Security Mechanisms of wireless Building Automation Systems

Karl Jonas
Bastian Vogl
Michael Rademacher

Publisher: Dean Prof. Dr. Wolfgang Heiden

Hochschule Bonn-Rhein-Sieg – University of Applied Sciences
Department of Computer Science

Sankt Augustin, Germany

March 2017

Technical Report 01-2017

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

ISSN 1869-5272

ISBN 978-3-96043-044-5

Digital Object Identifier doi:10.18418/978-3-96043-044-5
DOI-Resolver http://dx.doi.org/

Security Mechanisms of wireless Building Automation Systems*

Karl Jonas, Bastian Vogl, Michael Rademacher
Bonn-Rhein-Sieg University of Applied Science
firstname.lastname@h-brs.de

March 9, 2017

Abstract

This paper describes the security mechanisms of several wireless building automation technologies, namely ZigBee, EnOcean, ZWave, KNX, FS20, and HomeMatic. It is shown that none of the technologies provides the necessary measure of security that should be expected in building automation systems. One of the conclusions drawn is that software embedded in systems that are build for a lifetime of twenty years or more needs to be updatable.

Keywords: security, building automation, ZigBee, EnOcean, ZWave, KNX, FS20, HomeMatic

1 Introduction

Building automation systems (BAS) implement a message-based communication network to initiate actor activities (like temperature regulation or switching off the lights) as a result of some sensor information and simple rule matching. We will avoid to write about *smart home* because we think that there is not much smartness in a (usually static) rule like *"If temperature is below threshold, open valve"*.

Why security is relevant in BAS becomes obvious when a hotel guest gains control over lights and blinds in other guests rooms [SD-Agencies, 2014] or when a smart door lock allows any visitor to come in [Eikenberg, 2015]. Tests performed at the Technical University of Wien have shown that thousands of BAS are accessible over the Internet [Praus and Kastner, 2014].

In this paper we look at several wireless technologies that are typically used to upgrade an existing installation with some automation features. In this scenario it is common to deploy wireless systems rather than cable-based networks, in order to avoid significant (and expensive) construction work inside a building.

*Technical Report of the HBRS MediaCommunication Group; www.mc-lab.de 2016

Note that with one exception all investigated technologies claim to be secure, most of them implementing an AES 128 encryption mechanism.

The paper is structured as follows: After the presentation of related work in chapter 2, security goals for BAS are briefly summarised in chapter 3, followed by encryption methods commonly used in BAS are listed in chapter 4. Our main contribution is provided in chapter 5, which presents the security mechanisms of various wireless BAS communication protocols. Some of these communication protocols are discussed in more detail, with a short analysis of tapped messages, others are only included for completeness without discussion if they do implement any security mechanisms. Finally some conclusions are drawn and plans for our future work are presented.

The paper summarises findings from a thesis written by Bastian van Venrooy under the supervision of Prof. Karl Jonas at Bonn-Rhein-Sieg University in 2015. The complete thesis is available (in German) [van Venrooy, 2016].

2 Related Work

Security challenges and problems of BAS are more present in mass media and popular science than in the research community. However, some researchers have tried to investigate the specific challenges of BAS. In [Brush et al., 2011] the authors conducted semi-structured home visits to 14 households with home automation. One of the barriers for home automation they identified was the difficulty achieving security, leading to the request to provide users with simple security primitives that they can confidently configure. *"Remote access was a double-edged sword for people. The functionality was appealing, but participants worried about introducing a security risk."*

A risk analysis on a smart home automation system has been conducted in [Jacobsson et al., 2016], and it has been shown *"that connected devices may cause undesirable consequences to user privacy with respect to, e.g., access to potentially sensitive meta-information, and the misuse of user-intense mobile devices, and the risk of concept drift as novel devices, such as, surveillance cameras and personal wearables, which are often unplanned for, are dynamically attached to the smart home automation system."*

In [Denning et al., 2013] the authors seek to survey the security and privacy landscape for devices in the home. Their article includes an overview of the structure of attacks to the home ecosystem, differentiating between low-level mechanisms, intermediate goals and high level goals (such as altering logs, gathering incriminating data and physical theft). An important aspect is the possibility to import malicious software into the home network via a mobile device: *"If a device is mobile, then the chances are higher that it will come into contact with malicious or infected networks or devices."* In their conclusion the authors state that *"We need a strategy for how to secure devices in the home. We need to understand the potential risks: risks that are a function of a device's potential exposure to attack, its attractiveness as an attack target, and the potential impacts on human assets if the device is compromised.*

