Ubiquitous Smart Home Control on a Raspberry Pi Embedded System

Jan Gebhardt, Michael Massoth, Stefan Weber and Torsten Wiens
Department of Computer Sciences
Hochschule Darmstadt - University of Applied Science
Darmstadt, Germany
{jan-michael.gebhardt | stefan.b.weber}@stud.h-da.de
{michael.massoth | torsten.wiens}@h-da.de

***Abstract*—This paper describes an approach to use the embedded system Raspberry Pi to serve as a communication gateway between mobile devices and Konnex-Bus (KNX) home automation systems. The Session Initiation Protocol (SIP) and the Presence Service are used to build a system concept on open source and standardized software services. The concept focuses on the communication, access control and security of that gateway. This paper manifests the possible components and potential purposes of the concept. It is shown that small embedded systems like the Pi can provide a simple and cheap solution to enable ubiquitous Smart Home Control using existing infrastructures.**

Keywords-KNX; SIP; Smart Home; Raspberry Pi.

I. INTRODUCTION

The Konnex Bus (KNX) standard has been the de facto standard in home automation systems for many years. The system is widely used for new installations and has been retrofitted into many existing buildings. The standard is open, internationally accepted and standardized in several countries [1]. KNX is probably the most used building automation system on the market. The Internet brought new technologies for communication between people. The underlying technologies and protocols can also be used to communicate with machines. By merging these technologies, we get an intelligent or ”smart” home, which reflects a current trend in information technology. A smart home shall enable interaction with its owner, including the ability to monitor the status and control of home appliances and devices remotely from anywhere in the world. Such devices may consist of alarm systems, keyless access control, smoke detectors, light, heat, water or other energy management systems, medical devices, and all types of sensors, e.g., room-, door-, window- or security surveillance, monitoring and control, statistics and remote metering to every automated system and appliance in the home. With the increasing availability of smartphones and access to the Internet at any time, it is reasonable to use these devices to remotely control our smart home. The research of this project was focused on the development of a KNX-to-SIP proxy, to interconnect the home automation system with new communication protocols. The software should run on an embedded system, such as the Raspberry Pi [2], to ensure low resource consumption and to be cost-effective. Furthermore, the whole system should be compliant with open standards.

A. Purpose and Relevance

The purpose of this paper is to show that small embedded systems, in our case the Raspberry Pi, can be used to run the smart home software, developed within this project. Furthermore, a new communication model is introduced, which aims to improve the resource-consumption within the Session Initiation Protocol (SIP) [3]. More precisely, the actor and sensor information was stored within separate SIP profiles, which can be improved. The fact, that the presence information is visible to all SIP users, generates a need for a detailed security concept. This design includes a draft to implement access control, as well as communication security into the home automation remote control framework.

B. Structure of the Paper

Following this introduction, Section II describes related work and other interesting projects suitable for this concept. In Section III, an overview of the general approach is given. The components of a possible system design are discussed in Section IV. After that, we introduce two use-cases to create a basis for our concept, which will be described in Section VI, leading to a system design. Next, an overview over the possible communication security layers is given in Section VII. Sections IX and X conclude the paper and give an outlook on future work.

II. RELATED WORK

The concept of intelligent or smart home control has been well-established in IT. Many companies and institutions are working on solutions or even released their individual software. Some of these systems also come with their own hardware sensors and actors to create a Smart Home. Most of these software solutions are proprietary and not compatible with other home automation systems. With the approach developed by this project, it is possible to use the existing home automation system KNX to enable secure remote control. Whereas other projects like MavHome [4] aim to create the intelligence of a smart home, we use the existing intelligence, provided by the home automation system itself. The communication with remote clients is enabled by using existing infrastructure and open standards, such as SIP [3]. Henning Schulzrinne et al. presented how ubiquitous computing could be integrated into home networks with SIP [5]. Also, an IETF Working Group

