2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing

Considerations on Security in ZigBee Networks

Gianluca Dini and Marco Tiloca
Dipartimento di Ingegneria dell'Informazione
University of Pisa
Pisa, Italy
{*gianluca.dini, marco.tiloca*}*@iet.unipi.it*

***Abstract*—ZigBee outlines a new suite of protocols targeted at low-rate, low-power devices and sensor nodes. ZigBee Specification includes a number of security provisions and options. The security model specified in the Smart Energy Profile seems bound to become the reference security model for most of ZigBee applications. In this paper we review this security model and highlight places where its specification presents concerns and possible inefficiencies in security management. Specifically, we show that the specification does not adequately address the forward security requirement so allowing a number of threats at the routing and application layer. Furthermore, we show inefficiencies in managing both the Network Key and devices certificates. Finally, we make some proposals to address these problems.**

***Keywords*-ZigBee; security; key management;**

I. Introduction

ZigBee is an emerging standard for low-power, low-rate wireless communication which aims at interoperability and encompasses a full range of devices even including low-end battery-powered sensor nodes. ZigBee is built upon the physical layer and medium access control defined in the IEEE 802.15.4 standard (2003 version).

ZigBee Specification includes a number of security provisions and options. In particular, ZigBee provides facilities for carrying out secure communications, protecting establishment and transport of cryptographic keys, cyphering frames and controlling devices. ZigBee improves the basic security framework defined in IEEE 802.15.4, focusing also on key establishment and distribution.

ZigBee Specification provides two security models, Standard Security Mode and High Security Mode. While the former is designed for lower security residential applications, the latter is intended to be used for high security commercial applications. The security model of the Smart Energy Profile is asserting itself as a reference security model for ZigBee applications, since it constitutes a trade-off between the two standard modes.

In this paper we first introduce this security model and then we show that it presents two critical issues that have not been adequately addressed. First of all, the security model does not adequately address the *forward security* requirement [1]. Actually, upon leaving the network (or being forced to), a node still remains able to access communication because the onboard keying material is not properly revoked. A node may leave the network when it is dismissed, sent to maintenance, lost, compromised, or supposed so. In all these cases, the keys stored on the device may be compromised, and thus, if the they are not properly revoked, an adversary may exploit them to mount severe attacks against the network and application level.

Second, the model comprises a public-key protocol for device authentication and key establishment. In order to be open and interoperable, the model allows many subjects to issue certificates, namely manufacturers, distributors, and even end users. As a consequence, a device should be equipped with certificates of all potential certification subjects. However, this requirement raises a scalability problem, since it conflicts against the limited storage resources of ZigBee end devices.

The rest of the paper is organized as follows. In Section II we provide an overview of ZigBee and IEEE 802.15.4. In Section III we discuss the security model and the related key management mechanisms provided by the ZigBee Smart Energy Profile. In Section IV we present the security concerns regarding the forward security requirement and certificate management and propose possible approaches for solutions. Related works are discussed in Section V. Finally, in Section VI we draw our conclusive remarks.

II. Overview

ZigBee is a specification for a suite of high level communication protocols, intended for devices equipped with small and low-power digital radios based on the IEEE 802.15.4 standard. As reported in [2], ZigBee and IEEE 802.15.4 are standards-based protocols which provide the network infrastructure required for wireless sensor network applications. As depicted in Fig. 1, IEEE 802.15.4 defines the physical and MAC layers, while ZigBee defines the network and application layers.

The IEEE 802.15.4 MAC layer provides reliable communications between a node and its immediate neighbors, addressing collision avoidance and improving efficiency. The MAC layer also assembles and decomposes data packets and frames, while the physical layer provides the interface to the physical transmission medium (e.g. radio).