[Praus et al., 2016] provides an extensive survey of the security requirements for distributed control applications and analyses software protection methods. It includes an overview of security in BAS with a focus on BACnet and KNXnet/IP. The authors develop a secure control application architecture for BAS *"being adaptable to all common BAS standards (which) needs to cover BAS specific constraints and be capable of detecting possible attacks."*

The authors in [Armknecht et al., 2016] have identified that *"specifications of security mechanisms often lack explicit descriptions of the envisioned security goals and the underlying assumptions (which) makes it difficult for developers and customers to understand the level of security provided by the systems."* They have developed and provide a formal security model for ZigBee Light Link.

[Harald Glanzer, 2016] discusses the need of BAS for secure communication and high availability. *"For this purpose, the underlying communication system has to be robust and reliable against malicious manipulations."* The paper proposes extensions to KNX for a deployment in critical environments.

The contribution of this paper is an overview and practical investigation of security mechanisms of several wireless BA technologies.

3 Security goals

Security mechanisms implemented in a BAS can be characterised along the following security goals [Sikora, 2003]:

Confidentiality: If an offender is able to get hold of transmitted messages, the content (information) of that message should be hidden from the offender. The typical approach to achieve this goal is message encryption. While it seems obvious that it is impossible to prevent message tapping in wireless systems, it is also very difficult in wired infrastructures. Even if the initial installation secures all cables and lines inside a private area, the next public event will make those cable accessible to an intruder.

Integrity: If a message arrives at its final destination, the receiver should be sure that the content of that message is identical to the original transmission, without any modifications of that content. A method to implement integrity is a checksum attached to the message. If the receiver's calculation of the checksum is identical to the checksum in the message, the message has not been altered (assuming that the offender is not able to calculate a new checksum himself).

Authentication: Identifying the real sender of a message requires an authentication mechanism, like the transmission of a shared secret (password) to make sure that a received message was not generated by an offender.

Availability: Information is available when and where it is needed. Unauthorised persons do not have access to the information and they are not able to prevent availability for authorised persons.

These requirements result immediately into scenarios for an attack: In particular in a wireless communication environment it is very easy for an offender to tap messages that are transmitted over the air. While some technical approaches exist to prevent this (such as limiting transmission power, preventing access to premises), in a typical scenario it should be expected that messages are accessible. Of course this is also the case for large wired infrastructures such as public or commercial buildings, or hotels, where many people can gain access to the physical cabling.

Message transmissions from unauthorised sources can interfere with regular operation of the BAS in various ways. Almost unpreventable, but typically easy to detect, is an interfering signal that prevents regular messages to be received by the BAS. More likely are replay attacks of tapped messages and specifically generated new messages from unauthorised sources when encryption keys have been disclosed. The impact of these attacks depend on the capabilities of the receivers to identify unauthorised messages, enabling them to simply ignore them.

Finally, physical access to the components of a BAS enable an offender to access or manipulate the memory or the configuration of a system. Depending on the system architecture this may result in the failure of a simple device or disclose the security mechanisms of the whole network.

4 Encryption methods

Most of the technologies described in the next chapter use the *Advanced Encryption Standard* (AES) to encrypt data. AES is a symmetric encryption technique that requires knowledge of the same key on both ends of the communication. It comes with different key-lengths and different modes. 128 Bits and 256 Bits are typical key-lengths and considered "secure" [Barker, 2016].

A block cypher splits a message into message blocks and encrypts each block individually before transmission. Stream cypher encrypt a message bit-by-bit, without message splitting. If the same plain-text is encrypted with the same encryption key, the same cypher-text is created. This can be avoided if variable parts are attached to every message. Thus, AES supports various encryption modes and different BAS technologies may support different modes:

AES-128 ECB *Electronic Codebook*: This is the simplest and fastest AES mode. Identical plain-text leads to identical cypher-text. This is the main disadvantage of that mode, making it prone to some forms of security attacks. This is not directly related to the actual encryption method (AES in this case). ECB mode encryption should generally be avoided.

