A Home Manager Application for ZigBee Smart Home Networks

Fabio Luigi Bellifemine\textsuperscript{1}, Claudio Borean\textsuperscript{1}, Gianluca Dini\textsuperscript{2}, Pericle Perazzo, and Marco Tiloca\textsuperscript{2}

\textsuperscript{1} Telecom Italia SpA
\{fabioluigi.bellifemine, claudio.borean\}@telecomitalia.it
\textsuperscript{2} Dipartimento di Ingegneria dell’Informazione
University of Pisa - Pisa, Italy
\{g.dini, m.tiloca\}@iet.unipi.it

Abstract. In this paper we present the main ideas behind the Home Manager, a software infrastructure aimed at a usable and secure management of Cooperating Objects for Home Automation applications. Through the Home Manager, a user can add and remove devices, query and set them, and define device access control policies. The Home Manager encrypts and decrypts commands to and data from devices, distributes keying material to new devices, and rekeys the network upon device leaving. An early prototype of the Home Manager has been implemented.

Key words: Home Manager; Home Automation; Cooperating Objects; ZigBee; Security; Privacy;

1 Introduction

Nowadays, applications based on Wireless Sensor Networks (WSNs), embedded systems and pervasive computing are asserting themselves as particularly attractive, both for academia and industry. This has led to identification and definition of a new paradigm, known as a network of Cooperating Objects, within which single entities communicate with each other and aim to a common goal [1]. This innovative model can be fruitfully employed in several different scenarios, among which Home Automation and E-Health seem to be particularly noteworthy.

In this paper we consider a software infrastructure for Home Automation, where a ZigBee network of objects cooperate to monitor and control energy consumption in a home. Cooperating objects are smart plugs, which have both sensing and actuating capabilities, and different types of smart devices, i.e., actuators ranging from purely on-off devices, e.g., a TV-set, to devices offering some form of QoS, e.g., a air-conditioner. The infrastructure is aimed at applications that monitor and control behaviours of devices including collecting their power consumptions from smart plugs, turning devices on and off and/or changing the level of service they provide.
The software infrastructure has three main objectives, namely usability, security, and privacy. In terms of usability, a non-expert user must be able to manage the network, introducing new devices, binding together smart plugs and devices and so forth. In terms of security, the infrastructure must protect the authenticity and integrity of sensed data and control commands travelling on the network from an external attacker. Furthermore, as the network conveys user’s personal data, confidentiality of such data must be protected in order to protect the user privacy.

We achieve these requirements by means of the Home Manager, an application gateway that i) simplifies network management by providing an intuitive graphical user interface; ii) supports security by implementing device commissioning, device authentication, and key management [2], and, finally, iii) supports privacy by allowing a user to define a privacy policy in terms of classes of users, where a class specifies a set of access rights on a set of objects. A user does not directly interact with the network but through the Home Manager which grants or denies operations according to the privacy policy. During the execution of an operation, the Home Manager encrypts and decrypts commands and sensed data as needed. Finally, the Home Manager is responsible for distributing keying material to new devices and rekeying the network upon device leaving.

During these last years, several contributions have been provided regarding these topics, with particular attention to privacy preservation in Wireless Sensor Networks (WSNs) [3], [4], [5] as well as in context-aware environments, including Home Automation scenarios [6], [7], [8], [9], [10]. As widely discussed, WSNs aggravate the privacy problem and make it particularly difficult to be solved by means of standard techniques. In addition, it is very important to disclose sensitive data only if strictly necessary, as well as properly take into account preferences and perceptions of the end users, who still represent the weakest part of the whole protection chain.

The rest of the paper is organized as follows. Section 2 briefly describes the architecture of the Home Automation system whereas Section 3 provides an overview of the Home Manager in terms of provided operations and user interface. Section 4 discusses security and privacy issues. Section 5 briefly describes the Gateway Abstraction Layer (GAL), a Telecom Italia’s middleware layer that virtualizes the ZigBee network as a collection of Web-resources. Section 6 describes an early prototype of the Home Manager implementing the functional features. Finally, Section 7 reports our conclusions.

2 System architecture

Home Automation (or Domotics) is the practice of enhancing the automation and the cooperation between household appliances. A Home Automation service is a set of employed technologies that supplies Home Automation in a house.

In order to provide Home Automation services, a hub-and-spoke structure can be used, with a centralized manager entity and the cooperating objects as the spokes. The network that includes the manager and the devices inside the house
is called home network. In particular, the manager makes it possible to control the devices inside the home network. Control on devices is possible by means of operations, configurations or queries performed by the manager, either after a user’s order or automatically, according to a particular managing policy. The manager entity can be either a software application, installed on a computer, or a processing hardware. A prototype of such application, called Home Manager, has been developed. Home appliances are also called smart devices, they can be battery-powered and include smart plugs, smart refrigerators, smart HVAC devices, and lighting and closure devices, like doors, windows or curtains. In particular, smart plugs are electric outlets with communication capabilities, which are able to provide power and energy consumption reports, and perform on/off operations.

The Home Manager and the smart devices communicate over a ZigBee network, allowing to address devices placement and management in an easier and more flexible way. ZigBee [11] is a protocol suite for wireless communications, suitable for low-power consuming devices and home automation scenarios. It provides a stack of protocols designed to work with IEEE 802.15.4 [12], which deals with network management issues as well as application layer’s messages. ZigBee asserts itself as a good choice, being low-power and low-cost oriented, so making it possible to increase devices’ batteries life and reduce installation and maintenance costs. ZigBee specification defines a standard set of Clusters [13], which are collections of applicative messages (Commands) used to interact and communicate with smart devices. Besides, ZigBee Profiles define sets of Clusters and general rules to which refer to for particular application scenarios. The Home Manager refers to ZigBee Home Automation application profile [14], which specifies the ZigBee clusters that are likely to be used in a domotic application.

Fig. 1 provides an overall view of the system architecture. In particular, we suppose to have a special device named Network Coordinator, which is able to act as Coordinator in a ZigBee network and within which both the Home Manager application and the Gateway Abstraction Layer (GAL) middleware have been installed. As depicted in the picture, users do not directly interact with the smart devices, but they are supposed to establish an application level session with the Home Manager application. It is worth noting that the communication between the end users and the Home Manager can be realized by other technologies totally different from ZigBee, such as Ethernet [15], IEEE 802.11 [16] or Bluetooth [17]. Moreover, the Home Manager provides the users with control and monitoring functionalities as described in Section 3, and makes use of the ZigBee stack by means of the GAL middleware discussed in Section 5. Finally, communications can be established between the Network Coordinator and the ZigBee smart appliances, thanks to the ZigBee protocol stack.

3 Home Manager

The Home Manager is supposed to perform some important operations, such as detecting new smart devices installed in the house (detection), identifying
Fig. 1. System architecture.

a single smart device among a list of similar detected objects (identification), placing it on a house map (map placing), binding related devices with each other (binding) and manually or automatically controlling the devices (control). Each of the above operations is briefly explained in the following sections.

3.1 Detection

The detection process is performed by the ZigBee protocol itself. The Manager can perform a search for new smart devices either periodically or upon user request. The devices can also autonomously advertise their presence to the Manager, either after they have been installed or when they detect the presence of a new home network. Furthermore, the Manager can retrieve the type of the newly detected devices, such as smart plug, smart refrigerator.

3.2 Identification

Although it is possible to detect new smart devices inside the home network, their physical position cannot be determined. Specifically, the Home Manager is able to show only a list of detected devices specifying their type, with no information about their placement. However, the user can identify a single device among different detected devices of the same type, by means of a proper possibly out-of-band identification procedure, such as physical signals perceptible by humans,
like a sound signal or a blinking light, produced by the device. By interacting with the Manager, the user can manually ask a single device to reveal its identity, and then associate a unique information, such as a name or a label, to the same device.

3.3 Map placing

Assigning a unique identification name to each device can be complicated for the user, especially for devices like smart plugs, which can be in a significant number even inside a single house. An alternative procedure consists in a map placing. The user, once identified a device, can place it on a map representing the smart house, which can be managed and displayed by means of the Home Manager software. The user can place a device on the map by the Manager’s graphic user interface, thanks to a drag-and-drop operation and a common pointing device. The house map can be stored permanently on a file, such as in an XML-based format.

Fig. 2 shows a screenshot of the Home Manager application. The small circles represent lamps, yellow when turned on, gray otherwise. The white rectangles depict radiators, while the gray one in the top left room is a refrigerator and the light green circle near it is a smart plug. Devices shown in the map are called placed. Instead, some unplaced smart plugs are shown in the white area on the right side of the window. More specifically, a device is unplaced if the Manager has detected its presence but the user has not specified its position yet. With reference to Fig. 2, the refrigerator device has been selected, and the Device info box reports information concerning it, such as its content with the expiration date of each product, the current state of the door (closed/open) and the internal temperature graph over time.

3.4 Binding

Some devices are logically or physically related with one another. Possible examples are a smart plug powering another smart device, or presence detectors that need to activate an alarm. The automation level could be enhanced if the manager entity knew about such relationships between smart devices. For example, if the Manager knew which smart plug powers a particular device, it could specifically monitor the device’s power consumption and perform powersaving policies (some of which are described in the next section). By means of the provided graphical user interface, the user can also inform the Manager about relationships among devices, by simply dragging a device’s symbol on the related device’s symbol. An example is provided in Fig. 2, where a light green arc represents the physical binding between the refrigerator and the smart plug that powers it. Being aware of such a binding, the Manager is able to show the refrigerator’s power consumption over time, thanks to the information it periodically receives from the smart plug. However, the binding between devices performed by the end user is a suitable solution in presence of appliances which feature a very simple control interface, or are not supposed to be moved so often.
Otherwise, in case a smart device were capable of displaying some information and provided a simple keypad, it would be possible to bind it right away with another appliance, without interacting with the Home Manager application.

3.5 Control

According to their type, the Manager must be capable of performing different operations on the devices. For instance, it can ask a device to turn off or on; retrieve the measured temperature (setpoint) if the device is equipped with a thermostat (e.g. a refrigerator or a heating device); order a set of devices to adopt a configuration scenario, including closed windows, closed curtains, locked doors and lights off for a not-at-home scenario, or garden lights on and heating on for an evening scenario. Plus, the Manager can apply automatic power-saving policies. That is, if the user contracts a power consumption limited agreement with an electric power company, and the Manager detects that a smart plug powered device is consuming more than what provided by the contract, it can take proper countermeasures according to the device’s type. Specifically, if a device can only be turned off (e.g. a lighting device) the Manager can turn it off, while if the device can be lowered in intensity (e.g. an electric heating device) the Manager can properly adjust its consumption level.

Fig. 3 and 4 show the control panels for the refrigerator and the heating device, respectively. With regard to the former, the user is able to see the refrigerator content with the expiration dates as well as the state of the door (closed/open). Plus, it is possible to control the desired temperature and read
the present internal temperature. Finally, the Engine box shows whether the heat pump is currently working or not, while the button Failure makes it possible to simulate a generic failure. Similar information and controls are provided by the heating device’s panel, especially related to temperature management.

![Fig. 3. Refrigerator control panel.](image1)

![Fig. 4. Heating control panel.](image2)

4 Security and privacy

As well as assuring that home appliances are simple to manage and correctly work, it is wise to take care about information management within the network. In fact, in order to effectively provide the end user with the services he needs and requests, smart devices necessarily have to access potential sensitive data and exchange them with one another. Besides, such data may involve user personal information, which should be revealed or exposed with extreme caution. Obviously, both data confidentiality and integrity must be guaranteed within the home network, by protecting communication with smart devices as well as the interaction between the user and the smart environment altogether. Although communication among network nodes raises typical security problems, which can be faced by means of proven techniques (i.e. cryptography, digital signature, use of nonces), the involvement of personal information immediately leads to a privacy issue, since confidentiality of user sensitive data is put under risk. Speaking of which, we deal with privacy at two different levels, in accordance with the referential system architecture described in Section 2 and shown in Fig. 1. In particular, we have to take care of both the network layer communication between the Network Coordinator and the home devices, as well as the interaction between the end user and the Home Manager application.
4.1 ZigBee network security

As to the smart devices communication, data confidentiality and authenticity can be effectively assured by means of secure communication solutions offered by ZigBee. In the previous sections of this work, we referred to a ZigBee network and in particular to the ZigBee Home Automation profile. As extensively described in [11], ZigBee provides specific security services aimed at protecting devices communication in an end-to-end fashion. We assume that the Network Coordinator acts as Trust Center and is responsible for controlling network admittance policies, as well as handling key management and other security services. Key material consists in a common Network Key shared among all devices, and a pairwise Link Key sharable between any two devices.

However, since Home Automation has been designed for low-security residential applications, key material must be either pre-installed or obtained unsecurely before joining the network, although the Network Key delivery can be protected by means of a pre-installed Link Key. That being said, it is evident that Home Automation profile provides just the very essential security practices to secure network communication. In contrast, although it is designed for different and more specific employments (i.e. power consumption monitoring and pricing), ZigBee Smart Energy application profile [18] significantly enhances Home Automation security features and provides also a certificate-based Key Establishment procedure between the smart devices and the Trust Center. Therefore, we claim that Smart Energy profile should be used in conjunction with Home Automation profile, so allowing smart devices to join the network and communicate with one another in a more secure and reliable way. However, as observed in [2], Smart Energy profile still shows some deficiencies concerning devices authentication during commissioning and key material renewal. In particular, we think it is necessary to address new devices’ authentication by means of a simple home certification practice, as well as introduce a new proper ZigBee Rekeying Cluster in order to cope with compromised or leaving devices.

4.2 Users’ privacy preservation

As to user interaction with the home network, it is worth recalling that a user does not directly communicate with the smart devices, but through the Home Manager (see Section 2). The communication between the end user and the Manager application, which certainly has to be protected as well by SSL [19], TLS [20], or WTLS [21], can take place by means of hand-held devices and totally different communication technologies, such as Ethernet [15], IEEE 802.11 [16] or Bluetooth [17]. Therefore, the interaction between the Home Manager and the user is totally independent from the specific kind of home network that lies below.

Having said that, since several different users might interact with the home network, including guests and other non-permanent users, it is clear that they might have access to sensitive information, so putting other users’ privacy at serious risk. In order to address this issue, we take into consideration what suggested in [22] and propose to:
1. Consider each smart device as an object capable of providing a set of operations.
2. Define object classes as sets of permitted operations on a given set of devices. This should be done in accordance with the minimum privilege level required to perform such operations on the considered devices.
3. Classify users in different user classes, according to their supposed privilege level. Possible classes are Administrator, Friend, Child and Guest.
4. Bind each user class with one object class. For Guest users, each single action should be explicitly authorized.

What has been described above makes it possible to determine what each user will be able to do within the home network and what he is authorized to gain knowledge of. As a practical case, we can think about a typical family with little kids. Husband and wife are supposed to do whatever they want with their home network, thus they can interact with the Home Manager as Administrator users and deal also with devices’ placing and binding operations we described in Section 3. In contrast, Friend class is useful for temporary users, which are supposed to have a limited interaction with a reduced number of devices. As an example, they could be authorized to control a Hi-Fi system, but not to turn on and off an oven or open the home safe. Besides, homeowner’s kids should be classified as Child users, so that they are forbidden from doing something dangerous or unfitting for their age. Similarly, in case the smart fridge stopped working and a technician has been called, it would be better to treat him as a Guest user, assuring that he is able to interact only with the fridge he is supposed to fix.

