Magdeburger Journal zur Sicherheitsforschung

Gegründet 2011 | ISSN: 2192-4260
Herausgegeben von Stefan Schumacher
Erschienen im Magdeburger Institut für Sicherheitsforschung

This article appears in the special edition »In Depth Security – Proceedings of the DeepSec Conferences«.
Edited by Stefan Schumacher and René Pfeiffer

# ZigBee Exploited

## The good, the bad and the ugly

**Tobias Zillner**

The Internet of Things (IoT) is an emerging trend. IoT involves the integration of digital and wireless technologies in physical objects and systems, especially those historically unconnected, which are supposed to make our everyday life easy and convenient. One of the most widespread used wireless technologies to connect IoT devices is the ZigBee standard. This emerging technology needs to keep pace with customer demands for cheap, long-living and available devices. One of the major challenges besides user and industry acceptance is security. However, security is very often sacrificed or neglected due to fear of reduced or limited usability or fear of breaking backwards compatibility.

This paper describes the actual applied security measures in ZigBee, highlights the included weaknesses and introduces a software framework that can be used to automatically audit ZigBee communication and the implementation of ZigBee security services for various vulnerabilities and exploit them.

Citation: Zillner, T. (2016). ZigBee Exploited: The good, the bad and the ugly. *Magdeburger Journal zur Sicherheitsforschung*, *12*, 699–704. Retrieved September 2, 2016, from http://www.sicherheitsforschung-magdeburg.de/publikationen/journal.html

# 1 Introduction

IoT is considered to be the next phase of the Internet revolution. Linking physical objects in the real world to the virtual world and enabling anytime, anyplace and anything communication. (Santucci 2010, p. 11) Communication between devices is mainly carried out using wireless channels, which introduces various security issues. Some of these weaknesses are new, but most have actually been around for a long time. A desired short time-to-market, as well as backward compatibility and future proofing considerations lead to the persistence of known problems. The ZigBee standard is one of the dominating standards for wireless communication between IoT devices. Even though it was created with security in mind, low per-unit-costs and usability as well as compatibility factors lead to poor implementation of security controls, which pose security risks. With the availability of consumer-ready, programmable radio systems and low-cost devices with sufficient computational power, the field of Software-defined-radio (SDR) is experiencing rapid growth enabling researchers to audit wireless communication beside traditional Wi-Fi.

This paper highlights the main security risks in ZigBee implementations, the devices that are affected and the results of a practical assessments of ZigBee enabled device.

# 2 The ZigBee Standard

ZigBee is a standard for personal-area networks developed by the ZigBee Alliance (including companies like Samsung, Philips, Motorola, Texas Instruments and many others) with the aim of providing a low-cost, low-power consumption, two-way, reliable, wireless communication standard for short range applications. (ZigBee Alliance 2008, p. 29) The standard is completely open and gained ratification by the Institute of Electrical and Electronics Engineer (IEEE) in 2003. The protocol stack of ZigBee is based on IEEE 802.15.4. Advantages of choosing ZigBee are the provision of long battery lifetime, the support of a large number of nodes (up-to 65000) in a network, the easy deployment, the low costs and global usage. (Kaur & Sharma 2013, ZigBee Alliance 2014)

ZigBee is used for example in following areas (ZigBee Alliance 2014):

- Remote Control
- Input Devices
- Home Automation
- Building Automation
- Health Care
- Telecom Services
- Retail Services
- Smart Energy

The ZigBee stack consists of four layers: (ZigBee Alliance 2008, p. 35)

- Physical Layer (PHY)
- Medium Access Control Layer (MAC)
- Network Layer (NWK)
- Application Layer (APL)

The IEEE 802.15.4-2003 standard is used for the two lowest layers, the physical layer (PHY) and the medium access control layer (MAC). The other two layers are defined by the ZigBee Protocol Stack.

From a security perspective, the network and the application layer are of highest relevance and are therefore described in more detail in the next chapter.

# 3 ZigBee Security

The ZigBee standard includes complex security measures to ensure key establishment, secure networks, key transport and frame security. (ZigBee Alliance 2008, p. 419 f). Those services are implemented at the Network and the Application Support Sublayer (APS), a sub layer of the Application Layer. The ZigBee protocol is based on an »open trust« model. This means all protocol stack layers trust each other. Therefore cryptographic protection only occurs between devices. Every layer is responsible for the security of their respective frames.

The security of ZigBee networks is based on their encryption keys. It is possible to distinguish between two types of security keys. (ZigBee Alliance 2008, p. 422)

- Network key is used to secure broadcast communication. This 128-bit key is shared among all devices in the network. Usually multiple network keys are stored by the Trust Center, but only one network key is the active network key. The current active network key is identified by a sequence number and may be used by the NWK and APL layers of a device. A device must acquire a network key via key-transport or pre-installation.
- Link key is used to secure unicast communication at the Application layer. Each two communicating devices share a 128-bit key. