# Usability of Security Management: Defining the Permissions of Guests

Matthew Johnson and Frank Stajano

University of Cambridge

**Abstract.** Within the scenario of a Smart Home, we discuss the issues involved in allowing limited interaction with the environment for unidentified principals, or guests. The challenges include identifying and authenticating guests on one hand and delegating authorization to them on the other. While the technical mechanisms for doing so in generic distributed systems have been around for decades, existing solutions are in general not applicable to the smart home because they are too complex to manage. We focus on providing both security and usability; we therefore seek simple and easy to understand approaches that can be used by a normal computer-illiterate home owner, not just by a trained system administrator. This position paper describes ongoing research and does not claim to have all the answers.

## 1 Introduction

### 1.1 The Smart Home

Of the many possible applications scenarios of ubiquitous computing, numerous visions of "Smart Home" environments have been put forward [7,2,10,8,3]. In a smart home, everyday objects such as appliances and furniture, as well as systems such as heating and ventilation, will feature embedded processors and communications and will work as both sensors and actuators in an integrated home system. On the communications side, low power and low bit rate wireless networks suitable for control of home systems are the focus of an industrial consortium[1] and an IEEE standard [6]. These models address multiple embedded devices and control of them from authenticated principals. Many of these models also propose solutions for preventing usage by strangers. What have not been adequately addressed so far are the issues of guests.

Unlike the people who normally live in the house, guests should not have access to everything; and, to the extent that regular users are modelled as having "accounts" on the home system, guests should not have to be given accounts; however, they are not strangers either. There are many common situations where it is desirable to give some level of access to a guest. For example, a guest might wish to play some music from their portable device on your Hi-Fi or a movie on your TV, and you might be happy to allow this; given the technical capabilities

[1] Zigbee: http://www.zigbee.org/

of a smart home with a wireless network, it would be unreasonable if this were not possible. But you shouldn't for that be forced to register the guest as if they were a new permanent occupant.

You would not want guests to be able to change your play-lists, central heating timers, intruder alarm codes and so on, although you might want them to be able to upload a song for you to listen to later.

An example of a commonly used delegation procedure in pre-smart-home situations is to leave the guest with a key until they leave and have them post it back through the door. When taking this as a model, you might want to be able to guarantee that the guest's access rights expire at the correct time and cannot be copied and used later.

One of the biggest challenges is not just creating a system that can do all this, but creating one which is easy to use. Systems which are designed to mimic actions in the real world (such as lending a key, or letting someone play a CD) need to match people's expectations and conceptions for them to be convenient and easy to use.

### 1.2 Location awareness

In the majority of context-aware systems, the primary source of contextual information is the relative (and sometimes also absolute) location of the people and objects involved in the interaction.

There are a great variety of ways of obtaining location information; Hightower and Borriello [5] as well as Beresford [1] provide useful surveys and taxonomies in the field.

One possible distinction is between systems, such as the Active Bat [4], that provide absolute positioning of objects within some local reference frame, and systems where active objects can just sense, perhaps through radio signal strength, whether they are within a given range of each other. In the first case, a central facility can trigger events based on the positions of the objects. In the second case, instead, objects have only a vague idea of their position and so even guaranteeing containment of a device within a room is difficult.

A smart home setup that will work in the latter case will also work in the former; we will concentrate on location systems without a strong notion of position so that our model will support both.

### 1.3 Security and Usability

The process of allowing a principal to perform an action on the system can be broken up into *identification* ("what is your name?"), *verification* ("prove that you are the one you claim to be") and *authorization* ("based on who you are, I'll let you do this"). The first two steps are usually taken together as *authentication.*

We certainly won't be alone in claiming that traditional methods of authentication of users to computers, such as passwords, are not suitable, from a usability perspective, for ubiquitous computing environments. Our focus is to make the three steps above more usable, without compromising security.