ZigBee places itself on top of the IEEE 802.15.4 PHY and MAC layers. Basically, it is formed by the application (APL)

978-0-7695-4049-8/10 $26.00 © 2010 IEEE
DOI 10.1109/SUTC.2010.15

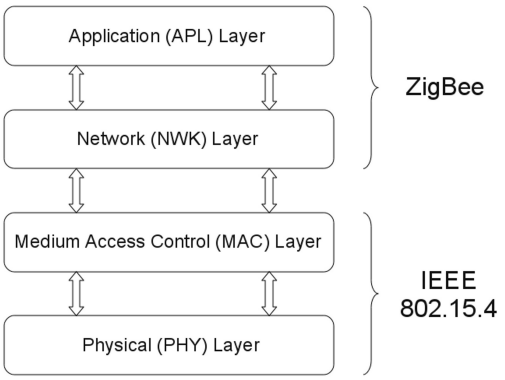

Figure 1. The ZigBee protocol stack.

layer and the network (NWK) layer. Among other things, the application layer specifies frame formats for transporting data and provides a data service to the applications, while the network layer handles network management and routing by invoking actions in the MAC layer. Security is provided in a cross-layered fashion, involving both the application and the network layer.

In the rest of this section we present the main features of IEEE 802.15.4 and ZigBee networks with more details.

A. IEEE 802.15.4

An IEEE 802.15.4 network can be composed of two types of device: Full-Function Devices (FFDs) and Reduced-Function Devices (RFDs) [3]. FFDs can talk with both other FFDs and RFDs, whereas RFDs can talk only with an FFD. Typically, an RFD is intended for very simple applications and features minimal resources, in terms of storage, memory and processing capability. An IEEE 802.15.4 network comprises at least one *Coordinator*, i.e. an FFD capable of relaying messages from other devices. Furthermore, one Coordinator is elected as *PAN Coordinator* and is responsible for network and security management. An RFD is associated to a single FFD at a time. In an IEEE 802.15.4 network two topologies are allowed: *Star* and *Peer-to-peer*. In the Star topology each RFD talks directly to the PAN Coordinator, while in the Peer-to-peer topology each device can communicate with any other FFD in its range, in order to define more complex network scenarios.

IEEE 802.15.4 provides security services on incoming and outgoing frames, when requested by the higher layers. The standard supports the following security services on a per-frame basis: data confidentiality, data authenticity, and replay protection. Such services are provided at the MAC level by means of (i) an *auxiliary security subheader* (carrying useful information for security processing, including how the frame is actually protected and which keying material is used); (ii) a collection of *security MAC layer attributes* (in order to configure security procedures in a flexible way and determine how to provide security), and, finally, (iii) *Frame security procedures* (i.e. operations to secure/unsecure frames or retrieve cryptographic keys). An important point to consider is that IEEE 802.15.4 does not consider at all key establishment and device authentication. The MAC layer entrusts such services to the higher layers (e.g. ZigBee), and solely provides communication security at the MAC level. Thus, IEEE 802.15.4 assumes that when a frame is transmitted (received) and it needs to be secured (unsecured), all the needed security material (i.e. cryptographic keys) is available and already established at both the sender and the recipient side. However, IEEE 802.15.4 recommends to use the 128-bit AES encryption scheme [4].

B. ZigBee

The ZigBee Alliance has developed a two-way wireless communications standard, which turns out to be low-cost and low-power consumption. Solutions adopting the ZigBee standard will be embedded in consumer electronics, home and building automation, industrial controls, PC peripherals, medical sensor applications, toys and games.

According to the Specification [5], a ZigBee network may comprise three types of devices: *Coordinator*, *Router*, and *end device*. With reference to the device types in an IEEE 802.15.4 network, the ZigBee Coordinator corresponds to the PAN Coordinator, a Router corresponds to a Coordinator and an end device corresponds to an RFD or an FFD which is neither a Coordinator nor the PAN Coordinator. In the rest of this document we will use the ZigBee terminology to indicate devices.

The ZigBee network layer (NWK) supports *Star*, *Tree*, and *Mesh* topologies. In the Star topology, the network is controlled by the Coordinator, which is responsible for initiating and maintaining the devices on the network, while the end devices directly communicate with the Coordinator. In Mesh and Tree topologies, the Coordinator is responsible for starting the network and for choosing certain key network parameters, but the network may also be extended through the use of ZigBee Routers, while routes are established by means of a routing protocol similar to the Ad hoc On-demand Distance Vector (AODV) protocol. In Tree networks, Routers move data and control messages through the network using a hierarchical routing strategy.