AES-128 CBC *Cipher Block Chaining*: This mode modifies the plain-text blocks prior to encryption. An arbitrary *initialisation vector* (IV) is XORed with the first block, and the cypher-text of the first block is used as the IV of the second block and so on. This use of different IVs with every block results in different cypher-text for identical plain-text.

AES-128 CBC-MAC *CBC-Message Authentication Code*: Uses the CBC-MAC concept to authenticate (sign) a message. The last block of a message is CBC-encrypted and attached to the message.

AES-128 CTR *Counter*: CTR turns a block cipher into a stream cipher. An initialisation vector includes a counter and a random number. Flipping a bit in the cipher-text produces a flipped bit in the plain-text at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

AES-128 CCM *Counter with CBC-MAC*: Provides encryption and authentication with a combination of Counter and CBC-MAC.

There is a trade-off between the level of security and the effort and cost to provide it. We want to ensure that an offender cannot take control over a BAS. If an offender gets access to a component of a BAS, she will be able to compromise or deactivate that component, possibly even to read a secure information out of the devices memory. We silently assume that this kind of security threads exist for all systems and we will not elaborate on them in the remaining of this article. However, they should only affect the information from the local device and not compromise the BAS as a whole.

5 Wireless BA technologies

5.1 ZigBee

ZigBee was introduced in 2004. Three versions exist: ZigBee2004, ZigBee2006 and ZigBee2007 which is equal to ZigBee Pro. Current products do not support the 2004 version anymore, making older and new ZigBee equipment incompatible. Updating a product to a newer and possibly more secure version of ZigBee is usually not supported. The standard is publicly available on the website of the ZigBee alliance. It takes advantage of the IEEE 802.15.4 physical layer for wireless transmission in a Wireless Personal Area Network (WPAN).

For ZigBee2007 two security levels are specified, *High Security (also called* Commercial Security) and *Standard Security (also called* Residential Security).

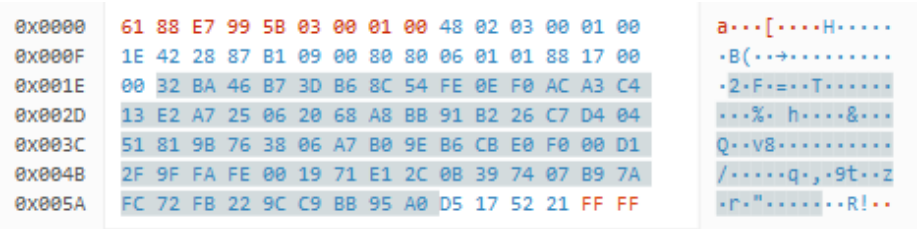

Figure 1: ZigBee packet captured by the sniffer

The major difference between the two levels lays in the methods for key distribution and management [Vidgren et al., 2013] *High Security* level has higher hardware requirements (such as more memory for more encryption keys) and results in more expensive (industrial) products. Most consumer products implement *Standard Security*, including the systems that we investigated. More specific, we used Philips Hue products (a bridge and a Philips Hue bloom) that implement ZigBee Light Link. This protocol is a simplified version of ZigBee that only supports a subset of possible components. The hardware under test was a Philips Hue bridge (model BSB001 with FreeRTOS 6.0.5) and a Philips Hue bloom (model 7299761PH). Like many vendors, Philips provides a smartphone app that can only be used if the products are connected to the public Internet.

The basic key management functionality relies on a pre-defined master-key that is used to encrypt an initial key-exchange between components in a ZigBee Light Link infrastructure. In order to connect the light to the bridge, the bridge needs to know the serial number of the light that is printed on a sticker of the light's power-supply.

We tapped the ZigBee traffic with a USB stick that uses a Texas Instruments CC2531 chip. The *Ubiqua Protocol Analyzer* software simplifies message interpretation. The light was switched on via the meethue website.

