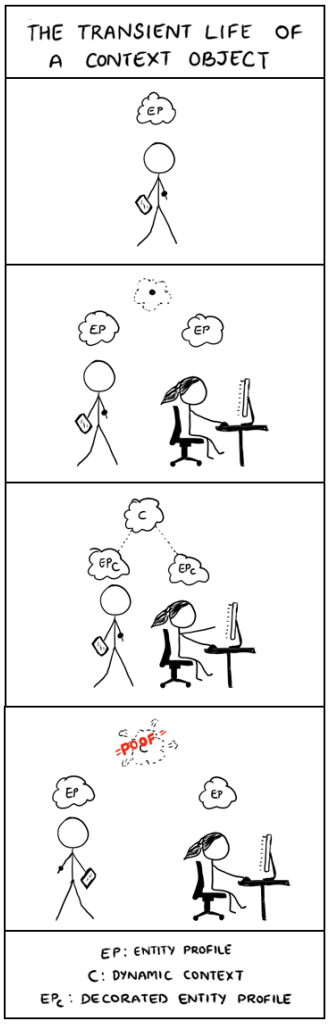

Figure 1. The concept of a transient and dynamic context object.

Secure Contexts: Context-Dependent, Dynamic, and Adaptive Security and Privacy

Raghav V. Sampangi
Faculty of Computer Science
Dalhousie University
PO Box 15000,
Halifax, NS B3H 4R2, Canada
raghav@cs.dal.ca

Kirstie Hawkey
Faculty of Computer Science
Dalhousie University
PO Box 15000,
Halifax, NS B3H 4R2, Canada
hawkey@cs.dal.ca

Abstract
All our interactions have some contextual information associated with them. Be it meeting with a co-worker in the hallway to discuss a project, or a team meeting in a meeting room, or a doctor and her staff meeting a patient, contexts are defined by attributes of all entities involved in the interaction. We consider the dynamic nature of contexts and propose a dynamic object model of context, and consider its application in access control. We envision that this model can be generalized to address context-dependent security and privacy needs in several application scenarios.

Author Keywords
Context-dependent security and privacy; access control; authentication; adaptive security and privacy; context-dependent access control.

ACM Classification Keywords
H.1.2 User/Machine Systems: Human factors; D.4.6 Security and Protection: Access controls; H.5.2 User Interfaces: User-centered design.

Motivation
User access to resources in an organization is primarily dependent on the user's role, and has been controlled by Role-based access control (RBAC) systems. Context information augments RBAC by making resource access

Access Control Use Case of our Context-Object Model

Figure 2. Illustration of a dynamic context with the doctor, nurse and patient (and their associated devices) as entities.

We discuss applying our model in context-dependent dynamic and adaptive access control, in a healthcare scenario. We assume the following: a patient is assigned a doctor and may be some resident doctors. Let's consider a scenario in which the doctor and nurse are in a patient's room. Context object creation is initiated as they are in physical proximity to the patient. Note that there is a logical proximity relationship between the medical staff and the patient as well.

…cont'd

controls more relevant for a given context. Kulkarni et al. [7] consider a context model that is defined by the application, which controls user access to resources. Though contextual parameters associated with an application are relevant, it is also important to consider context information associated with other entities, especially users, in the system. On the other hand, Corradi et al. [4] considered a more dynamic context scenario. They considered that users' ability to access resources will be determined by policies decided based on physical presence of the user and physical/logical properties of other mobile clients in the user's location. They define user profiles to be composed of user properties and desired views, where views are representative of available resources and actions, and user properties are defined by user characteristics such as physical and logical contexts. Their work does not consider if a user is trustworthy, an aspect defined by Bhatti et al. [3] in devising policies. Trust is computed by the system using logged transactions and locations of these transactions. Trust is used in addition to context parameters such as time, environmental state and location to allow/deny access to resources for a user. While they define context as being composed of changing parameters, it is still not adequate to capture the overall dynamic nature of context, as it does not consider the big picture, if you will, of user interactions with the system.

But how does one infer context? Hulsebosch et al. [6] suggest using proximity beacons that enable services providers to control access depending on whether users are near trusted access points. On the other hand, Miettinen et al. [8] use location (determined using GPS or WiFi) as the context parameter, to determine "Contexts of Interest (CoI)". CoI help determine presence of a user device in a location, and allows them to infer a "social context" based on other devices that can be detected by the user's device. This does not take into account the possibility of a user having more than one device on them at any point in time.

We consider a dynamic and adaptive object model to define context, and use this definition in designing an adaptive access control system. The notion of entity profiles is central to our model. An entity is any actor in the system, including users, devices, services, etc. We use the definitions of the physical and logical elements of a user profile defined by Corradi et al. [4] as the foundation for our work. In their work, users can access resources based on policies that are decided depending on physical location and logical properties of other mobile clients in the same location. However, accounting for changing workplace habits, possibility of holding unplanned meetings, and ability for one or more persons to join a meeting remotely, we consider that every entity in the system, not just physical locations or resources being accessed, is able to drive the creation of a new context. We thus use dynamic and adaptive contexts, whose properties are dependent on users, user attributes and their access privileges, and are driven by the need to quickly adapt to changing attributes of the context itself.