With regard to security issues, ZigBee fits very well with the services provided by IEEE 802.15.4 and uses symmetric key encryption for end-to-end communications. In particular, a ZigBee network must comprise a *Trust Center*, a node, typically the ZigBee Coordinator, which provides key management and other security services.

ZigBee offers some *application profiles* which specify both a possible collection of devices and a set of messages used by devices to communicate with one another. Each application profile describes also some *clusters*, sets of parameters and commands (some mandatory) the devices

have to use in order to interoperate within the network. Nowadays, the most important and promising ZigBee application profiles seem to be *Home Automation* [6] and *Smart Energy* [7]. In the rest of this document we will focus on the latter, since it considers security as a major issue and includes precise mechanisms for secure communications as well as a proper *Key Establishment Cluster* [7].

III. Security in Smart Energy Profile

Smart Energy Profile (SEP) provides device descriptions and standard practices for demand-response and load management applications, tailored for residential or light commercial environment. Possible scenarios include single homes or even an entire apartment complex. Currently, key application domains are metering, pricing and demand response and load control applications. SEP specification provides standard interfaces and device definitions to allow interoperability among ZigBee devices produced by various manufacturers of electrical equipment, meters, and Smart Energy enabling products.

In order to become part of a ZigBee network, a new device has to pass through a process called *Commissioning*, which is known as the task of configuring devices and networks to achieve the needs of the specific installation [8]. The ZigBee Alliance has recognized the importance of Commissioning and, in particular, the importance of specifications for Commissioning in a multi-vendor environment. According to SEP, devices can form their own network or join an existing network. Having said that, there should be some indication to the user that the network has been formed properly in the former case, or that the device has joined the network successfully in the latter. The indication can be implemented in a number of ways, including blinking indicator lights, colored indicator lights, arrays of indicator lights, text displays, graphic displays, audible indicators such as buzzers and speakers, or through separate means.

The Commissioning process is critical from a security viewpoint, but it has to be simple from a user perspective, and capable to provide some sort of feedback. Besides, it is assumed to be accomplished by a trusted person. A Commissioning Cluster is currently under development [9]. However, from a very high level of abstraction, Commissioning consists of the following steps:

1) The network must be informed of the device that is to be joined. This operation is done through out-of-band means, which could include a web login, a phone call to a service center, or an interaction through an hand-held appliance. By means of this operation, the network is made aware of the device identifier (ID) and security information appropriate for the device.
2) The network is put into *permit joining ON* state.
3) The installer/homeowner is prompted to press a button or complete a menu sequence that tells the device to attempt to join the network.
4) The device attempts to join the network. In doing that, the device is authenticated using the appropriate security mechanisms and new keying material is distributed using the Key Establishment Cluster. In the rest of this document we will refer to this step as *Join procedure*.
5) An indicator is provided for the installer/homeowner indicating the device has joined the network and authenticated properly or provides information about improper authentication.
6) The device can now operate normally on the network.

Step 1 assures that the device is under the physical control of the installer/homeowner and has been explicitly authorized by him to join the ZigBee network. The Join procedure is the most complex step and it will be described in detail later in this section. Therefore, the Commissioning process seems to provide a pretty good robustness regarding the devices access to the network, thanks to the very first step and to the Key Establishment Cluster features.

In this section we take into account the main features regarding security services provided by ZigBee. In particular, the rest of this section will refer to the Smart Energy Profile and the Key Establishment Cluster.

A. Keys

SEP assumes three kinds of keys: the Link Key, the Network Key, and the Transport Key. A *Link Key* is an end-to-end key that a device may share with another device. However, a device must share a Link Key with the Trust Center (TC). This key is called the Trust Center Link Key (TCLK). The end device and the Trust Center establish the TCLK during the Join procedure (step 4 of the Commissioning process). The TCLK is used to protect application level messages and stack commands. Specifically, some of the clusters whose messages are protected with the Link Key are Time, Commissioning, Price, Demand Response and Load Control, Simple Metering, Message, Smart Energy Tunneling, and Pre-Payment. SEP allows the Trust Center to refresh the TCLK established with an end device, but suggests that it should be an infrequent operation.

