A Secure Interoperable Architecture for Building-Automation Applications

Steffen Wendzel[1], Thomas Rist[1], Elisabeth André[2] , Masood Masoodian[3]
[1]Department of Computer Science, University of Applied Sciences Augsburg, Germany
[2]Department of Computer Science, University of Augsburg, Germany
[3]Department of Computer Science, The University of Waikato, New Zealand
{steffen.wendzel1, thomas.rist}@hs-augsburg.de
elisabeth.andre@informatik.uni-augsburg.de
masood@cs.waikato.ac.nz

ABSTRACT

Building-automation systems enable building administrators and inhabitants to monitor and control their buildings from within the building itself, as well as from outside using remote access tools. In recent years, many such control systems have been developed for both residential and nonresidential buildings, with increasing levels of functionality. Due to the growing number of building-automation system manufacturers, which use different network and communication protocols with no interoperability, it is often not possible to integrate the use of components from different manufacturers in a single building. To overcome this major limitation, we have developed a multilayer architecture for building-automation, designed to allow remote management of buildings, while making it possible to use components from different manufacturers. Our architecture implements an advanced secure interface for building-automation software, using secure event-handling and Role-Based Access Control (RBAC). The architecture is designed to provide energy consumption and monitoring applications with an interface protecting their privacy.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Applications; K.6.5 [**Security and Protection**]: Unauthorized access

General Terms

Building Automation Systems (BAS), Energy Awareness

1. INTRODUCTION

There are currently many building-automation products in the market, including the HomeMatic (*homematic.com*), SyncoLiving (*www.siemens.com/syncoliving*), the consumption monitoring system CurrentCost (*www.currentcost.com*), and SmartHome (*www.rwe-smarthome.de*), just to mention a few. These systems aim to support inhabitants of automated buildings by helping them to save energy, as well as assisting them with carrying out simple or complex automated tasks. Originally most Building-Automation Systems (BAS) aimed to automate only climate control operations such as heating, ventilation, and air-conditioning (HVAC) [4]. However, these days there are systems that provide other building-automation facilities like Physical Access Control (PAC). Complete building-automation, therefore, generally requires combining components from different manufacturers that have very little interoperability. Furthermore, this lack of interoperability in turn also leads to lack of support for communication and network security between different building-automation components.

Although an architecture for interoperability in the context of building-automation has been presented in [9], it does not take into account any security aspects. A security analysis of building automation protocols has been given in [3]. Hoffmann et al. have combined both aspects to develop an interoperable as well as secure middleware [6]. They have also introduced Role/Attribute-Based Access Control (RBAC/ABAC) to the field of building-automation.

We have developed a similar architecture to that of Hoffmann et. al., but with a specific focus on providing a lightweight adoptable design and an underlying communication protocol for supporting energy consumption and monitoring application in domestic home environments [7]. Our architecture allows combining automation mechanism provided by hardware from different vendors. Using this architecture, in a demonstrative prototype called the *Home Analytical System Interface* (HASI), we have implemented support for the home automation system HomeMatic as well as for the energy consumption monitoring system CurrentCost.

Additionally, in a related project (*USEM*) we have developed a similar architecture [5]. While HASI and USEM conceptually share a layered architecture, each follows its own implementation strategy. The USEM communication infrastructure relies on the JSON-RPC technology and enables a two-way communication between event and data recording and client applications. The advantage of this implementation approach is that client applications register with the infrastructure to receive certain event notifications rather than having to continuously poll for new information updates (see [5] for more details).

In the remainder of this paper we only focus on HASI,

ISABEL '11, October 26-29, Barcelona, Spain.
Copyright 2011 ACM 978-1-4503-0913-4

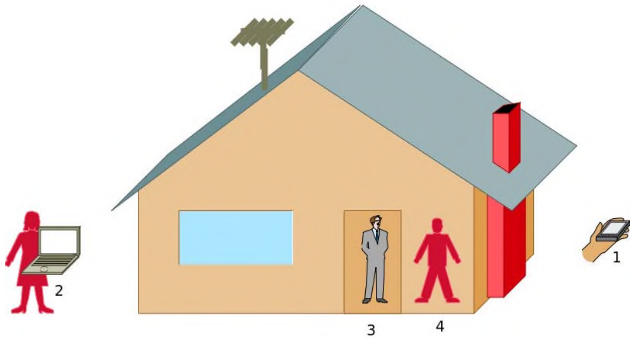


Figure 1: Internal and external attackers and users of a building's automation system: 1. Remote user, 2. external attacker, 3. inhabitant (not attacking), 4. internal attacker.

as the additional security features described here are specific to HASI. HASI has a clear multilayered architecture for building-automation, containing both a security module to implement role-based access control, as well as a network protocol able to deal with different security roles. Applications implemented using this architecture access a unified API that abstracts all low-level hardware communication, security functions, as well as underlying protocol differences (similar to that of Hoffmann et. al. [6]). When implemented in a number of rooms, hardware from different manufacturers can be added to provide new functionality, and new inhabitants can be added to the system by linking them to roles within the security model. Our system supports an unlimited number of rooms and buildings; so that if for instance a building administrator is responsible for more than one building, a central system for controlling and monitoring can be deployed.

The remainder of this paper is structured as follows: Section 2 introduces the attacker model while Sections 3 and 4 describe the multilayer architecture, as well as the security mechanism we have implemented. Section 5 discusses HASI-based applications and Section 6 provides some conclusions.

2. ATTACKER MODEL

In our model, we focus on both, internal as well as external attackers (Figure 1) within the *high-level* control systems, i.e. attackers aiming to attack remote interfaces as well as building-internal control interfaces. Building-internal interfaces are mandatory to provide inhabitants with a means of managing the BAS. Nowadays, remote interfaces are common as well (e.g. for turning on the building heating before an inhabitant comes back home from work).

In contrast to high-level attacks, *low-level* attacks include for instance the insertion of malicious messages (faked control command messages and faked responses, such as wrong temperature measurements), or eavesdropping to get information about actions taking place within a building.

The reason for not focusing on low-level attacks is that existing techniques do already provide low-level protection means. For instance, EIBsec enables a secured communication by ensuring data integrity, confidentiality, data freshness as well as authentication [3]. Additionally, we do not focus on physical access control systems (these systems are used to control access to restricted areas) since they are already available as well (e.g. with BACnet [8]).

High-level attacks aim to exploit security bugs in user-interfaces (mostly HTTP-based). Common BAS user interfaces do of course provide user-based access control, but do not implement role-based access control (RBAC) nor privacy protection means. Although some work in the area of RBAC in embedded building automation has been done by Hoffmann et. al. in the context of the *Hydra* project [6], this is mainly a general approach and does not focus on the specific field of privacy control for energy consumption and monitoring applications that we are concerned with here.

In the context of our work, privacy is important since a BAS must ensure that energy consumption information are kept private and are visible to the consumer only. If this privacy protection is not provided, an attacker could, for instance, access information about when a user has used an electronic device and for how long, or when a user has turned the heating on, etc. Although it is possible to think of many scenarios where unauthorised access to such information would be considered harmful, here we provide one case scenario: a company aims to save energy by evaluating the energy reduction of each of its departments individually. However, the desire to save energy differs from employee to employee, i.e. some employees in a department will save more energy than others. Providing personal energy consumption information of everyone to all employees within a department can lead to annoyance between the employees due to the fact that ambitious employees may notice energy wasting behaviour of others.

The following section describes our privacy enhancing architecture which aims to prevent remote and local attacks by using an RBAC-capable middleware. A detailed analysis of security threats for building-automation systems can be found in [4].