 ISBN: 978-1-61208-353-7

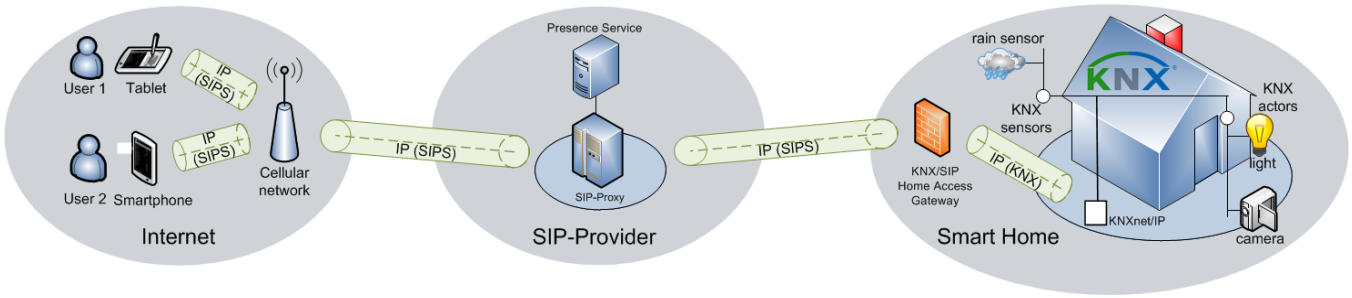

Figure 1. System architecture and components of our Smart Home.

described that future buildings will probably be equipped with a full featured IP network. They also chose SIP as the communication protocol [6]. In previous steps of the project, a proof-of-concept has been implemented [7]. The information describing the Smart Home components' status is stored within the Presence Information of a SIP user, and other SIP users can subscribe to this information via the Presence service. The existing implementation uses a separate server system as a host for the SIP proxy and the KNX/SIP Bridge. An Android tablet with a self-developed application is used to communicate with the setup. The HomeSip project described the use of SIP as a communication middleware to support home automation applications [8]. The concept is similar to our approach, where a SIP proxy is used as a gateway to enable communication between the devices. Another way to connect to the home automation network over the Internet would be to establish a Virtual Private Network (VPN) connection. In this case, the external host is virtually placed into the internal network and is able to directly communicate with the KNX/IP-bus. This solution requires a continual connection to the network. With our approach, there is no need for a continual connection. If a state change occurs, the client receives a push notification specified within the standard SIP protocol. This significantly reduces the connections in comparison to a traditional pull/continual configuration. Furthermore, a VPN connection by default enables the client to get access to all internal network devices. With our approach, we introduce access control lists to the home automation gateway, which eliminate this lack of security.

III. Approach

The new approach aims to use the standardized SIP server technology to show that our implementation does successfully interact with standard SIP proxy in the intended way. The approach will eliminate the need for a separate SIP server within the Smart Home. It is possible to use an external SIP account from any service provider. Therefore, the new implementation uses FreeSwitch as a SIP proxy with presence service. For efficiency and resource-optimization, we now use a new way to store the whole information about all sensors and actors inside the smart home within the presence state of one single SIP account. By using the Raspberry Pi as an embedded system, the separate server system is obsolete. In addition to these changes, we introduce access control, as well as general security to the project.

IV. Components

The following section details all components of our approach and sketches a general overview over the used technologies. The whole architecture of the project is shown in Figure 1.

A. Mobile Clients

Mobile Clients are used to connect to the Smart Home. Figure 1 illustrates, that several Mobile Clients can connect to the Smart Home simultaneously over various access networks. With the self-developed client, it is possible to connect from anywhere in the world to the Smart Home. The software uses SIP/SIPS to communicate with the presence service of the SIP provider to gather the information about the Smart Home. Based on that information, the KNX-sensors and -actors are displayed to the user, and it is possible to interact with actors.

B. SIP-Provider

The basic advantage of our concept is the usage of the presence service implemented as an extension to the SIP protocol. It enables event-based notifications in near real-time. The sensor and actor states are stored inside the presence information of the corresponding Smart Home SIP user. Push messages are sent whenever a sensor or actor changes its state. SIP providers already maintain the infrastructure, basically consisting of a SIP proxy and the Presence Service. The KNX/SIP Home Access Gateway sends all information about the Smart Home to the Presence Service.

C. KNX/SIP Home Access Gateway

The main task of the device is to function as a gateway between the two technologies KNX and SIP. On one side, the KNX/IP-Bus is monitored for state changes. It is also possible to write on the bus, for example to switch the light on. On the other side, the relevant information is published to the SIP Presence Service. This device also enforces the security concept detailed in section VII. The embedded system Raspberry Pi is used as platform for the KNX/SIP Home Access Gateway. The Raspberry Pi was developed by Raspberry Pi Foundation from the UK. It is a small embedded system, operating at 700Mhz, with a graphic chipset able to render up to 1080p [9]. It is a cheap but also fully functional computer system, whose performance is sufficient to run the Smart Home software. Furthermore, it is capable to be extended with other features by using the built-in GPIO-Pins [9]. As an additional feature,

 ISBN: 978-1-61208-353-7

the chipset used in this unit is equivalent to a chipset used in cellphones, which does not need additional cooling. This is a benefit for our concept, because it is silent and can be put into the electric cabinet of the Smart Home. In our concept, the Raspberry Pi is acting as a gateway between the KNX- and the SIP-protocol.

V. USE CASES

Our concept, especially our security concept as improvement to known techniques, is based on the following use cases. These scenarios focus on the access to the Smart Homes, particularly regarding the security aspects.

A. Facility Manager

As one example, we selected a facility manager, who needs to control a lot of facilities, for example Smart Homes. When we transfer this to our university, every one of its facilities be assumed a unique Smart Home, which is controlled by a facility manager. The buildings are open from 8:00 to 19:00 o'clock. If there is a lecture before or after that time, a special building has to be opened by someone. Because of this, the facility manager has to open these buildings with his user account. On the other hand, he has to monitor the state of all doors or lights to close them or turn them off at the end of the day to save energy. Besides the facility manager, every department has to be able to control their own buildings to manage the lecture halls. Because of this, there is a need of splitting actors and sensors into groups or merging Smart Homes to a bundle under the consideration of access rules. In Section VII, we will introduce such a feature.

B. Guestroom

As another example, we selected a scenario, where we have a guest at our Smart Home. Displayed in Figure 1, we assume that User1 is the Smart Home owner and User2 is our guest. The Smart Home is able to separate each room from another. Our guest should be able to control the guestroom during his visit, so he can set up the radiator to be warm, when he comes home. On the other hand, it is intended to limit his access only to relevant parts of the Smart Home. This motivates a configuration mechanism, allowing granting or denying access to specific actors and sensors, so the guest only is capable of controlling the guestroom. This leads us to the need of a technology to set up access conditions to actors and sensors, which will be introduced in Section VII.

VI. CONCEPT

Our concept aims to combine an independent Smart Home with next generation network techniques to reach a stable and secure connection over the Internet. The basic idea is to use systems already in place without any modification, so that additional implementation work is only necessary at the communication endpoints. Based on preliminary work by Massoth et al., called *"Ubiquitous Home Control based on SIP and Presence Service"* [7], we realized an enhanced example of combining a Smart Home with NGN-Technology. The main difference between this research and the work at hand is the exchange of data through the presence service. The past research needs one SIP profile for each sensor or actor, so the server-side effort is very high and not compatible to current implementations of providers. Within the new concept, all sensor data is firstly merged and then transferred to the presence service. This accumulation will be done by using a standardized XML-Scheme named Presence Information Data Format (PDIF), which will be described in the following section. Therefore this concept does not depend on cooperation with SIP providers, because it mainly adds Smart Home control functionality on top of the status information.

