

# Context-Adaptive Information Security for UbiComp Environments

Boris Dragovic and Jon Crowcroft

University of Cambridge Computer Laboratory

{firstname.lastname}@cl.cam.ac.uk

## 1. Introduction and Motivation

Traditional computer applications expect a static execution environment. Such environments imply non- or slowly-evolving information security and privacy threat models. This assumption is no longer realistic in the Ubiquitous world scenario, where the environment around a piece of information, contained on a device or within a communications channel, is frequently changing. Thus, the information contained in the Ubiquitous world is exposed to varying threat models throughout its lifetime. Users expect high degree of information availability anytime and anywhere as needed, leading to serious security and privacy risks and access control problems.

The work which we are conducting, still in its infancy, tries to address the problem of protecting information security and privacy in the UbiComp world through matching (restrictiveness of) the format in which the information exists to a particular threat model, relative to the information sensitivity level, implicit in the surrounding environment - context. The process is continuous throughout the lifetime of the information's data representation in the UbiComp world.

We draw from observations of the human behavior in compromising situations. Just how many times have you caught yourself lowering the volume of your speech or switching to specially "crafted" vocabulary to prevent information leakage to third-parties. In effect, humans match the form and characteristics of information to the perceived security and privacy risks. Such human behavior proves to be very effective if the complexity of a situation is within our reasoning capabilities.

The aim of our work is to develop a novel security paradigm inspired by the observed human behavior. Probably the most obvious application is within user interface, e.g. steerable user interfaces, Personal Server [7], etc. scenarios. Essentially the same reasoning applies, although arguably in a more subtle way, for all data objects within the UbiComp world whether contained on a device or within a communications channel, e.g. data objects on a storage medium of a mobile device are under a higher degree of risk if not within a proximity of the owner, not all communications links are trusted in the same way, etc. We plan to address the issue in full.

To best illustrate the characteristics of the proposed approach we contrast it to the traditional notion of access con-

trol: we drift away from binary access decisions and create an continuum between granting and refusing access by performing fine-grained data format transformations; data is constantly tracked throughout its lifetime and both reactive and proactive policy specified actions are performed triggered by threat model changes; the model is aimed at minimizing the exposure of information to surrounding entities in the context while the data is being accessed by a legitimate, authenticated and authorized by other means, entity. The proposed approach can be seen as complementary and orthogonal to traditional notion of access control, including the novel context-aware variants such as Generalized Role-Based Access Control [5].

## 2. Model Overview

We define a *container* as immediate physical enclosure in which a data object exists, e.g. a device's display, storage device, working memory etc. *Containment* denotes the state of being within a container. *Levels of Exposure* (LoE) quantify the extent to which data is accessible to its surrounding. The meaning of each of LoE is uniform across the system and we expect it to be extrapolated from a wider system security policy. For example, extrapolating from traditional UNIX access control lists, we could define three system-wide Levels of Exposure for each data object: non-accessible, readable (or visible) and readable & writable (or visible and modifiable).

Physical properties of containers determine contextual attributes, and their states, that cause particular LoE to be active within the containment. The set of contextual attributes denoting a particular (*container, LoE*) is not unique and may vary as long as the overall confidence within the set is above a predefined threshold. For example, data displayed on an overhead screen is considered to be at a "visible" LoE if it is within a visibility field of a third party; the presence and proximity of a third party may be determined by a dedicated location infrastructure (e.g. Spirit [1]) or a set of individually less specific attributes, obtained through non-specialized interfaces e.g. environment audio analysis, personal device bluetooth proximity, history and diary information, etc., which together provide above-threshold confidence level of respective context state. It should be emphasized that LoE denote exposure of information to contextual entities strictly other than the legitimate data users. Logically, LoE make sense at the level below the standard authorization mechanisms. We say that two containers are

of a same type if the LoE they experience coincides for every possible contextual state.

Very few devices still contain specialized context-sensing capabilities. However many of the capabilities primarily intended for other purposes can be used for limited context analysis. For example, reachability of an office printer, may, with a certain probability, mean that the device is within a corporate headquarters; which in turn may imply certain characteristics of the secure perimeter; which in turn means that the device is less likely to be stolen; which, possibly with support of other means of context attribute inference, finally causes activation of lower level of LoE for data on local device's storage medium. So, LoE determination starts on individual devices, using their built in capabilities. This guarantees a degree of independence. Every device is certified to individually determine LoE up to a certain level for every of the containers it comprises. Further, devices may form trusted groups for threat model assessment, usually within a well known Personal Area Network e.g. mobile phone, PDA and a hands-free kit. Finally, the task may be off-loaded to trusted dedicated services e.g. Spirit [1] in well-instrumented environments.

