer systems: Mathematical foundations and model," MITRE Corporation, Bedford, MA, USA, Technical Report M74-244, 1973.

[9] K. Biba, "Integrity considerations for secure computer systems," Electronic Systems Div., Air Force, Hanscom AFB, MA, USA, Technical Report ESD-TR-76-372, 1977.

[10] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in Operating Systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, Aug 1976.

[11] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, Feb 1996.

[12] M. Covington, W. Long, S. Srinivasan, A. Dev, M. Ahamad, and G. Abowd, "Securing context-aware applications using environment roles," in *Proceedings of the sixth ACM symposium on Access control models and technologies*. Chantilly, Virginia, United States: ACM, May 2001, pp. 10–20.

[13] M. Moyer and M. Ahamad, "Generalized Role-Based Access Control," in *Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS-01)*. Los Alamitos, CA: IEEE Computer Society, Apr 2001, pp. 391–398.

[14] A. Kumar, N. Karnik, and G. Chafle, "Context sensitivity in role-based access control," *ACM SIGOPS Operating System Review*, vol. 36, no. 3, pp. 53–66, Jul 2002.

[15] M. Wilikens, S. Feriti, A. Sanna, and M. Masera, "A context-related authorization and access control method based on RBAC: A case study from the health care domain," in *Proceedings of the seventh ACM symposium on Access control models and technologies*, Monterey, California, USA, 2002, pp. 117–124.

[16] G. Zhang and M. Parashar, "Context-aware Dynamic Access Control for Pervasive Applications," San Diego, California, USA, Jan 2004. [Online]. Available: http://www.caip.rutgers.edu/TASSL/Papers/automate-sesame-cnds-04.pdf

[17] G. Neumann and M. Strembeck, "An approach to engineer and enforce context constraints in an RBAC environment," in *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT-03)*. New York: ACM Press, Jun 2003, pp. 65–79.

[18] W. Tolone, G.-J. Ahn, T. Pai, and S.-P. Hong, "Access control in collaborative systems," *ACM Computing Surveys (CSUR)*, vol. 37, no. 1, pp. 29–41, Mar 2005.

[19] P. Osbakk and N. Ryan, "A Privacy Enhancing Infrastructure for Context-Awareness," UK-UbiNet webpage, p. 2, September 2003, position Paper for the 1st UK-UbiNet Workshop, Imperial College, London, UK. [Online]. Available: http://www.cs.kent.ac.uk/pubs/2003/1769

[20] M. Y. Dan Hong, Vincent Y Shen, "Dynamic Privacy Management: a Plug-in Service for the Middleware in Pervasive Computing," in *MobileHCI 05*, 2002.

[21] M. Langheinrich, "A privacy awareness system for ubiquitous computing environments," in *UbiComp '02: Proceedings of the 4th international*

conference on Ubiquitous Computing, ser. Lecture Notes in Computer Science, vol. 2498. London, UK: Springer-Verlag, 2002, pp. 237–245.

[22] ——, "Privacy by design – principles of privacy-aware ubiquitous systems," in *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, ser. Lecture Notes in Computer Science, vol. 2201. London, UK: Springer-Verlag, 2001, pp. 273–291.

[23] J. M. Scott Lederer, Anind K. Dey, "Everyday privacy in ubiquitous computing environments," in *Ubicomp 2002 Workshop on Socially-informed Design of Privacy*, 2002.

[24] X. Jiang and J. A. Landay, "Modeling privacy control in context-aware systems," *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 59–63, 2002.

[25] Snekkenes, "Concepts for Personal Location Privacy Policies," in *CECOMM: ACM Conference on Electronic Commerce*, 2001.

[26] B. Dragovic and J. Crowcroft, "Information exposure control through data manipulation for ubiquitous computing," in *NSPW '04: Proceedings of the 2004 workshop on New security paradigms*. New York, NY, USA: ACM Press, 2005, pp. 57–64.