The tapped packet in figure 1 was transmitted between the bridge and the light. Figure 2 shows the meaning of the packet. The first nine bytes are MAC header, including source and destination address, the destination PAN ID, a sequence number and control bytes. The MAC payload covers 94 bytes and starts with the NWK header information, again including a source and destination address. The protocol analyser shows that the payload is encrypted. A *Message Integrity Code* (MIC) ensures integrity of the NWK data. We transmitted several identical messages (*Switch lights on*) and could see that the transmitted cyper-text was different in each message. It was not possible for us to identify a pattern in the encrypted part of the messages. The ZigBee-based communications between the Hue bridge and the light appears to be secure.

ZigBee makes use of three types of encryption keys: A master-key is used for initial key-exchange when a new component enters the network. It allows an encrypted transmission of local keys that are used for all further communication. The master-key is pre-set by the equipment vendor. Link-keys are used for individual encryption between any two components in the ZigBee network. If an attacker can get hold of the link-key of two components A and B, it will not be able to decrypt the communication between B and C. Standard security does not support link-keys.

Finally, the network-key is a single key used by all components in the network to encrypt ZigBee traffic of that network. Getting access to the network key allows an attacker to decrypt all traffic in that network. The controller provides this network key to new components during their initial network join. The master-key is used to secure this transmission. In our case this is the ZigBee Light Link master-key that Philips has pre-configured in every component.

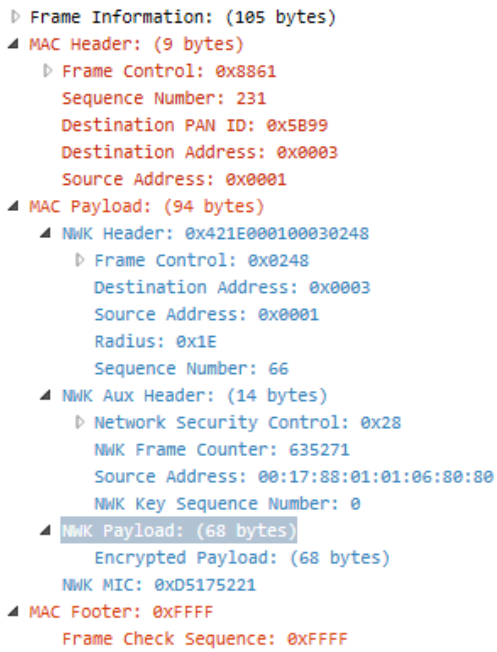

Figure 2: Content of the ZigBee packet

[Durech and Franekova, 2014] and [ZigBee, 2010] describe ZigBee security threads, e.g. replay attacks because an implementation does not verify the frame sequence number. Dhanjani shows how the system can be attacked via its Internet connection [Dhanjani, 2013], describing malware that can cause perpetual blackout. The ZigBee Light Link master key was leaked in 2015 on Twitter by user MayaZigBee [qznc_bot, 2015], compromising the whole security concept. Since any component can request to join the network anytime (insecure join), an intruder can easily get access to the network key. Standard mode does not allow to change this behaviour. ZigBee security is based on secure mechanisms, but the master-key concept, insecure join, the standard and the implementations result in an insecure BAS.

5.2 EnOcean

EnOcean was specified as an international standard ISO/IEC 14543-3-10 in 2012 [ISO, 2012]. Its focus includes low-energy BAS, enabling devices to perform energy harvesting: pushing the button of a switch produces enough energy to transmit the signal. Solar cells or differences in temperature are other methods supported by EnOcean hardware. The energy concepts limit the communication and encryption capabilities of the systems.

EnOcean differentiates between an insecure and a secure mode. Secure mode allows a combination of encryption, authentication and rolling code. The initial distribution of the encryption key can be encrypted (if a shared pre-defined key exists) or unencrypted. Encryption is based on AES and includes AES-CBC, details of the variants deployed by EnOcean can be found in [EnOcean, 2013]. In order to activate pairing and negotiation, it is required to push a physical button at the enOcean base station.

We used the ESK300 Starter Kit for our experiments, more specifically the USB300DB gateway and a PTM215 switch. Dolphin View software (Advanced Version 3.6.0.0) enabled message analysis. Pushing and releasing button A0 results in two unencrypted messages as depicted in figure 3.