To be able to perform the format transformations reactively to the LoE changes, the system needs to continuously track individual data-objects throughout their life-time in the UbiComp world. All data objects need to be tagged by a policy linking all applicable LoE to respective format transformation specification. Every container is also associated with a policy specifying, for each applicable LoE, set of container attributes and their required confidence values that set the LoE active. Depending on the level of application-awareness format transformations may range from bulk operations, such as e.g. encryption, deletion etc., to ideally data type specific, fine grained, e.g. anonymization, quality degradation, etc. The former also depends on the level of data-type awareness of the transformation-performing platform, highly dictated by the target device capabilities. Meaning of data transformations may vary among container types, for example, "deletion" may mean "screen-blanking" for a display while it may mean "permanent erasure" for a storage device. Format transformations may be performed by the local or a remote device. The latter occurs if the local device possesses insufficient capabilities to carry out the operation and is negotiated prior to data object being exported to the container.

Apart from the described reactive behavior, the approach provides for proactive multiplexing of data objects among containers of the same *class* (e.g. displays, permanent storage, communications channels etc.) to minimize both current and projected LoE for the (data object, container type, context) triple while maximizing data availability. Container-data multiplexing can also be a result of a reactive action in cases where it is of less *cost* than a required format transformation. The multiplexing approach, in addition to fine-grained data format transformations, maximizes the trade-off between data protection and availability in face of different threat models.

Representing an aid for fighting the complexities involved

in reasoning about information security and privacy in the UbiComp world, the approach assumes the notion of a *co-operating* user. Users are allowed to express their preferences with respect to data format and data-container multiplexing decisions while they are in line with the relevant policy. For example, a user should be allowed to opt to display data on a public display in a constrained format rather than on a mobile phone's display under a lower LoE.

In cases of frequent LoE fluctuations it sometimes makes sense to compromise data availability for resource preservation and perform a more restrictive format transformation instead of one or more less restrictive but more computationally complex ones. For example, simply blanking screen instead of information quality degradation or device multiplexing in case of a brief exposure level elevation caused by a person walking by the screen. These decisions are a task of the *predictor* module which may take into account high level data such as e.g. context attribute trends, history, diary information etc. in its decision making process.

### 3. Conclusion and Future Work

In this paper, we have proposed a novel, to the best of our knowledge, security paradigm which tries to maximize the trade-off between data availability and its security and privacy protection in the UbiComp world. We draw from the human behavior, which we believe is very effective within the reasoning abilities of humans, and hope to extend it into the UbiComp environments.

We intend to proceed by building a framework and a set of proof of concept applications targeted at the Personal Servers [7], Virtual Network Computing [6] and various user interfaces applications. We see Context Toolkit [4] as an initially suitable concept for rapid prototyping of context-aware applications and policy languages such as Ponder [3] and KeyNote [2] as venues we will explore for policy expressing. Evaluation will concentrate on performance in heterogeneous environments and overall usability.

### REFERENCES

- [1] N. Adly, P. Steggle, and A. Harter. Spirit: a resource database for mobile users. In *Proceedings of The ACM CHI'97 Workshop on Ubiquitous Computing*, 1997.
- [2] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. The keynote trust-management system, version 2. Request for comments, rfc 2704, 1999.
- [3] N. Damianou, N. Dulay, E. Lupu, and M. Sloman. The ponder policy specification language. In *Workshop on Policies for Distributed Systems and Networks*, 2001.
- [4] A. K. Dey, D. Salber, and G. D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction Journal*, 16(2-4):97-166, 2001.
- [5] M. Moyer and M. Ahamad. Generalized role-based access control. In *Proceedings of The 21st ICDCS*, 2001.
- [6] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1), January 1998.
- [7] R. Want, T. Pering, G. Danneels, M. Kummar, M. Sundar, and J. Light. The personal server - changing the way we think about ubiquitous computing. In *Proceedings of the UbiComp 2002*, LNCS, pages 194 - 209, October 2002.