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:...">
  <dm:person id="p1">
    <dm:note>Available</dm:note>
  </dm:person>
</presence>
```

Figure 2. Simple PIDF-Extract.

A. FreeSwitch as SIP Proxy

To implement our concept, we chose FreeSwitch as a SIP proxy. FreeSwitch is one of the most commonly used SIP servers on the market. It provides a very high level of standard conformity, which is useful for thorough interconnection testing as intended. It is licensed under the Mozilla Public License (MPL). In comparison to the second leading VoIP-Daemon, FreeSwitch proves better stability on a higher scale of client-usage. FreeSwitch supports a bundle of modules to achieve communication through different protocols, e.g., SIP or XMPP. It also provides the presence service, which is actually needed by this concept.

B. Construction of the Communications Protocol

The Protocol is based on the PIDF, which is introduced in the next subsection. Basically, it is a standardized XML-Scheme to exchange status information over the presence service. Our concept is to use this protocol and further enhance it by embedding additional data into that scheme.

1) Presence Information Data Format (PIDF): PIDF as standardized XML-Scheme is used as previously explained to exchange status information through the presence service. The scheme is able to divide real persons from simple or complex devices, like an answering machine or a fax. Figure 2 depicts an example of a person's current availability status. Everyone who is subscribed to him will get this information.

2) JavaScript Object Notation (JSON): JSON offers a smart and compact way to store sensor data into a PIDF-Scheme. This scheme is highly compressed and furthermore can be interpreted by JavaScript as well. Our end-users directly benefit from this behavior, because not only JavaScript or a simple Webpage is able to interpret this scheme, but also very complex software in Java. They are both able to interpret it by default. The scheme is shown in Figure 3. It can encapsulate as many arrays as exist in a database. Variable names and values are separated by a colon, whereby variables are separated by a comma. Every value can thereby encapsulate an extended set of variables. Within this structure, every actor and sensor is represented by a variable with a set of extended variables. So, the *temperature-sensor* is saved as variable with two additional

 ISBN: 978-1-61208-353-7