More specifically, before a new user can interact with the Home Manager, the home network administrator, that is the homeowner, must create a new user account on the Network Coordinator, having care to indicate the specific type of user who asks for network access. Similarly, the administrator will have to delete an account previously created, once the corresponding user leaves or is supposed to leave the network. Besides, each user class can be associated with a different account expiration time, in order to automatically remove a user account once the user has left the network. According to our model, the Home Manager provides the homeowner with the sign up panel shown in Fig. 5, so making it possible to simply manage new user account creation. In particular, user ID, password and user type must be provided while creating a new account, and a picture can be added to the new user profile as well. Once his account has been created, the new user is able to perform a login to the Home Manager application and interact with the home network. However, such an interaction will be constantly supervised by the Home Manager on the Network Coordinator, which will evaluate each request the user makes to the network. Thus, in case the user tried to do something without being authorized, the Home Manager does not process the request and reports an error.

Thanks to what has been described above, the end user is capable to determine who is supposed to control his appliances and access personal data, without forgetting that sensitive information should be revealed only if strictly
necessary. Thus, he has control of accesses and behaviours of other users, potentially different with regard to their privilege level, with particular attention to those actions which leads to personal information disclosure. Besides, it is worth noting that the end user is the only one who really knows which actions should be forbidden to which particular users and from whom personal data are supposed to be hidden. Therefore, he definitely cannot avoid to personally determine which data are to be considered confidential, as well as what each specific user should be allowed to do within the home network. Finally, we claim that allowing users to simply manage security and privacy preservation within their smart home, it will be possible to make them more confident and trustful about Home Automation technology as a whole, encouraging its adoption and further promoting its diffusion.

5 Gateway Abstraction Layer

A relevant role for the upcoming ZigBee services is getting covered by the gateways, units able to interconnect the Wireless Sensor Network (WSN) world to IP based backend systems in order to manage and control the end user applications. The object of this section is to present an abstraction layer for a ZigBee Gateway (Gateway Abstraction Layer or \textit{GAL}) that enables fast development of different applications (e.g. elderly monitoring, energy management) using a simple API. By using the GAL an application relying on a gateway could effectively run a discovery of the nodes in the networks, retrieve the list of available services and set up the communication with a specific device. In particular, the main idea behind the GAL is to enable the development of both local applications, installed on-board mobile or fixed terminals, and remote applications running on different hosts. The resulting middleware allows flexible and simultaneous access of multiple (local and remote) applications to the ZigBee Gateway: in this scenario, network management could be performed by a remote platform, while a local application could run directly on the terminal itself.
5.1 GAL system Architecture

The reference architecture for the development of the GAL is described in Fig. 6. The nodes participating in a ZigBee mesh network, as widely known by people developing applications with wireless sensor network technologies, work in an ad-hoc configuration independently from a centralized point that can be connected to a distribution network (e.g., IP network).

Fig. 6. GAL system architecture.

The use of a proper gateway, able to connect to the ZigBee network and capable of configuring the devices in order to get and set the proper frame into the network and run the required tasks, becomes then a key element for the overall architecture and an effective deployment of the application. Another key element for the gateway device is the independency from the specific application running in the ZigBee network itself: a general purpose gateway exposing a generic interface, enables the usage of a predefined layer for the interconnection towards local and remote applications. The resulting abstraction layer does not need to be changed according to the application itself. Moreover, a relevant requirement for an application using the GAL is the possibility to be resident on the gateway itself or remote on a server, that is beyond the distribution network of a Telco operator. Nevertheless, the Telco operator could build the application itself on the top of a WSN management platform. The local application could
be very effective whenever a Gateway with sufficient resources is used, because it can provide the end user with a simple GUI and immediate feedbacks about the application. In the reference architecture, both local applications and the WSN platform access the functionalities exposed by the GAL by using a REST stub which acts as a library and enables the usage of the features implemented by the GAL core.

5.2 GAL core functions

The logical architecture of the GAL considered in this section is described in Fig. 7 and is basically made of four main building blocks.