But more usable for whom? Certainly for the guests themselves; but also, and this is probably the more substantial challenge, for their hosts, who are the ones responsible for the complex task of defining the authorization rules, as well as the ones who stand to lose the most in case of security failures.

## 1.4 Identification: Defining Guests

To implement a system where guests can perform certain actions we need to define what a guest is and distinguish them from strangers who happen to be in range of the system.

There are various properties that guests have in the real world. Detecting such properties directly would allow people to use a guest system in an intuitive manner. Firstly, and most simply, a guest is inside the house, whereas a stranger is outside. With a position-based smart home this would be relatively easy to determine; however, only knowing proximity would give rise to too many annoying false negatives and dangerous false positives on the boundaries.

Guests can also be distinguished in that they have the permission of the owner (or more generally, any authorized principal) to be there.

## 1.5 Authorization: Defining Access Permissions

Smart Homes will, in the future, be used by people who are not normally security conscious and do not want to spend any effort configuring a system for guests. Therefore, even if we can determine who is a guest, the way in which we grant them permissions must be one that a non-technical user can easily understand and operate.

Ease of use is one of the major challenges in this work. With existing models it is possible to define a security policy that expresses the desired rules; however, it would not be easily understood by the common man. A corollary to this is that normal use should not be made more difficult with the addition of guest support. While guests are fairly common, they are the exceptional case and not the common one and should not make day to day running of a smart home more complicated for hosts.

The easy way out is to have experts pre-configure a set of canned profiles and just give the naïve home owner a menu of such profiles to choose from, when authorizing a guest to use facilities in the home. But this does not solve the hard problem of actually giving hosts fine control over the permissions they grant, while still retaining usability. Using the simple-minded profile-based system, a host may be forced to assign a guest to a totally inappropriate profile if that is the only way that a certain necessary permission (e.g. changing the thermostat temperature in the guest's bedroom) can be granted.

## 2 Research Ideas

### 2.1 Mental Models and Social Expectations

If the ideas we present are to be acceptable to the general public they must be easy to use. One of the ways this can be achieved is by harnessing existing expectations about how appliances work by making new systems behave similarly to current systems.

In the case of most household appliances the security policy is that of the 'Big Stick' principle [9, §4.2.8]. That is, whoever has physical access to the device is allowed to control it. This is sufficient security because there are social restrictions on people's actions. Guests are expected to behave in a certain fashion and there are social penalties which apply if they don't.

We can harness these social expectations when moving to smart environments and in a lot of cases the Big Stick principle, combined with social restrictions, is still sufficient.

We can, of course, use the technology to improve on the current situation. Firstly, we can restrict more of the actions. While social restrictions may be enough to control some actions we may well want to enforce some of the more sensitive actions via technology. We can also provide improved logging to make it a lot more obvious when a social rule is being breached.

### 2.2 Mimicking the Big Stick

If we are dealing with a system that uses a wireless connection then merely having access to the control interface does not guarantee physical possession. We may not be a guest, but actually an attacker in the street.

A great variety of cryptographic key setup protocols have been developed to ensure that a secure channel is established between exactly two devices and that we can keep this channel on subsequent connections between the two devices. The problem is then reduced to identifying and verifying the devices correctly when they first connect.

Combining this with our desire to mimic the existing methods of controlling a device leads to the solution of requiring a physical button press on the device the first time an action is requested. This proves the presence of someone in the room and therefore their status as a guest. This is not quite enough, though; we need to confirm that the action which is confirmed is the same one that they requested and not an attacker. The use of multi-factor authentication [11] can solve this: for example the guest may be shown a nonce on the TV screen and asked to text the nonce back to the smart home via his mobile phone to prove that he could read it and therefore that he is inside.

If we have an integrated smart environment then this identification as a guest can be propagated across several devices.

### 2.3 Granting Permissions

The second part of this research is about how to give permissions to guests in an easy to manage way. In the traditional security model you would define permissions for all actions to each guest in advance, akin to setting the read, write and execute permission bits to appropriate values on all the files of your computer before letting a guest use it. However, this is not feasible or desirable when you don't know about principals in advance, nor is it something that a non-trained user would be expected to do.