A *Network key* is shared by all devices and it is used to protect management and control communications. It is worth observing that also application level data and commands can be protected by means of the Network Key, in case either a Link Key can not be retrieved or the network layer is explicitly requested to secure outgoing frames. Some of the clusters whose messages are protected with the Network Key are Basic, Identify, Alarms, Power Configuration and Key Establishment.

Finally, any given device shares a *Transport Key* with the Trust Center. This key is derived from the TCLK and its main use consists in securing the Network Key refresh process within the ZigBee network.

According to SEP, the Trust Center has to periodically refresh the Network Key. Such a rekeying is protected by

means of the Transport Keys. It follows that this rekeying is performed in a point-to-point way and thus its complexity in terms of rekeying messages amounts to $\mathcal{O}$(n), where n is the number of devices in the network. In order to prevent potential de-synchronization problems, the Trust Center can order to start using the new Network Key by sending a proper SWITCH_KEY command to all devices after the new Network Key has been distributed all over the network.

B. Device authentication

Now we will focus on the security mechanisms which take place during the secure Join procedure. Basically, as specified at step 4 of the Commissioning process, an end device has to authenticate itself and exchange security information items with the Trust Center after it has joined the network. Specifically, a device has to obtain the current Network Key from the Trust Center, and establish a new end-to-end Trust Center Link Key with it. The key establishment process consists of the following steps:

1) *Establishing the TCLK*. The Key Establishment Cluster specifies that each device i has a pre-installed Trust Center Link Key LK_i, typically obtained from the device Installation Code, or similar. LK_i is provided to the local Trust Center through out-of-band means. This operation could take place at step 1 of the Commissioning process, while informing the Trust Center about the device that is to be joined.
2) *Establishing the Transport Key*. The device i and the Trust Center can now obtain the Transport Key TK_i deriving it from LK_i.
3) *Distributing the Network Key*. As soon as the device i has joined the network, the Trust Center sends it the Network Key NK encrypted by means of TK_i.
4) *Establishing a new Link Key*. As soon as the Join procedure has been completed, the Trust Center must update the Trust Center Link Key LK_i of the joining device i as described below.

C. Key establishment

According to the Key Establishment Cluster, during the Join procedure the key establishment process should follow the Certificate-Based Key Establishment (CBKE) method, since it provides the most comprehensive form of key exchange among two nodes in the network. Every device holds a certificate issued by a trusted Certification Authority (CA). Through the certificate, it is possible to retrieve the device public key and other useful security information. The main reason that led to adopt the CBKE method is the need to safely identify a device, before it can start data communications.

The key establishment process between an initiator and a responder consists essentially in the following four steps:

1) Exchange Static and Ephemeral Data.
2) Generate Key Bitstream.
3) Derive Message Authentication Code (MAC) key and Key Data.
4) Confirm Key using MAC.

Regarding the second and third step, the key establishment procedure refers to the Elliptic Curve MQV key agreement scheme and a Key Derivation Function respectively, both described in [10]. At the end of this process, the Trust Center and the end device i share a new Link Key LK_i that is going to be used to protect data communications between them. Observe that the new Transport Key TK_i is obtained from the new Link Key LK_i just established.

Once a device i has joined and been authenticated via key establishment and obtained an authorized Link Key LK_i with the Trust Center, it may need to communicate with another device j on the network, using application layer encryption. Rather than using key establishment between them, it would be advantageous to leverage the Trust Center to broker trust with other devices on the network. In fact, if two devices i and j have both obtained their Link Key with the Trust Center via key establishment, then they both trust the Trust Center. So both devices will use the Trust Center to request a Link Key with each other. The Trust Center will respond to each node individually, sending a randomly generated Link Key LK_{ij}, protecting it by means of the respective Link Key LK_i and LK_j.