Byte	Description	HexVal	Meaning
1	sync byte	55	
2,3	length data	00 07	7
4	length optional data	07	7
5	packet type	01	radio msg
6	checksum	7A	
7	RORG	F6	repeated switch comm.
8	payload	20	release
9,10,11,12	sender ID	FF FE E1 E5	
13	status	30	0011 0000
14	msg part	02	
15,16,17,18	receiver ID	FF FF FF FF	broadcast
19	receive strength	34	-52 dBm
20	payload	00	
21	checksum	C9	

Table 1: Meaning of the EnOcean packet

The meaning of the message content of the lower-line message is summarised in table 1.

The initial key exchange for secure mode operation can be encrypted (if a pre-shared secret exists) or unencrypted. Initialising the process requires physical access to the controller (push-button). Dolphin View supports various security parameters such as the length of the rolling code (24 or 32 bits).

The tapped message in figure 4 was the result of pushing the A1 button of the switch. It uses 24-bit CMAC and VXOR AES encryption, and a 16-bit rolling code. The first line is the packet header with sync-byte, packet-size and packet type. The second line starts with REORG set to 0x30 which means that this is a secure message, followed by four bytes of encrypted message content. The remaining bytes are the packet trailer, including sender and receiver addresses. First and last line are very similar to the corresponding unencrypted message.

```
55 00 0A 07 01 EB  \
30 02 14 6E 09 FE FE E1 E5 00 \
02 FF FF FF FF 31 00 01
```

Figure 4: Encrypted EnOcean message

The initial distribution of the encryption key is based on a pre-shared master key or unencrypted. The key is identical for all components served by a single base station. Since key-exchange requires physical access to the gateway, it is difficult for an intruder to initiate this key exchange

```
55 00 07 07 01 7A F6 30 FE FE E1 E5 30 02 FF FF FF FF 37 00 89
55 00 07 07 01 7A F6 20 FE FE E1 E5 30 02 FF FF FF FF 34 00 C9
```

Figure 3: EnOcean packet captured by the sniffer

in order to get access to the encryption key in the network. *After* this initial distribution of the encryption key, enOcean is a rather secure BAS, supporting encryption, authentication and integrity. It seems that enOcean encryption provides integrity, authentication and prevention against replay. Unfortunately, in CBC mode EnOcean uses the same IV for all messages.

Experiments with a second enOcean component revealed another problem. A window handle uses enOcean to inform the base station about its current status (open or closed). While the messages themselves were encrypted, *OPEN* and *CLOSED* messages were of different length. So while it was not possible to decypt the message itself, it was easy to determine the status from the message length.

5.3 Z-Wave

The proprietary protocol was developed by Sigma Designs and licensed to many BAS vendors. The specification is confidential and business partners have to sign a Non-Disclosure-Agreement before they can implement the protocol. In order to ensure interoperability, all products require certification by the Z-Wave Alliance. A product that provides encryption may use the Z-Wave Plus Logo, so it can be assumed that only those components support security mechanisms.

Parts of the protocol have been re-engineered, some confidential specifications are available on the Internet. OpenZwave [OpenZwave, 2016a] [OpenZwave, 2016b] provides an open source implementation that supports *"most switches, dimmers, thermostats, energy monitors, motion sensors, appliance modules, key fobs, door/window sensors"* including security related devices such as locks.

For our experiments we used the *MT2600 Home Control Central Unit* and a *MT02646 Metering Plug* from Devolo. Installation of the controller requires Internet connectivity and an account on the Devolo website. During the mandatory registration process, private user data has to be provided to Devolo in order to gain access to the controller. In order to include a new component in the home network, the controller needs to be brought in peering mode via the Internet connection. While the specification of the metering plug states that the plug supports 128 Bit AES encryption, the webinterface does not provide any related information and does not offer a method to enable or disable these settings.

We used third-party components for unencrypted communication to the plug. FHEM open source software running on a Raspberry Pi used a USB stick from Z-Wave.Me to send unencrypted messages to the plug. The plug reacted immediately and could be switched with messages `00 13 0303 25 01 FF 25 03` (ON) and `00 13 0303 25 01 00 25 03` (OFF).

If secure mode is used, a network-wide encryption key K_n is used to create a message key and an authentication key:

$$K_c = AES - ECB_K n(Word_c)$$

$$K_m = AES - ECB_K n(Word_m)$$

These two keys are used to encrypt and authenticate messages in secure mode. $Word_c$ and $Word_m$ are stored in the firmware of the Z-Wave components. The message authentication code (MAC) which is used to authenticate the message includes a random number (nonce) to prevent message replay.