3. MULTILAYER ARCHITECTURE WITH SECURITY CONTROL

As shown in Figure 2, our multilayer architecture provides an interface for different types of applications, e.g. for energy consumption and monitoring applications. All these applications access the system via the same API calls in the *Unified Application Programming Interface* (UAPI) layer. The UAPI layer accesses the management layer, called the *Unified Control interface Service* (UCIS) layer via TCP sockets using SSL. UAPI abstracts all low-level socket actions and multiplexing. It is also capable of handling multiple applications at the same time.

The UCIS layer receives all network messages from the UAPI layer, interprets them, authenticates connections, verifies security permissions via roles, monitors all activities, keeps history state information in a database (MySQL), and accesses all hardware components of the supported manufacturers. Such hardware components can be located in different buildings and rooms (even in remote locations). The monitoring option at the UCIS layer is capable of generating energy consumption information for all the rooms in different buildings. Such monitoring capabilities result in the possibility of generating organization-wide energy consumption information that can then be compared to each other. Therefore, direct access to the UCIS layer is only possible for building administrators. The communication between UAPI and UCIS layers is done via a security protocol as

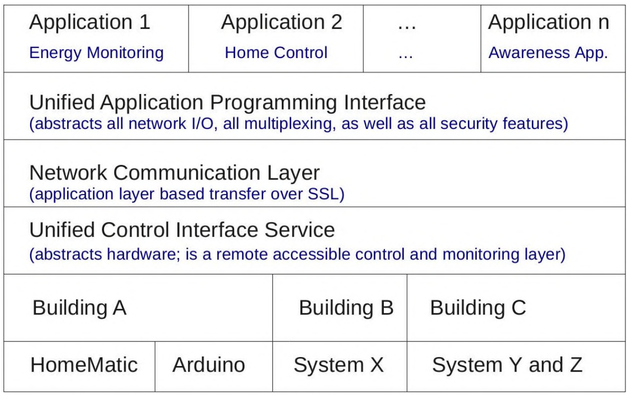

Application 1
Energy Monitoring
Application 2
Home Control
...
...
Application n
Awareness App.
Unified Application Programming Interface
(abstracts all network I/O, all multiplexing, as well as all security features)
Network Communication Layer
(application layer based transfer over SSL)
Unified Control Interface Service
(abstracts hardware; is a remote accessible control and monitoring layer)
Building A
Building B
Building C
HomeMatic
Arduino
System X
System Y and Z

Figure 2: Multilayer Architecture for Building Automation

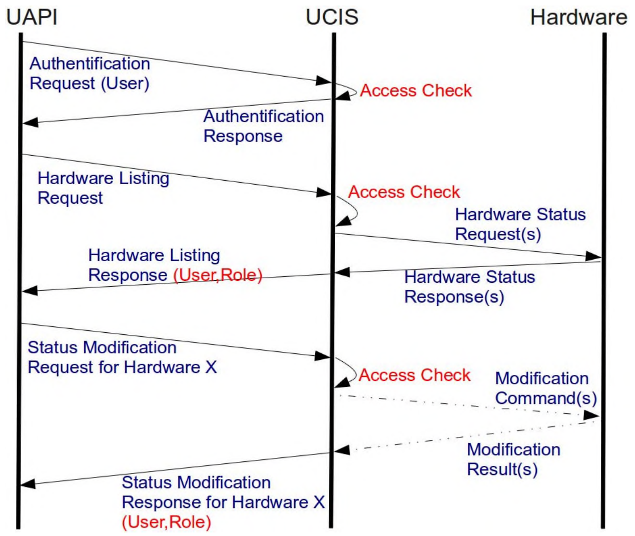

UAPI
UCIS
Hardware
Authentification Request (User)
Access Check
Authentification Response
Hardware Listing Request
Access Check
Hardware Status Request(s)
Hardware Listing Response (User,Role)
Hardware Status Response(s)
Status Modification Request for Hardware X
Access Check
Modification Command(s)
Modification Result(s)
Status Modification Response for Hardware X (User,Role)

Figure 3: Communication of the Secure Multi-Building Management Protocol (SMBMP)

described in Section 4.