D. Leaving the network

A device that has temporarily lost its connection to the network can perform a rejoin by means of a NWK *Rejoin procedure*. First the device must attempt a *secured* rejoin, using the current Network Key. In case of failure (i.e. the Network Key has just been refreshed), it must attempt an unsecured rejoin, which will be successful only if the Trust Center has a Link Key with the device that was established using the Key Establishment Cluster. In case even the *unsecured* rejoin fails, the device has no other options but to repeat the standard Join procedure from the start.

Finally, if a device i leaves the network, the Trust Center must remove the Trust Center Link Key LK_i assigned to that device.

IV. SECURITY CONCERNS AND POSSIBLE SOLUTIONS

In this section we highlight security concerns we have found in the Smart Energy Profile. For each of them we propose a possible approach for a solution.

A. On Supporting Forward Security

In general, a device leaves the network when it has accomplished its mission and thus it is dismissed or when it is momentarily sent to maintenance. Furthermore, a device may be forced to leave the network if it is compromised or suspected to be so. In any case, a device that has left the network must not be able to access any further communication in the

network (forward security) or, otherwise, if it ends up into an adversary's hands this could abuse of the keying material still stored in the device. For this reason, the forward security requirement is typically achieved by a proper key revocation and redistribution (rekeying) policy [1]. In this section we argue that the Smart Energy Profile fails to specify a proper rekeying policy so raising security and efficiency concerns.

As stated in Section III-D, when a device i leaves, or is forced to leave, the network, the Trust Center must revoke the Trust Center Link Key LK_i assigned to that device. In order to do that, the Trust Center can simply delete that key. By doing so, the device will be unable to establish any further connection with the Trust Center, since it is not associated to any valid Link Key anymore.

However, the Smart Energy Profile says nothing about the Network and Link Key management upon device's leaving. More precisely, a device that has left the network still retains the Network Key NK and all the Link Keys LKs it established with peer devices. If these keys are not properly revoked and redistributed, the device remains able to overhear and/or actively take part in all communications protected by means of these keys. In the case of the Network Key, this threat is particularly serious because the device would be able to access all communication protected by the NWK layer security, namely network and application layer commands and even application data messages, when allowed by the specific application profile. Thus, if the device is compromised, the adversary controlling it could exploit the Network Key to spoof and inject bogus routing information—e.g., false routes, bogus information about network status and link conditions—and perform highly disruptive routing attacks such as the sinkhole and selective forwarding attack [11].

As to the Network Key, the ZigBee Specification dictates that the Trust Center must refresh such a key periodically, but it neither clarifies how to determine the refresh period nor, more importantly, specifies any event that asynchronously triggers the Network Key refresh. This implies that an implementation that does not refresh the Network Key upon a node's leaving, and thus becomes exposed to the aforementioned threats, would be still compliant with ZigBee Specification.

A possible solution consists in revoking the Network Key every time a device leaves and redistributing a new one to all remaining nodes. Rekeying also assures that a device which has left will not be able to perform a secured rejoin, being forced to employ the Key Establishment Cluster procedure to rejoin the network.

In fact, ZigBee provides two ways to refresh the Network Key: broadcast-based refresh and unicast-based refresh. In the *broadcast-based* refreshing, the new Network Key NK^+ is protected by means of the current Network Key NK. This is certainly a suitable solution for protecting the Network Key periodic refreshing against an external adversary. However, it is not acceptable for refreshing the Network Key upon a device's leaving of the network. Actually, in this case the current Network Key NK is compromised and cannot be trusted anymore. In such a case, the *unicast-based* refreshing can be used instead. In the unicast-based refreshing, the new Network Key NK^+ is delivered to every device i in a one-to-one fashion, protecting it by means of the device's Transport Key TK_i. Although this solution is secure, it clearly has scalability limitations due to the number of encryptions and rekeying messages that grow according to $\mathcal{O}$(n), where *n* is the number of devices in the network [1].