```
{temperatur-sensor: {
          type: Double,
          value: 22.3}}
```

Figure 3. Example of JSON encoded sensor-data.

variables. These additional variables define the type and value of the *temperature-sensor*-data. Like *double*, we also defined simple data types like: *integer, string* and *boolean*.

3) Changing actor status: In comparison to the status monitor for sensor data, the process to change the status of the data does not need the presence service. To set a new status to an actor there is only the need to send a simple message to it. These messages contain the same JSON structure, as the presence information of sensors explained above. To compare this information, we can assume *radiator-control* instead of *temperature-sensor* in Figure 3.

C. Smart Home

As already mentioned, the Smart Home is connected through a gateway, which transcodes KNX bus data into SIP and our enhanced presence exchange protocol. Every sensor pushes its current status to the KNX bus, where the bridge reads and also pushes it to the presence service. Thereby, the bridge accumulates a group of sensors and converts their data into the JSON structure explained above, which then can be sent to the presence service. On arrival of that data, the presence service sends it to all subscribers, so the end-users are able to read the current statuses of their sensors at home. Figure 4 shows an example on how the information is exchanged. Furthermore, as a extension of that behavior, the user is able

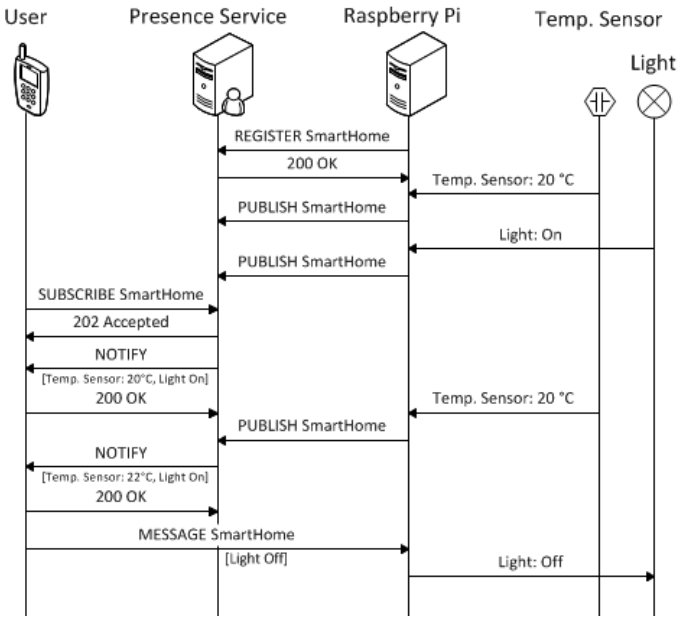

Figure 4. Smart Home information exchange.

to define these groups, like a set of all sensors and actors of one room, or to define an access control list for it. This is done by applying security layers as described hereafter.

VII. SECURITY

Nowadays, it is very important to prioritize the use of security techniques to achieve the key security concepts, like confidentiality or authenticity. In our example, we do not want anybody to control our Smart Home. To achieve this, a security concept is needed, which allows to grant access to specific persons or to prevent read and write access from third parties. It is also important to refrain from using known techniques, like a VPN tunnel, to control a Smart Home, because they only cover layer 1 of the following layers. With a VPN tunnel, the person who has access to it, has also the access to the whole Smart Home without any restrictions.

Besides a simple encryption of the connection to our Smart Home, we introduced two use-cases in Section V, which are motivating the need for an user-controlled access control list (ACL). This ACL helps us to define users who have access to specific sensors and actors. To achieve this, we elaborated a security layer concept as it is known from the OSI layer model [10].

Every layer needs to be applied on top of the lower layers, so the highest security is only reached by applying all layers of this concept. Figure 5 shows three examples on how to combine these layers. As can be seen, all layers are different

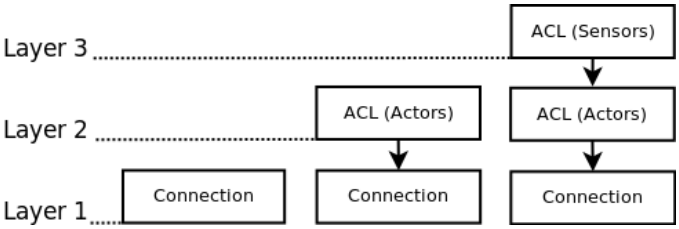

Figure 5. Security Concept - layer architecture

from each other and are needed to achieve different security goals. The exact function of each layer will be described in the next subsections. The main benefit of using such a layer concept is the ease of applying each layer on top of the others. When using this concept, one is not forced to apply all layers, but only the layers, which are needed in a specific scenario, to not overload and complicate the application. Another reason for choosing this layer concept is the separation of adaptions of the connection endpoints. Mostly, the first layer already exists in SIP applications and is supported by SIP providers. The second layer needs to be adjusted to the Smart Home, to allow the use of an ACL. At the end, the third layer requires an additional adaption at the Smart Home and the users control endpoint, because an encryption extension has to be added. In the following subsections, we are taking a closer look at these layers.

A. First Layer

The first security layer is done by the using the underlying protocol secure SIP (SIPS). SIPS enables basic encryption with the Transport Layer Security (TLS) protocol. As mentioned before, it provides the integrity and confidentiality of the communication between server and client. An attacker needs to gain access to an endpoint of the connection to decrypt or

manipulate the transferred data. This is the basic and recommended security layer, which does not need any adaptions on the communication protocol, because it is already implemented as a transport layer in almost every operation system, since they are implementing the OSI-Layer model [10].

B. Second Layer

To achieve access control to the actors, the second layer creates an access control list, which lists all users with access capabilities. Only users who are listed in that ACL are able to set a new state of an actor. For example, the user *owner@myhouse* is able to turn off all lights inside his house. Every command send from another user will be dropped. The whole ACL stays by the end user, so only the end user respectively the Smart Home is able to modify the ACL. So, the owner can add or remove users from it to grant or deny their access.