Although clearly an architecture built on top of existing hardware components is not able to fix security problems of those components, our approach focuses on the security of higher level interfaces and on the unification of security features. To provide applications with a unified security service, events, such as adding or removing low-level hardware components are also handled by our multilayer architecture. Therefore, an application has no access to the low-level information in a direct way, but can nevertheless react to a triggered event using the notification system. All occurring events are reported through a logging interface and can be sent to the central logging system.

4. SECURITY PROTOCOL AND MODULE

As mentioned above, our protocol uses an SSL encrypted application-layer connection via TCP. In this request-response type protocol, the UAPI layer sends requests to the UCIS layer, which in turn answers them with a response message containing the result. The protocol is named *Secure Multi-Building Management Protocol* (SMBMP).

Figure 3 illustrates the typical situations that occur when SMBMP is used. At the beginning of each transaction, an application needs to authenticate itself at the UCIS layer using the UAPI. If the user is allowed to communicate with UCIS, access will be granted. To get the current list of hardware available within all the buildings, the application needs to send a *Hardware Listing Request* (HLR) using the UAPI. If UCIS receives a HLR, it checks the permissions of the authenticated user. All hardware information, a user (or role) has read-permission for, is then read from the hardware layer and transferred to the UAPI. If a single hardware, a single room, floor or a whole building is not accessible to a given user (or the related role), access will not be granted (e.g. most employees will only be allowed to control building-automation objects in their own office or private rooms).

If a status modification (i.e. an action like “open the kitchen window” or “turn off lightening in the living room”) is received, an access verification (for modification rights of the user and the associated role) is done by the UCIS layer. The response indicates whether a modification request was successful.

Further to the simple role based access control, the security system in HASI also handles the verification of values within a user’s requests. For instance, in case a user configures the heating to warm up a room to 29 degrees, when it is only allowed to warm up to 20 degrees, the request is rejected.

To implement these features, SMBMP requests contain a number of mandatory parameters. While an authentication request contains a username and a hash value of a password, the modification requests contain a building identifier, a hardware identifier, a request type, the number of included values (n) to modify, as well as a vector of n values. Our communication protocol is designed to be backward compatible and flexible to support new classes of building-automation components. The UAPI layer’s design is based on these dynamic needs.

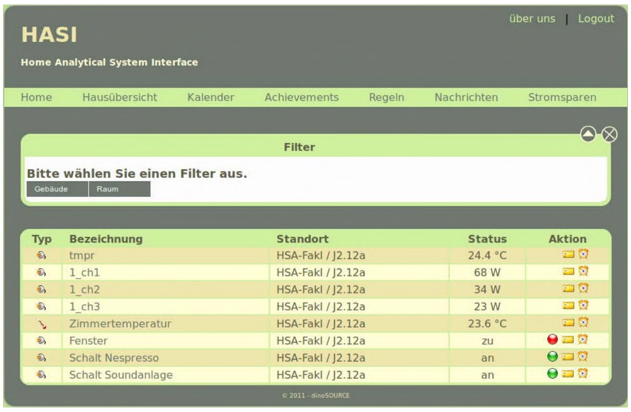
However, SMBMP and the HASI security module do not provide security features for all kinds of high-level threats. For instance, they do not focus on covert channels (these are malicious communication channels which are not foreseen by a system designer and which do not stay conform with a given security policy, c.f. [10]), or on side channel protection within HASI itself (an approach to detect side-channels in network applications has been given in [2]) and Denial-of-Service attacks [4].