As to the Link Key between two devices, neither the ZigBee Specification nor the Smart Energy Profile specify how to deal with it when one of the two devices leaves the network. Thus it would be reasonable and wise to invalidate also all the Link Keys a leaving node has established with every other peer device which is still a member of the ZigBee network. However, ZigBee provides no mechanisms to explicitly inform a device that another one has left.

It must be said that ZigBee provides a mechanism to notify that a device is about to leave the network. However, this mechanism is designed just to assure that the network activity, basically the routing process, can be kept alive after a device has left the network. Also an application layer command is present, but it is meant to be used by Routers to inform just the Trust Center that another device has a status that needs to be updated (i.e. it left the network). So, as a matter of fact, ZigBee does not provide an explicit way to inform the nodes about a device which left the network.

A possible solution to efficiently and securely managing rekeying could be at the application level so avoiding the security features directly provided by ZigBee. However, as clarified within ZigBee Specification, every application level protocol message requires proper identifiers for the presently considered application profile, cluster and command, each one with its specific payload. Therefore, the introduction of an application level protocol might involve the extension of existing clusters with new commands or, as an alternative, even the definition of a brand-new manufacturer-specific cluster.

That being said, we claim it is worth defining and introducing a new ZigBee *Key Revocation Cluster*, aimed at coping with devices removal and capable to provide an efficient and secure Link Key and Network Key revocation and redistribution procedure. This cluster considers rekeying in two steps. The first rekeying step deals with Network Key revocation and redistribution. There are many network rekeying protocols properly conceived for networks composed of low-power, low-rate devices [12], [13], [14]. The advantage of implementing network rekeying at the application level consists in letting the application/system developer to choose the rekeying scheme most suitable to the specific scenario requirements and constraints. The second rekeying step deals with the Link Key revocation

and leverages on the previous step. After a device i has left the network and the Network Key has been revoked and redistributed, the Trust Center notifies every remaining device of that event by means of a broadcast authenticated by the new Network Key. Upon receiving the notification message, each device $j, i \neq j$, can verify if it is sharing a Link Key LK_{ij} with i and, if this is the case, discard it.

B. On Supporting Backward Security

In order to guarantee backward security as well, it would be wise to refresh the current Network Key NK each time a new device i is about to join the ZigBee network. If we exclude the presence of malicious nodes within the network, it is sufficient to broadcast a new Network Key NK^+ to all present devices, protecting it via the current Network Key NK. Then every node will start using NK^+ as the current Network Key. Once this procedure has been completed, the new device i is allowed to securely join the network and the Trust Center will provide it with the Network Key NK^+, as described in Section III.

C. Certificate Management

As discussed in Section III-C, the key establishment process follows the Certificate-Based Key Establishment (CBKE) scheme. This implies that every device holds a certificate issued by a Certification Authority (CA). In order to generate certificates and verify their validity, Smart Energy Profile refers to ECQV Implicit Certificate Scheme [15].

CBKE provides each device with an implicit certificate and the public key of the CA releasing the certificate, also called the CA *root key*. Implicit certificates do not include neither the subject public key nor a traditional CA's signature. Thereby they are supposed to be smaller than conventional certificates, as well as more efficient to handle, since there is no signature to verify. However, they make it possible to compute the certified public key, which is retrieved by means of the CA root key.

ZigBee Key Establishment Cluster claims that many different subjects can issue certificates, namely device manufacturers, device distributors, and even end-customers. However, it seems to underestimate the management issue that ensues from the limited storage resources on the ZigBee end devices. Actually, in order that any Coordinator and any end device, possibly coming from different manufacturers or different distributors, can inter-operate, it is necessary that they are able to authenticate each other. This requires that the one is able to verify the other's certificate. It follows that each device should store the root key of every possible certification authority releasing implicit certificates for devices. While we can reasonably assume that a Coordinator has no storage limitations, and thus can keep a large set of root keys, the same does not hold for end devices that have scarce storage resources. Of course, at the other extreme of the spectrum of solutions we can assume the existence of a single certification authority or, at least, a very limited number, so as to manage a meager number of root keys. However, practice proves that this approach is pretty unrealistic and might lead to a monopoly regime.