The first problem can be mitigated by giving the same permissions to all guests, but this is inflexible and still not something most users would be able to do. Another solution ("lazy evaluation" approach) could be to request approval for all actions by a guest from an administrator. This solves the problem of needing to do probably unnecessary work in advance, but gives a large penalty to actions done by guests—both for hosts, who are continuously interrupted with "can she do this?" requests, and for guests, who always have to wait until each action is individually authorized.

To mitigate these problems we suggest grouping actions in several ways. Firstly we have defined four types of action as related to guests:

1. Any guest may invoke without further authorization.
2. Authorization once for this action suffices for further invocations.
3. Each invocation requires individual authorization.
4. Guests may never invoke this.

The first category covers all functions for which the 'Big Stick' policy is still appropriate and we anticipate it will cover many of the functions available in household appliances. Category two contains functions which you might want to grant to some guests but not to others; however, once granted you are happy for those guests to access them as much as they like.

Using this grouping of functions allows appliance manufacturers to perform further grouping of security-equivalent functions. For example, on a smart Hi-Fi granting the play permission would always go hand in hand with the stop and pause permissions and so on. In the majority of cases, such functions would default to type 1.

For type 2 and 3 functions, invocation by a guest could cause a message to one of the known principals to authorize the request. This has several desirable properties. Firstly it allows the action to be authorized without any configuration in advance; and secondly it reinforces the social restriction on the action since the household member now knows what the guest is trying to do.

### 2.4 Ownership Delegation

In the situations described above we are giving permission to invoke functions whilst still retaining control of the device. There may be situations where we want to delegate control over a device, at least partially. For example, for guests staying for an extended period of time, we may wish to delegate the control of one room, including all the facilities and devices inside it, and the access control of that room or the house. We will ultimately want to retain full control over the room, but we might want to make some guarantees about how that control can be exercised so as to provide privacy to the guest.

## 3 Prototype System

We are building a prototype with the two purposes of trying out new ideas and then verifying whether they are as usable as we hope. The demo system shown at the workshop and described below was just a concept demo and therefore only explored the first point. As for the second point, we aim to develop a user study by deploying the system currently under development and letting non-technical users interact with it for some time. This aims to help us understand whether we are achieving an appropriate balance between security and usability.

### 3.1 Workshop Demo

The demo we presented at the workshop modelled a smart home environment containing two devices; A Hi-Fi and a display device such as a TV or a smart picture frame. The system is controlled through tokens (such as PDAs or mobile phones) carried by all the principals. The system was simulated using two laptops, one with the smart devices and the other showing the interface on the tokens.

This demonstrated the techniques of multi-factor authentication (having control of the PDA, being able to read the TV screen and being able to press a button on the Hi-Fi) to locate the guest inside the house. It also demonstrated the classification of functions according to their security relevance. Authorization for performing restricted functions was via a request to an identified administrator, caching this authorization for a period of time so that the permission did not need to be requested on subsequent invocations of that function.

### 3.2 Further Demos

We hope to expand this Demo into an environment which is closer to our target environment of a smart home and to involve users who are not members of our research group. Interviewing these users will give us valuable feedback about how the research should progress.

We are planning a deployment in which we can periodically evaluate the demo and then add features and improve the design based on the user feedback we collect. The evaluation will involve interviewing a selection of the users in the study to identify in which areas our design has succeeded and where it has failed. To this end we have started to design the questions we plan to use to collect useful feedback about our system design and on whether we are achieving a reasonable balance between security and usability.

### Questions for Guests

- **Identification**
    - How inconvenienced are you by having to prove you are inside the home?
    - How easy was it to understand/do?
    - How easy/understandable was the multi-factor authentication? (depends on authentication method used)
- **Permissions**
    - Was it obvious what you could/couldn't do?
    - Was it easy to gain extra permissions?
    - Did you have to do this too often?