5. HASI-BASED ENERGY AWARENESS

As mentioned earlier, a HASI demonstrative prototype has been developed using this multilayered architecture, to provide support for the home automation system HomeMatic and the energy consumption monitoring system CurrentCost. This HASI prototype is a light-weight Linux-based system written in Python and containing a MySQL backend. It is designed to run on small hardware to reduce its own energy consumption. Local and remote control of these management and monitoring systems are done by a dynamic configurable web-interface.

5.1 Energy-Aware and Energy-Efficient User

While the need for more efficient use of energy is commonly accepted, for the individual it is often difficult to act



HASI

Home Analytical System Interface

über uns | Logout

Home Hausübersicht Kalender Achievements Regeln Nachrichten Stromsparen

Filter

Bitte wählen Sie einen Filter aus.

Gebäude Raum

Typ	Bezeichnung	Standort	Status	Aktion
	tmpr	HSA-Fakl / J2.12a	24.4 °C	▭ ⏰
	1_ch1	HSA-Fakl / J2.12a	68 W	▭ ⏰
	1_ch2	HSA-Fakl / J2.12a	34 W	▭ ⏰
	1_ch3	HSA-Fakl / J2.12a	23 W	▭ ⏰
	Zimmertemperatur	HSA-Fakl / J2.12a	23.6 °C	▭ ⏰
	Fenster	HSA-Fakl / J2.12a	zu	🔴 ▭ ⏰
	Schalt Nespresso	HSA-Fakl / J2.12a	an	🟢 ▭ ⏰
	Schalt Soundanlage	HSA-Fakl / J2.12a	an	🟢 ▭ ⏰

© 2011 - dinoSOURCE

Figure 4: Snapshot of an early HASI-based web application. Sensor status information is presented as a table.

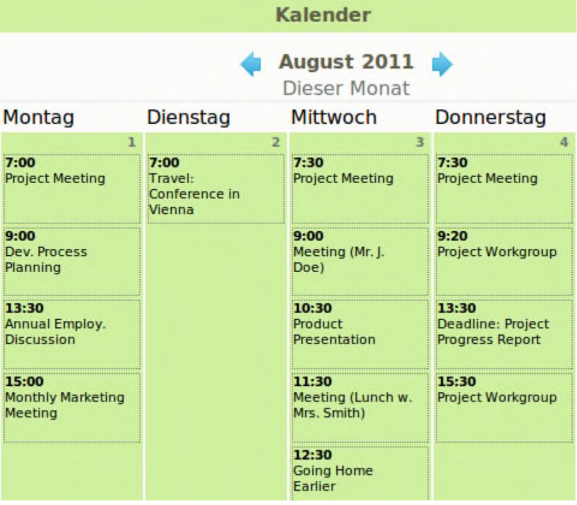
accordingly. One reason for this is generally the lack of information about the energy costs of a certain activity, such as leaving on lights or electrical appliances in stand-by mode. So-called eMetering devices aim at filling this information gap. Mounted between a socket outlet and an electrical appliance they measure power consumption and may translate consumption values into monetary costs. However, the provision of information about energy consumption alone does not necessarily imply acting. Rather, assisting people in actually changing their behaviour is a much harder challenge.

The question of how to utilize technology to assist people in changing their behaviour, especially to get rid of bad habits, is dealt within the field of "persuasive computing". BJ Fogg, a pioneer in this field, postulates the statement "Put triggers in the path of motivated people" as a design mantra for behaviour change [1]. According to Fogg's behaviour model, for a behaviour change to happen three factors must coincide: the person must be motivated in principle, a trigger must be present that just-in-time reminds the user to do the right thing instantly, and the person must be able to act. Adopting this model for the purpose of creating more energy-efficient user behaviour means:

- motivation: strengthen users' motivation to save energy and keep them motivated;
- trigger: identify opportunities at which users should perform actions that contribute to efficient energy use, and provide appropriate notifications as triggers;
- enabling: set-up of technical infrastructure and usable services as enabling means for the execution of actions as easily as possible.