With this approach, the provider of SIP proxy cannot modify the status of any actor of the Smart Home, which is a security benefit. If we look at the second use-case in Section V, we have a guest at our Smart Home. That shows us the need of granting access to specific actors, like the guestroom's light, to other persons. With such a simple ACL, we are able to grant that access to almost everyone who has a valid SIP account at any provider.

A problem occurs when we look at sensor states. Everyone in the system is able to see anybody's presence state by default. This might be problematic, because everybody would be able to monitor any house if they know their SIP address.

C. Third Layer

The problem explained above can be avoided by using encryption of the sensor data. To do so, the encryption key can be requested from the Smart Home before decrypting the presence information. This is achieved by using the technique of the second layer, where only users listed in the ACL are allowed to communicate with specific actors at the Smart Home. The end user's application sends a message, requesting the encryption key, to the Smart Home. After that, the Smart Home sends back the key, also as a simple message. The whole transfer is encrypted by using the first layer (SIPS). At this point, one can see the need of such a layer-concept, because without using the other layers below, the security of the third layer is useless. For example, without layer one, an attacker is able to gain the encryption key by simply listening or without the third layer, an attacker can simply request that key and will not be rejected by the Smart Home.

VIII. Discussion

To discuss this concept, we take a deeper look at the compatibility to other products and the performance of the Raspberry Pi. With this procedure, we show the result of this research. To do so, we discuss the following aspects:

1) Power consumption
2) Total costs

A. Compatibility

By using standardized software, we achieve a high level of compatibility with other products. Only the bridge component between KNX-Bus and the SIP Service had to be implemented, so it just uses the new protocol. A comparably small effort is necessary to connect different devices to each other by only creating the interfaces. Furthermore, our protocol does not influence other services, which are using a presence-based exchange of status information, because the enhancement is based on the standardized PIDF-Scheme.

Also, the three security layers are designed on top of the normal protocol in order to not influence other services. In comparison to existing Smart Home communication solutions, which are using a standard VPN connection, the new security design achieves additional security specifications to control the access to selected actors and sensors. A VPN connection grants access to the whole Smart Home network and thereby to all sensors and actor without any policing.

Through these achievements, this concept offers a stable foundation to develop home automation systems or further software with higher complexity, e.g., monitoring or control over a Smart Grid.

B. Usage of Raspberry Pi

In this concept, communication is not the only focus. The second important aspect is the usage of resource-poor implementation. Because of this, we selected the Raspberry Pi as our hardware platform. It is especially characterized by its low power consumption and its low costs. With that solution, we save a lot of space in comparison to a normal desktop computer.

1) Power consumption: The Raspberry Pi consumes around 750 mA at 5 V, resulting in 3.75 W per hour. At a workload of about 100 %, it consumes up to 1 A, which corresponds to 5 W [9]. This heavy workload was never measured during our tests. This brings us to an average consumption of about 3 W per hour, which results in a total power consumption of 17 kW hours a year.

2) Costs: The costs of power consumption amounts to about 5 EUR a year, referring to the German electricity prices of 2012 [11], which is very low in comparison to a normal destktop computer with a power consumption of about 100 W per hour. Furthermore, the cost of purchase is very small as well. The Raspberry Pi only costs around 60 EUR with all needed peripheral equipment [9]. That leads us to a unique cost of 60 EUR and permanent costs of about 5 EUR a year. In case of a defective device, there is mostly only the need of replacing the Raspberry Pi and to replug the SD-Card, which holds the whole software and configurations. So even in the case of a faulty device, the cost is minimal.

C. Usage of FreeSwitch

The preparation of external SIP server usage has been chosen because of existing infrastructure. Our current setup is a development environment of an equivalent one of a SIP provider. With such an environment, we are able to simulate all possible situations in a high scale network, so we can evaluate this concept. To do so, we analyzed selected performance data in the following subsection.