The network key is provided by the controller and encrypted with a master-key. Z-Wave components that received the network key and used it to generate packet- and authentication keys should only use these keys for a secure communication. In their security evaluation [Fouladi and Ghanoun, 2013] the authors took advantage of the fact that the master-key is stored in the Z-Wave hardware and has been revealed. An attacker can generate a valid new network key and send it to a component. The authors describe a security attack against a key-lock where the lock could be convinced to accept this new network key. The attacker was then able to control the lock. Since it was not possible to update the lock software, all locks needed to be replaced.

The example shows two important aspects of security in BAS:

1. Closed source prevents early detection of security problems. Many products may be installed before the problem becomes obvious. (However, various security problems in open source ssh implementations show that open source does not provide any security guarantees either)

2. Implementation of software update mechanisms should be mandatory in order to prevent the necessity of physical replacement of components with a compromised security concept.

5.4 KNX

While KNX has its main market share as a wired bus-based BAS, its RF-version supports wireless command and control. The standard failed to develop any security mechanisms for many years and has just recently released initial security mechanisms based on AES. Since vendors have not started to implement them, to the knowledge of the authors all KNX sensors and actors use insecure communication and can be easily controlled by an offender. Not to mention that also KNX components do not support an update of the security mechanisms.

Since KNX is the market leading BA technology in large commercial and public buildings, including critical infrastructure such as hospitals, its ignorance towards security mechanisms appears to be particularly problematic. In addition, KNX component cost is at the upper end of the line, where the additional cost for a secure implementation should be easier to compensate.

Many examples for security attacks against KNX exist and we will only mention the one in Shenzhen because it is interesting how the KNX association reacted to it: *"Jesus Molina, who was staying at the St. Regis Shenzhen hotel, found that he could easily take control of the thermostats, lights, TVs and window blinds in all of the hotel's 250-plus rooms, as well as alter the electronic "Do Not Disturb" lights outside each door—all from the comfort of his luxurious bed"* [Zetter, 2014].

The official reply from KNX focused on the request that *"it is essential that separate Wi-Fi networks are used for these purposes, one for public access of which the key can be known and one for communication between the digital butler and the hotel room, with a key that is not revealed. In latter case, the researcher would first have had to hack his way into the Wi-Fi, before he could have achieved anything via KNX IP"*. The association states that *"The cradle of the KNX system lies as far back as the nineties, where security issues were not such a hot topic as they are today"*, leaving the security burden to the network administrator (who probably does not have an idea about how KNX works) [KNX Association, 2014] .

Our own experiments included a simple replay attack against a roller shutter whose vendor promotes the shutter with the additional security it provides. It took a student only a few minutes to control the shutter, using a laptop and a low-cost USB transceiver.

5.5 HomeMatic

This system developed by eq-3 is one of the dominating BAS for private houses in Germany. It followed the earlier FS20 system that is especially popular in budget-priced wireless BAS like simple sockets and did not implement any kind of security. Some of the systems support AES encryption and are equipped with a system default key that is identical in all HM components. The default key is publicly available on the Internet.

For a long time eq-3 discouraged users to change the default system key. In the meantime (since 2014), since the security problems became public to a wider audience, eq-3 changed its mind and requests users to change the system key [HomeMatic Inside, 2014].

We used a HM CCU2 as a central controller and a wireless plug socket HM-ES-PMSw1-Pl for our experiments. The BidCoS *Bidirectional Communication Standard* is the proprietary protocol developed and used by eq-3. It has not been made public, but most features have been reverse-engineered by open-source projects such as FHEM [FHEM, 2016].

```
A 0E 2A A0 11 F11034 2C0EA8 0201C8
```

Figure 5: HomeMatic Message

Figure 5 shows a message from the controller to the socket. Its interpretation is provided in table 2.

We used a CUL to replicate the message. While the plug indicates that it has received a message with a wrong counter, it switches state according to the replayed packet. Since the plug sends regular status information (approx. every three minutes), sender and receiver ID can be considered public.