Given the task of informing users about their energy consumption one might at first think of designated dashboard displays or data browser which allow users to access consumption values on demand. Indeed, in our HASI and USEM prototypes, a user is given the option of browsing through sensor data, e.g. using a classical hierarchical menu structure. For instance, Figure 4 shows a snapshot taken from an early HASI-based web interface which allows browsing through sensor status information.

The major problem with such an approach is, however, that it assumes an always actively involved user – which is certainly a too strong assumption given that users are often concerned with other things than just saving energy. Therefore, a different, and perhaps more promising strategy, is to follow Fogg's design mantra and find opportunities for puttin triggers in the path of motivated people. In the case of HASI and USEM opportunities are primarily in applications that are used almost daily by individuals on their computers or smart phones, for instance to access their calendar, email, or social networks. The main idea here is to incorporate into such applications information on energy consumption and provide advice on how to reduce energy consumption.



Kalender

August 2011

Dieser Monat

Montag	Dienstag	Mittwoch	Donnerstag
1	2	3	4
7:00 Project Meeting	7:00 Travel: Conference in Vienna	7:30 Project Meeting	7:30 Project Meeting
9:00 Dev. Process Planning		9:00 Meeting (Mr. J. Doe)	9:20 Project Workgroup
13:30 Annual Employ. Discussion		10:30 Product Presentation	13:30 Deadline: Project Progress Report
15:00 Monthly Marketing Meeting		11:30 Meeting (Lunch w. Mrs. Smith)	15:30 Project Workgroup
		12:30 Going Home Earlier	

Figure 5: Close-up of the HASI-based augmented calendar system. Entering a new appointment after regular business hours triggers a notification and asks the user whether switching on the heating system at home should be delayed.

5.2 SmartLiving

In a related project called *SmartLiving* we have developed a series of energy awareness applications using the HASI system. The goal of SmartLiving is to use HASI's trigger functionality (cf. Section 3) to develop applications that change peoples' energy consumption behaviour. Three applications have been developed so far: an augmented calendar system, Smart Living TV, and Smart Garden.

Using the *augmented calendar system*, a user can associate certain time slots with notifications relevant to energy use (cf. Figure 5). A standard use case is the addition of new appointment after regular business hours which implies leaving for home later than previously planned. HASI can register this event, and the user is asked whether switching on the heating should be delayed due to the additional calendar event. Of course, a user could achieve the same effect with two different applications, i.e., a conventional calendar system and a separate remote home-control application. However, the question is whether the user would do so. The HASI solution at least has the advantage of minimizing the user's distraction and extra effort required for acting. HASI provides relevant information just-in-time and execution of the energy-saving action is just a click away from the user's primary path, i.e., entering a further appointment.

Smart Garden is a dynamically changing picture of a gar-

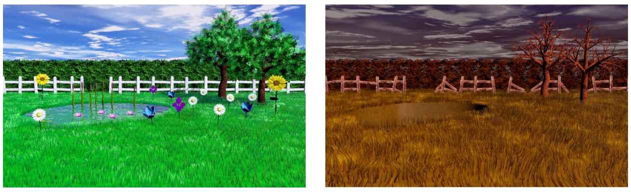


Figure 6: Ambient visualisation of energy consumption. Left: flowering garden indicates consumption below average. Right: consumption above average.

den. It is designed as an ambient display which is sensitive to a user‘s measured energy consumption. If the overall consumption is below average the garden starts flourishing, while consumption above average results in a withering garden (cf. Figure 6). The Smart Garden is used as a background image for the key-holder panel but may also be displayed in a digital photo frame.

The idea behind *Smart Living TV* is to substitute in an ordinary TV programme commercials by automatically generated personalized info clips about a person's energy consumption. The interesting point here is that unsolicited commercials are usually perceived as annoying obstacles in a user‘s path towards viewing their programme. One reason may be a lack of relevance for a particular viewer. Replacing such a commercial by a personalized information clip may be at least less annoying, if not an acceptable means of increasing a user‘s energy awareness. So far, for illustration purposes we have manually created a sample clip and spliced it into a recorded TV programme using a video editing tool. Automated clip generation and real-time programme manipulation (e.g. using the open source VDR software) has not yet been implemented.