While defining/using contexts and contextual information can be considered to be more of a system implementation problem, we cannot afford to ignore the policies that would need to be adapted for different contexts. An organization may have a set of broad policies applicable to most scenarios, however, ubiquitous presence of mobile devices makes it difficult to conceptualize most troublesome contexts. Policies must not only be machine readable, but be in a manner understandable to the users if the system needs to let them know of a policy that is preventing an action they intend to perform. One way to show relevant policies to users is to simply present a grid of policies in place, as shown in an Expandable Grid [9]. Our approach

Access Control Use Case of our Context-Object Model (Cont'd)

The context object decorates entity profiles of all persons in the room. In such a scenario, the doctor may delegate higher access privileges to the nurse (role delegation), and their updated privileges would continue to exist only as long as the decorated context exists or as long as permitted by the doctor. These higher access privileges would still be restricted and not represent the entirety of access privileges available to the doctor.

These privileges are maintained by system rules that enforce access policies. This ensures that the transient policies only continue to exist until the time or other context limit set by the doctor.

In a similar manner, this model can be used to manage access control in universities, software and hardware companies, and in public service organizations (e.g. public libraries, etc.).

considers the dynamic policy model used by Bauer et al. [1], where they enable users to access resources and delegate policies when needed, either proactively or reactively.

We also consider that a present day user may use more than one mobile device, all of which may not be with the user at all times. This implies that inferring devices currently active in a given context and updating policies to be relevant to said devices will be critical, and so will be any necessary updates if the user were to log in to another device when in a context. Furthermore, access and other policies in any situational context vary not only with location, but also with logical characteristics of entities and their devices [2][12].

Allowing a system to infer everything about the context and have no user input would be detrimental to the system, because varying user expertise makes it more complex for the system to understand what each user would expect or understand. Requiring users to acknowledge and confirm every inference by a system would again be detrimental, as it increases user frustration. We therefore envision a system that infers context, determines policies and rules for that context, and makes it possible for entities in charge of the context to make changes if necessary. Thus, the model we adopt follows these principles identified by Reeder [11]: *flexible and simple to use, with a user interface that does not expect any advanced knowledge from the user*. Our work aims to address the following research questions:

- *Ease of use and willingness to adopt*: How easy will the system be for people to use, given that context and associated policies vary with time and depending on other entities in the context?
- From what is being shown to users on the interface, is it clear to users as to what action needs to be performed, and what will be the consequences of their actions?
- *Trust issues*: Are users willing to trust a system with dynamic and context-dependent security? Will it make them feel that their data and/or their identities are safe? Will the users feel that the system is trustworthy if it were able to prove itself (mutual authentication)?
- Does this system pose infrastructure and maintenance expectations, in addition to what is already being used?
- Do administrators believe that such a system is trustworthy to protect the organization's data and resources, and keep their users safe?

Dynamic Context-Object Model

We define an object profile for each entity (user, resource, etc.), i.e. possessing certain attributes and having acceptable behaviours. We consider that a situational context exists whether two entities are co-located physically or happen to collaborate remotely. Changing attributes of a situational context would include temporal changes in logical properties or in activities and information needs of each activity, and any changes in security and privacy. Several contextual parameters may constantly change, which is analogous to updating an object dynamically with additional responsibilities or removing them to return the object to its prior state. We therefore use Decorator design pattern [5] as a foundation for our object model. Furthermore, if a context updates, changes must be communicated to the objects that are a part of the context. This is similar to objects subscribing to and unsubscribing from updates, which led us to use the concept of Observer design pattern [5] as another essential concept in our model.

Our Dynamic Context-Object Model is composed of six elements or context profiles, as summarized in Figure 3. Each entity in our system has an entity profile, a behaviour of which is to check for any additional entities in its physical

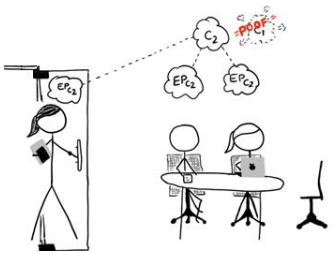

Figure 4. Conceptual illustration of a context being updated when a person joins an existing context. In such a scenario, the old context object is replaced by the new object, which decorates all entity profiles.

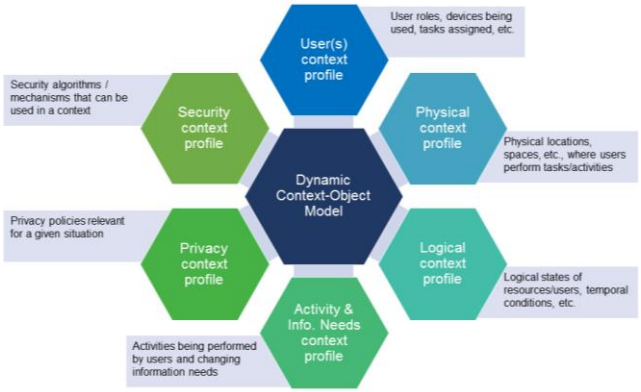

Figure 3. Elements of the proposed *Dynamic Context-Object Model*.