![GAL Core building blocks](image)

The **Low-layer interface** provides the low level interface between the ZigBee Gateway and the so-called ZigBee Network processor, i.e. a device that integrates an IEEE 802.15.4 compliant radio transceiver and has built-in ZigBee networking capabilities (i.e. the protocol stack). The **GAL Core** logically groups together different APS-level ZigBee commands to define high level macro-functions and exposes them to the RPC layer by means of the GAL interface. In particular, GAL core is organized into four agents, i.e. TrustCenter Agent, Discovery Agent, Management Agent and Service Agent: these are software modules implementing separate logical functions and actually providing the Gateway state machine. The **GAL** represents the real abstraction layer and defines the programming interface used to expose GAL core functionalities to higher software layers. Finally, the **RPC layer** allows local or remote applications to invoke the macro-functions.
exposed by the GAL interface. To this aim, different technologies can be used, e.g. REST [23] or SOAP [24].

6 Implementation

It is likely for the Manager to run on a device provided with a display and rarely turned off, such as a home Internet gateway or an always-on personal computer. Such a device can also be equipped with a permanent Internet connection, which makes it possible to control and configure the Manager software even by a remote access.

The Home Manager application has to be cross-platform, in order to be installed on different machines and different operative systems. Speaking of which, the prototype has been implemented by using a .NET framework [25] in C# language [26] and a .NET virtual machine for execution and testing. That is, the Home Manager can run on a Windows system and, by means of a Mono framework [27], on a Linux or Macintosh system, with no recompilation needed. In order to virtualize the ZigBee Applicative Protocol, a .NET library called ZigApi has been developed as well. Communications actually take place among Telegesis ETRX2 USB sticks [28], which implement ZigBee application and network layers as well as the IEEE 802.15.4 MAC and physical layers. Plus, in order to virtualize the communication with the USB sticks, a server-like program called Gateway Abstraction Layer (GAL) has been used. This program was discussed in detail in Section 5.

For tests and demonstrations, we used physical smart plugs devices provided by Telecom Italia, as well as virtual smart plugs, virtual refrigerators and virtual heating devices. The virtual devices have been implemented as programs which run on a personal computer and access the ZigBee network by means of a ETRX2 USB stick through GAL. These applications emulate all device’s capabilities, so being able to exchange messages, react to the Manager’s commands and reply to its queries, and reproduce physical phenomenons like temperature or voltage variations, as well as trigger events like hardware faults.

Some devices use manufacturer-specific ZigBee clusters, which have not been included in the standards so far. The Home Manager is able to communicate with devices that implement the standard ZigBee clusters, but can be also extended in order to handle with manufactory-defined clusters as well. With software plug-ins (e.g. installable .NET dynamic libraries), possibly provided by smart devices’ manufacturers, the Home Manager could be able to deal also with manufacturer-specific ZigBee clusters, manufacturer-specific managing policies, and/or GUI extensions as additional menu entries, which allow the user to manage non-standard smart devices.

Finally, a version of the Manager without a user interface can be implemented, in case a display were not available on the hardware on which the Home Manager runs. Then, a proper front-end program, running on an hardware equipped with a display, can be used to control and configure the Manager.
7 Conclusion

In this paper, we have presented the Home Manager, a software application for Home Automation in ZigBee networks. Through the Home Manager a user can interact with his/her home devices and monitor and control energy consumption in his/her home. The application provides users with simple management operations (e.g. new devices installation), and copes with data security and privacy within the home network.

In particular, the Home Manager features a simple and intuitive graphical user interface for management operations, takes care of communication security by means of devices authentication and key management, and finally allows the user to define personal privacy policies.

So far, an early prototype of the Home Manager has been implemented. Future works will consist in improving our implementation, with particular attention to new device type support, security features enhancement, and extensions regarding access control and users permissions within the network.

References

12. Institute of Electrical and Electronics Engineers, Inc. New York: IEEE Std. 802.15.4-2006, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). (September 2006)