In order to provide a practical solution to this problem, we introduce another level of certification. We assume that the network administrator, i.e. the installer or the homeowner, runs a *Home Certification Authority* (CA_H) which stores the root keys of the certification authorities releasing implicit certificates for devices. We call this set of keys the *Root Keys Database*. The task of the Home Certification Authority consists in verifying the certificate pre-installed in a device and, if the verification succeeds, issuing a new certificate for that device. We call this process *home-certification*.

More in detail, let D be a device, K_D its public key and $\langle D \rangle_{CA}$ be a certificate released to D by a given certification authority CA (see Fig. 2). The certificate is pre-installed in the device. The Home Certification Authority home-certificates the device according to the following steps.

1) The Home Certification Authority CA_H obtains the device's certificate $\langle D \rangle_{CA}$.
2) CA_H retrieves the CA's root key from the Root Keys Database and verifies $\langle D \rangle_{CA}$ by means of that key (if the key is not present, CA_H obtains it from the Internet and updates the database).
3) CA_H issues a new home-certificate for D, $\langle D \rangle_{CA_H}$.
4) The new home-certificate $\langle D \rangle_{CA_H}$ is installed in the device D.

If the home-certification is applied to the Trust Center at the moment the network is started up, later any device that joins the network needs only to know and store K_{CA_H} in order to authenticate the Trust Center certificate $\langle TC \rangle_{CA_H}$. So doing, the storage demand for authentication is drastically reduced to just one key. The most reasonable choice is to provide a device with K_{CA_H} during the execution instance of the Commissioning process for that device and before the Join procedure is carried out (see Section III). Observe that devices can trust K_{CA_H} since it has been created by CA_H which, in turn, is managed by the network manager. So, trustworthiness of K_{CA_H} is clearly related to the network manager capability to manage CA_H.

If the home-certification is applied not only to the Trust Center but also to the other devices, then also the Trust Center needs to store only the public key K_{CA_H} of the Home

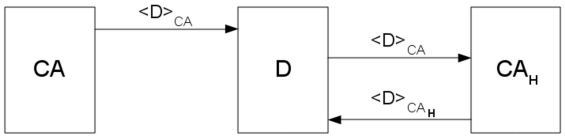

Figure 2. Home-certification.

Certification Authority CA_H to authenticate every device. Thus storage saving may be also at the Trust Center side.

Finally we observe that even the Trust Center can act as the Home Certification Authority CA_H. However, in this case it should have to store the Root Keys Database so losing every benefit in terms of storage saving deriving from the home-certification of devices.

The proposal discussed above introduces some new challenges: the user would have in his own hands a core component of the security architecture. Therefore, he should be capable of managing his home CA and issuing new certificates. Thus, this approach could evidently turn out to be in contrast with the simplicity recommendations provided by ZigBee Specification and ZigBee application profiles about user duties within the network. On the other hand, it is evident the lack of clear specifications about how to organize the Public Key Infrastructure, in order to validate devices certificates. Nevertheless, it is really important to focus on assuring simplicity to the final user while coping with that.

V. Related work

During the last years, security in Home Automation scenarios has been taken into account, pointing out the main issues and suggesting some possible solutions, as well as network architecture models to adopt.

In [16], Pishva and Takeda examine the importance of Smart Home appliances, with particular attention to security challenges, and discuss some countermeasures. They assume this kind of appliances can connect to the Internet, allowing remote control features, but making the home network vulnerable to external attacks too. Security breaches quickly lead to user's privacy violation as well. They also notice the limited resources available on Smart Home devices, as well as the fact that home appliances users are tipically technology-unaware people. Considering the risk of a user/device impersonation attack, they claim that a certification mechanism based on standard and Public Key Infrastructure (PKI) must be used among the entities involved.