(near to one another in space and time) and logical (near to one another in organizational roles, team memberships, etc.) proximity. This ensures that a dynamic context object is created only when required. A dynamic context object may be created when—entities are co-located in a physical location (e.g. ad hoc meetings, meeting room, classroom, hospital ward, etc.) or entities are co-located in a virtual location (e.g. virtual meeting, video/audio conferencing, etc.), and the entities have logical proximity. When a new context is created, entity profiles are decorated with it and any resulting policies. The context object is updated when an entity joins or leaves the context. Each time the context is updated, all entity profiles are undecorated from the old context and decorated with the new context.

Context owner initiates the context update, by default. However, if the initiator joins the context at a later time (e.g. organizer joining the meeting late), the initiated context is updated and control is transferred to the logical owner when they join. This setup facilitates role delegation, i.e. if an initiator or owner were to leave a context, they may delegate some or all of their role privileges to any member of the context. Thus, although an entity may have higher access levels, it will be limited to that usage context and will be disabled when the context ceases to exist. Creation of a new context or subsequent update of a context allows the system to re-evaluate policies and rules, and associate those rules only with the relevant context object. Therefore, although organizational policies may exist, our model facilitates creation of transient policies, applicable to a specific context. In terms of system implementation, creation of dynamic objects and subsequent maintenance of their states allows for parallel/asynchronous processing, reducing load on the server, and thereby makes our solution scalable. A use case of our model is presented in the sidebar.

Access Policies and Conflict Resolution

When creating dynamic and adaptive context objects, roles of users and their access privileges may result in access policy conflicts. For our model to be usable, we need to determine ways to resolve such conflicts. We aim to evaluate conflict resolution algorithms and use one most relevant for our work. At this stage in development of our model, we draw encouragement from the words of Reeder et al. [10], who state that taking a user-centered approach towards security and considering user interface details when designing security models could help make systems usable.

Secure Contexts: The Way Ahead

We propose a novel object-oriented technique to model context, and discuss its application in access control. In the days to come, we will evaluate the feasibility of our model, and explore more use cases of its application, which we expect will make our model more comprehensive and make it generally applicable. We will explore applicable policy conflict resolution algorithms. We will also consider other security and privacy applications that might benefit from such a dynamic context model.

References

1. Lujo Bauer, Lorrie Faith Cranor, Robert W. Reeder, Michael K. Reiter, and Kami Vaniea. 2008. A user study of policy creation in a flexible access-control system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08). ACM, New York, NY, USA, 543-552.
2. Alireza Behrooz, and Alisa Devlic. A context-aware privacy policy language for controlling access to context information of mobile users. In *Security and Privacy in Mobile Information and Communication Systems*. Springer Berlin Heidelberg, 2011. 25-39.
3. Rafae Bhatti, Elisa Bertino and Arif Ghafoor. 2005. A Trust-based context-aware access control model for web-services. *Distributed and Parallel Databases*. 18, 1 (July 2005), 83-105. http://dx.doi.org/10.1007/ s10619-005-1075-7.
4. Antonio Corradi, Rebecca Montanari and Daniela Tibaldi, "Context-based access control management in ubiquitous environments," In *Proceedings of Third IEEE International Symposium on Network Computing and Applications* (NCA), 2004, pp. 253-260.
5. Erich Gamma et al. 1995. *Design Patterns.* Reading, MA: Addison-Wesley Publishing Co, Inc.
6. R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma. 2005. Context sensitive access control. In *Proceedings of the tenth ACM symposium on Access control models and technologies* (SACMAT '05). ACM, New York, NY, USA, 111-119.
7. Devdatta Kulkarni and Anand Tripathi. 2008. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 13th ACM symposium on Access control models and technologies* (SACMAT '08). ACM, New York, NY, USA, 113-122.
8. Markus Miettinen, Stephan Heuser, Wiebke Kronz, Ahmad-Reza Sadeghi, and N. Asokan. 2014. ConXsense: automated context classification for context-aware access control. In *Proceedings of the 9th ACM symposium on Information, computer and communications security* (ASIA CCS '14). ACM, New York, NY, USA, 293-304.
9. Robert W. Reeder, Patrick Gage Kelley, Aleecia M. McDonald, and Lorrie Faith Cranor. 2008. A user study of the expandable grid applied to P3P privacy policy visualization. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society* (WPES '08). ACM, New York, NY, USA, 45-54.
10. Robert W. Reeder, Lujo Bauer, Lorrie F. Cranor, Michael K. Reiter, and Kami Vaniea. 2011. More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 2065-2074.
11. Robert W. Reeder. 2011. Usable access control for all. In *Proceedings of the 16th ACM symposium on Access control models and technologies* (SACMAT '11). ACM, New York, NY, USA, 153-154.
12. Florian Schaub, Bastian Könings, and Michael Weber. 2015.Context-adaptive privacy: Leveraging context awareness to support privacy decision making. In *IEEE Pervasive Computing* 14.1 (2015): 34-43.