Changing connection mode from *standard* to *secure* puts the communication between the controller and the plug into secure mode. Tapping HM messages reveals that even in secure mode messages are not encrypted. In fact, while HM

Byte	Description	HexVal	Meaning
1	command	A	AskSin
2	length	0E	14
3	counter	2A	42
4	flag	A0	160
5	type	11	17
6,7,8	sender	F11034	Sender ID
9,10,11	receiver	2C0EA8	Receiver ID
14,13,14	payload	0201C8	ON

Table 2: Meaning of the HomeMatic packet

markets its products as secure with AES support, all messages are transmitted as plain text. *Secure mode* is implemented as a simple challenge-response procedure where the AES key is used to authenticate the sender of a message.

A 0E78A011318EC02C0EA80201C80000

A 1178A0022C0EA8318EC0044F1E66D0C65B00

A 1978A003318EC02C0EA85394CD90BF455F1F066AEB343415028D

A 127880022C0EA8318EC00101C8001CBCB0986A

Figure 6: HomeMatic Challenge-Response sequence

The first message in figure 6 contains the controller's request 0x0201C8 to switch on the plug socket. The socket replies with an AES-encrypted challenge 0x044F1E66D0C65B00 that needs to be solved by the controller in order to verify that the controller is entitled to send requests. The controller responses to the challenge and finally the socket confirms that it will switch on.

So while message content is not encrypted by secure mode, the challenge-response prevents unauthorised commands to be accepted by the plug.

6 Conclusions and Future Work

None of the studied home automation technologies can be considered secure. The ZigBee Light Link experience teaches that a "secret" master key is not a good idea. The KNX approach to move the burden of security "out of scope" cannot be a solution. HomeMatic ignores any privacy issues and implementation specific problems can make any security architecture fail.

As we have learned from many areas, security is not a static feature, but develops over time. It becomes obvious that it is a major problem when existing systems cannot be updated after the security mechanisms have been compromised. And compatibility requires that even new devices implement compro-

mised mechanisms. Another aspect is that implementation of security mechanisms should be open-source. Obfuscation is not a concept to make software safe and secure.

Future work follows two directions: We want to find a good concept to update firmware, even in very small devices of a BAS, such as a light switch. Open Mobile Alliance Device Management and Eclipse IoT seem to be frameworks to start with. This will also allow an adaptation of the communication protocols to changing user needs, such as switching from one protocol to another, provided that the physical requirements (such as frequency and modulation) can be met by the hardware. But in particular switching between ZigBee, Bluetooth and 6Lowpan is simple, since they all rely on the same underlying protocol. Second, we need to find key distribution mechanisms that are not prone to the publication of a single master key. Component specific keys may be an approach that prevents networks from being compromised if a single component is.

It is assumed that future BAS sensors and actors can rely on more powerful micro-controllers than today, making it easier to implement reliability and security even in very small systems. An open issue is heterogeneity, both in terms of protocols on all levels and in case of (wired and wireless) transmission infrastructure.

Finally, technical security goes hand-in-hand with usability. Security mechanisms must be implemented in a way that the user wants to use them, otherwise he will just ignore or circumvent them.

References

[Armknecht et al., 2016] Armknecht, F., Benenson, Z., Morgner, P., and Müller, C. (2016). On the Security of the ZigBee Light Link Touchlink Commissioning Procedure. In *Sicherheit 2016, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016*, pages 229–240.

[Barker, 2016] Barker, E. (2016). Recommendation for Key Management. NIST Special Publication 800-57 Part 1 Revision 4. `http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4`.

[Brush et al., 2011] Brush, A. B., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., and Dixon, C. (2011). Home Automation in the Wild: Challenges and Opportunities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2115–2124, New York, NY, USA. ACM.

[Denning et al., 2013] Denning, T., Kohno, T., and Levy, H. M. (2013). Computer Security and the Modern Home. *Commun. ACM*, 56(1):94–103.

[Dhanjani, 2013] Dhanjani, N. (2013). Hacking Lightbulbs. Website accessed 2016-06-08. `http://www.dhanjani.com/docs/Hacking%20Lighbulbs%20Hue%20Dhanjani%202013.pdf`.

[Durech and Franekova, 2014] Durech, J. and Franekova, M. (2014). Security attacks to ZigBee technology and their practical realization. In *Proceedings of the 12th International Symposium on In Applied Machine Intelligence and Informatics SAMI)*, page 345–349.