In [17], Bergstrom, Driscoll and Kimball focus on an architecture solution for Home Automation. They describe a Global Home Server (GHS) approach, to allow remote control operations to the end users in a simple way, for example by means of a web browser. They assume the presence of a Home Controller Gateway (HCG) within the Home Automation network. It is supposed to be a broadband-enabled device that retains a continuous connection with the GHS to provide a rich user-interface experience. Web-enabled applications allow the homeowners to use the Internet to monitor and control home devices from remote locations. After a secure login, the GHS system establishes a communication session with the appropriate home system and permits the user to view and adjust the home controls. The sensitivity of involved information (i.e. the knowledge that the home is presently in "away" mode) makes communications privacy critical.

VI. Conclusion

We have presented the security model of the ZigBee Smart Energy Profile. This security model is asserting itself as a reference security model for ZigBee application scenarios as it provides a good trade-off between security and complexity. However, the model presents deficiencies concerning key and certificate management that may limit its application. More in details, our paper provides the following contributions.

- A description of the security model in the Smart Energy Profile and, in particular, the commissioning process, device authentication, and key management. We believe that this description may be useful for researchers and developers who approach this topic for the first time.
- We highlight a deficiency in the Network and Link Key management that may cause a violation of the forward security requirement with severe repercussions on the application and network layer. We also propose that the necessary rekeying is managed at the application level by introducing a specific cluster. Such a solution makes it possible to select the rekeying scheme that better suits the application requirements and constraints.
- We highlight that the objectives of openness and interoperability may result in a not scalable certificate management, due to the limited storage resources of ZigBee end devices. In order to overcome this problem we proposed a home-certification mechanism that drastically reduces the storage requirements without endangering security.

Future work will leverage on these results to build a secure middleware for home automation applications.

Acknowledgment

This work has been supported by EU FP7 Network of Excellence CONET (Grant Agreement no. FP7-224053) and EU FP7 Project CHAT (Grant Agreement no. FP7-224428). Authors also wish to thank Claudio Borean and Fabio Luigi Bellifemine from Telecom Italia S.p.A. for discussions in the early stages of this work.

References

[1] Rafaeli and Hutchison, "A survey of key management for secure group communication," *ACM Computing Surveys*, vol. 35, no. 3, pp. 309–329, Sep. 2003.

[2] *Getting Started with ZigBee and IEEE 802.15.4*, Daintree Networks, February 2008, http://www.daintree.net.

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

[4] *Federal Information Processing Standards Publication 197, Specification for the ADVANCED ENCRYPTION STANDARD (AES)*, National Institute of Standards and Technology, November 2001.

[5] *ZigBee Document 053474r17, ZigBee Specification*, ZigBee Alliance, January 2008, http://www.zigbee.org/.

[6] *ZigBee Home Automation Public Application Profile*, ZigBee Alliance, October 2007, http://www.zigbee.org/.

[7] *ZigBee Smart Energy Profile Specification*, ZigBee Alliance, December 2008, http://www.zigbee.org/.

[8] *Understanding ZigBee Commissioning*, Daintree Networks, January 2007, http://www.daintree.net.

[9] *ZigBee Document 064309r04, Commissioning Framework*, ZigBee Alliance.

[10] *Standards for Efficient Cryptography: SEC 1 (working draft) ver 1.7: Elliptic Curve Cryptography*, Certicom Research, November 2006.

[11] Chris Karlof and David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 113–127.

[12] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson, "Group key management scheme for large-scale sensor networks," *Ad Hoc Networks*, vol. 3, no. 5, pp. 668–688, 2005.

[13] Gianluca Dini and Ida Maria Savino, "S2RP: a Secure and Scalable Rekeying Protocol for wireless sensor networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, October 2006, pp. 457–466.

[14] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[15] *Standards for Efficient Cryptography: SEC 4 (working draft) ver 1.1r1: Elliptic Curve Cryptography*, Certicom Research, June 2006.

[16] Davar Pishva and Keiij Takeda, "A product based security model for smart home appliances," in *Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International*, October 2006, pp. 234–242.

[17] Peter Bergstrom, Kevin Driscoll and John Kimball, "Making home automation communications secure," *Computer*, vol. 34, no. 10, pp. 50–56, October 2001.