6. CONCLUSION

This paper presented a building-automation architecture and its implementation called *HASI*, that is designed to provide interoperability between building-automation hardware from different vendors, as well as a security mechanism for protecting users' privacy. HASI provides a unified API that communicates with the base system using our *Secure Multi-Building Management Protocol* (SMBMP) and implements Role Based Access Control (RBAC).

The HASI architecture serves as a base for the development of applications in the context of energy consumption and monitoring. The project SmartLiving has led to the development of a set of HASI-based applications which use the mantra of Fogg (putting triggers in the path of motivated people) with the aim of changing people's energy use behaviour.

Future work will include the development of additional augmented concepts and their evaluation in the area of energy awareness. Furthermore, we are working on an improved HASI system which aims to prevent covert channels and side channels in building automation.

Acknowledgements

We like to thank all members of the project teams at the University of Applied Sciences Augsburg, the University of Augsburg, and the University of Waikato for their valuable development work.

This research was supported by the IT4SE research cooperation (Grant number NZL 10/803 IT4SE) under the APRA initiative funded by the German Federal Ministry of Education and Research (BMBF). More information can be found on the IT4SE project website: *http://www.it4se.net/index.html*.

7. REFERENCES

[1] B. Fogg. The new rules of persuasion, In RSA Digital Journal. `http://www.thersa.org/fellowship/journal/archive/summer-2009/`, Summer 2009.

[2] F. Freiling and S. Schinzel. Detecting hidden storage side channel vulnerabilities in networked applications. In J. Camenisch, S. Fischer-Hübner, Y. Murayama, A. Portmann, and C. Rieder, editors, *Future Challenges in Security and Privacy for Academia and Industry*, volume 354 of *IFIP Advances in Information and Communication Technology*, pages 41–55. Springer Boston, 2011.

[3] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus. Security in Networked Building Automation Systems. In *Proc. 6th IEEE Int. Workshop on Factory Comm. Systems*, pages 283–292, 2006.

[4] W. Granzer, F. Praus, and W. Kastner. Security in building automation systems. *Industrial Electronics, IEEE Transactions on*, 57(11):3622–3630, November 2010.

[5] M. Kugler, F. Reinhart, K. Schlieper, M. Masoodian, B. Rogers, E. André, and T. Rist. Architecture of a ubiquitous smart energy management system for residential homes. In *Proc. 12th CHINZ Conference*, pages 283–292, 2011.

[6] A. Maña, C. Rudolph, M. Hoffmann, A. Badii, S. Engberg, R. Nair, D. Thiemert, M. Matthess, and J. Schütte. Towards semantic resolution of security in ambient environments. In *Developing Ambient Intelligence*, pages 13–22. Springer Paris, 2008.

[7] T. Rist, S. Wendzel, M. Masoodian, and E. André. Creating awareness for efficient energy use in smart homes. In G. Feuerstein and W. Ritter, editors, *Intelligent Wohnen. Zusammenfassung der Beiträge zum Usability Day IX*, pages 162–168, 2011.

[8] D. Ritter, B. Isler, H.-J. Mundt, and S. Treado. Access control in BACnet. *BACnet Today*, November 2006.

[9] D. Snoonian. Smart buildings. *IEEE Spectrum*, 40(8):18–23, August 2003.

[10] S. Wendzel and J. Keller. Low-attention forwarding for mobile network covert channels. In B. de Decker et. al., editor, *Proc. 12th Conference on Communications and Multimedia Security (CMS 2011)*, volume 7025 of *LNCS*, pages 122–133. IFIP International Federation for Information Processing, Springer, October 2011.