[Eikenberg, 2015] Eikenberg, R. (2015). Smartes Türschloss August war zu gastfreundlich. Website accessed 2016-04-19. `http://www.heise.de/security/meldung/Smartes-Tuerschloss-August-war-zu-gastfreundlich-2593822.html`.

[EnOcean, 2013] EnOcean (2013). Security of EnOcean Radio Networks. Website accessed 2016-05-21. `https://www.enocean.com/fileadmin/redaktion/pdf/tec_docs/Security_of_EnOcean_Radio_Networks.pdf`.

[FHEM, 2016] FHEM (2016). FHEM project. Website accessed 2016-07-07. `http://fhem.de/fhem.html`.

[Fouladi and Ghanoun, 2013] Fouladi, B. and Ghanoun, S. (2013). Security evaluation of the Z-Wave wireless protocol. *Black hat USA*, 24.

[Harald Glanzer, 2016] Harald Glanzer, Lukas Krammer, W. K. (2016). Increasing security and availability in KNX networks. In *Sicherheit 2016, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016*, pages 241–252.

[HomeMatic Inside, 2014] HomeMatic Inside (2014). Der System-Sicherheitsschlüssel. Website accessed 2016-07-07. `https://www.homematic-inside.de/tecbase/homematic/webui/item/der-system-sicherheitsschluessel`.

[ISO, 2012] ISO (2012). Information technology – Home Electronic Systems (HES) – Part 3-10: Wireless Short-Packet (WSP) protocol optimized for energy harvesting – Architecture and lower layer protocols. standard 14543-3-10:2012.

[Jacobsson et al., 2016] Jacobsson, A., Boldt, M., and Carlsson, B. (2016). A Risk Analysis of a Smart Home Automation System. *Future Gener. Comput. Syst.*, 56(C):719–733.

[KNX Association, 2014] KNX Association (2014). Knx security statement. Website accessed 2016-07-07. `https://www.knx.org/knx-en/news/KNX-Security-Statement/details.php?ref=487`.

[OpenZwave, 2016a] OpenZwave (2016a). OpenZwave. Website accessed 2016-07-07. `http://www.openzwave.com/`.

[OpenZwave, 2016b] OpenZwave (2016b). OpenZwave on Github. Website accessed 2016-07-07. `https://github.com/openzwave/`.

[Praus and Kastner, 2014] Praus, F. and Kastner, W. (2014). Identifying unsecured building automation installations. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–4.

[Praus et al., 2016] Praus, F., Kastner, W., and Palensky, P. (2016). Software Security Requirements in Building Automation. In *Sicherheit 2016, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016*, pages 217–228.

[qznc_bot, 2015] qznc_bot (2015). ZigBee light link master key. Website accessed 2016-06-08. `https://www.reddit.com/r/hackernews/comments/2zzt2x/zigbee_light_link_master_key/`.

[SD-Agencies, 2014] SD-Agencies (2014). Hotel's security flaws exposed. Website accessed 2016-04-19. `http://www.szdaily.com/content/2014-07/22/content_9847323.htm`.

[Sikora, 2003] Sikora, A. (2003). *Technische Grundlagen der Rechnerkommunikation: Internet-Protokolle und Anwendungen*. Carl Hanser Verlag GmbH & Co. KG.

[van Venrooy, 2016] van Venrooy, B. (2016). Sicherheit in der Heimautomatisierung. Bachelor thesis, Bonn-Rhein-Sieg University. `http://mc-lab.inf.h-brs.de/doc/ha/2016_Venrooy_BA_Sicherheitsmechanismen.pdf`.

[Vidgren et al., 2013] Vidgren, N., Haataja, K., Patiño-Andres, J. L., Ramírez-Sanchis, J. J., and Toivanen, P. (2013). Security threats in zigbee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 5132–5138.

[Zetter, 2014] Zetter, K. (2014). Here's How Easy It Could Be for Hackers to Control Your Hotel Room. Website accessed 2016-07-07. `https://www.wired.com/2014/07/hacking-hotel-room-controls/`.

[ZigBee, 2010] ZigBee (2010). ZigBee Specification. `PublishedbytheZigBeeStandardsOrganisation,accessed:30/11/2015`.