 ISBN: 978-1-61208-353-7

1) Power consumption: Currently, the FreeSwitch is running besides the KNX-to-SIP Bridge on the Raspberry Pi. In the future, the SIP-Server of an existing VoIP-Provider will be used, so our power consumption is effectively zero. That is because we do not need an own device, as we are using an existing infrastructure.

2) Costs: Like the power consumption, our cost is nearly zero. Nearly because we need at least one valid user account. Normally, a lot of end users still have an existing user account for a VoIP-Provider, because their telephone already runs the SIP-Protocol through VoIP.

IX. Conclusion

In this paper, an updated communication and security design for the Smart Home project at University of Applied Sciences Darmstadt was presented. It is shown that only one SIP profile can store the information of all actors and sensors within a Smart Home, instead of using separate profiles. Also, it is shown that the Raspberry Pi can be used for a home access gateway as an embedded system solution. With the implementation of the Raspberry Pi the general performance could be improved as well as the energy-consumption could be reduced, compared to a standard desktop computer. This approach makes use of well-known and open source information technology standards, instead of developing new commands for SIP or any proprietary application. This ensures future compatibility and makes the approach adoptable to other home automation systems.

X. Future Work

In future work, one step would be to extend the client to comply with the new communication concept or to develop a new client, which is platform independent, so it could be used with every mobile platform. Therefore, the aim should be a web-based client using WebSocket technology.

A. Cooperation with SIP-Providers

The communication concept detailed in this paper is also extendable to fit future needs. Furthermore, sensor information may be evaluated by smart grid- or weather stations. This can be done by splitting the sensor and actor data into several groups, which then needs the cooperation with the SIP-Providers to get access to a bundle of valid SIP-Accounts to provide access to each group through these accounts. With such an extension, the network load will be lowered and the communication may be differentiated in a more efficient way.

B. Security

A disadvantage of this approach is that the server is able to read the transferred encryption key. To solve this problem, there is an ability to generate a temporary encryption key by using a Diffie-Hellman [12] key exchange and encrypt the original encryption key with that temporary one.

Another disadvantage is, that all users who are capable of calling for the password are able to see all the data that is transferred. A solution for this is to define groups of sensors and actors, as well as different passwords for each group. With such a proceeding, a permitted user is only allowed to read the data, which is required for him. An additional need is to renew all passwords inside a timeframe, so all revocations of monitoring rights are successfully deployed.

These three modifications should make up layer four of our communication architecture in the next step of our research. Another possibility is to implement a public key infrastructure to control the access to sensors and actors. This could replace layer two to four in one step.

References

[1] KNX Association, Standardisation, http://www.knx.org/knx-en/knx/technology/standardisation, [retrieved: Apr. 2014]

[2] Raspberry Pi Foundation. Raspberry Pi, http://www.raspberrypi.org/, [retrieved: June 2014]

[3] J. Rosenberg, et al., SIP: session initiation protocol, IETF, RFC 3261, Jun. 2002.

[4] D. J. Cook et al., MavHome: an agent-based smart home, Proceedings Of The First IEEE International Conference On Pervasive Computing And Communications, pp. 521-524, March 2003.

[5] H. Schulzrinne, X. Wu, S. Sidiroglou, and S. Berger, Ubiquitous Computing in Home Networks, IEEE Communications Magazine, pp. 128-135, Nov. 2003.

[6] S. Moyer, D. Marples, S. Tsang, J, Katz, P. Gurung, T.Cheng, et al., Framework Draft for Networked Appliances using the Session Initiation Protocol. IETF Internet Draft, May 2001.

[7] R. Acker, S. Brandt, N. Buchmann, T. Fugmann, and M. Massoth, Ubiquitous Home Control based on SIP and Presence Service. Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, pp. 759-762, Nov. 2010.

[8] B. Bertran, C. Consel, P. Kadionik and B. Lamer, A SIP-basedhome automation platform: an experimental study, Proceedings of the 13th International Conference on Intelligence in Next Generation Networks, pp. 1-6, Oct. 2009.

[9] Raspberry Pi Foundation, FAQs, http://www.raspberrypi.org/help/faqs, [retrieved: Apr. 2014]

[10] H. Zimmermann, IEEE Transactions on Communications, vol. 28, no. 4, pp. 425432, Apr. 1980.

[11] Eurostat. Electricity prices for domestic consumers from 2007 onwards, http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_pc_204, [retrieved: Apr. 2014]

[12] IETF. RFC 2631: Diffie-Hellman Key Agreement Method, http://tools.ietf.org/html/rfc2631, [retrieved: June 2014]

 ISBN: 978-1-61208-353-7