### Questions for Administrators

- Did you have to grant anything you didn't want to?
- Was there anything you wanted to grant but couldn't?
- How easy was it to grant the correct permissions?
- Would you like to have done more in advance and less later?
- Would you like to have done less in advance and were happy to do more later?
- Were there areas where you would have liked to have finer grain control?
- Were there areas where the control was too complicated?

## 4 Conclusion

Smart Homes are on the verge of becoming a reality and, like all new systems, they will start off without much security. No satisfactory solutions to the problem of defining permissions for guests have been produced yet; but the issue needs to be solved, otherwise hosts will simply be forced to disable security features in order to accommodate their guests.

We aim to produce a solution that is flexible, easy to use and matches closely with the existing models of appliance use. We believe the correct balance between usability and security can only be reached through several iterations of user testing: we must allow non-experts to try out our ideas and tell us whether we got them right or not.

## Acknowledgements

We are grateful to Alan Blackwell for precious advice about our future user study.

## References

1. Alastair R. Beresford. “Location privacy in ubiquitous computing”. Tech. Rep. UCAM-CL-TR-612, University of Cambridge, Computer Laboratory, Jan 2005. http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-612.pdf.
2. J. R. Cooperstock, K. Tanikoshi, G. Beirne, T. Narine and W. Buxton. “Evolution of a Reactive Environment”. In “Proc. of CHI-95”, pp. 170–177. Denver, CO, 1995. http://www.billbuxton.com/ReactEnv.html.
3. Richard Harper (ed.). *Inside the Smart Home.* Springer, 2003. ISBN 1-85233-688-9. http://www.springer.com/uk/home/computer/user+interfaces?SGWID=3-154-22-2277285-detailsPage.
4. Andy Harter, Andy Hopper, Pete Steggles, Andy Ward and Paul Webster. “The Anatomy of a Context-Aware Application”. In “Mobile Computing and Networking”, pp. 59–68. 1999. http://www.cl.cam.ac.uk/Research/DTG/publications/public/files/tr.1999.7.pdf.
5. Jeffrey Hightower and Gaetano Borriello. “Location Systems for Ubiquitous Computing”. *IEEE Computer*, **34**(8):57–66, August 2001. http://seattle.intel-research.net/people/jhightower/pubs/hightower2001location/hightower2001location.pdf.
6. IEEE 802.15 WPAN Task Group 4. *ANSI/IEEE 802.15.4-2003, Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks.* IEEE, New York, Oct 2003. http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf.
7. Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner and Wendy Newstetter. “The Aware Home: A Living Laboratory for Ubiquitous Computing Research”. In “Cooperative Buildings”, pp. 191–198. 1999. http://www.parc.com/zhao/stanford-cs428/readings/apps/Kidd_AwareHome_cobuild99.pdf.
8. Abigail Sellen, Rachel Eardley, Shahram Izadi and Richard Harper. “The whereabouts clock: early testing of a situated awareness device”. In “CHI ’06: CHI ’06 extended abstracts on Human factors in computing systems”, pp. 1307–1312. ACM Press, New York, NY, USA, 2006. http://doi.acm.org/10.1145/1125451.1125694.
9. Frank Stajano. *Security for Ubiquitous Computing.* John Wiley and Sons, Feb 2002. ISBN 0-470-84493-0. http://www.cl.cam.ac.uk/~fms27/secubicomp/.
10. James Weatherall and Alan Jones. “Ubiquitous Networks and their Applications”. **9**:18–19, Feb 2002. http://www.realvnc.com/~jnw/papers/wireless.ps.gz.
11. Ford-Long Wong and Frank Stajano. “Multi-channel protocols”. In “The Thirteenth International Workshop on Security Protocols”, April 2005. http://www.cl.cam.ac.uk/~fms27/papers/2005-WongSta-multichannel.pdf. To appear.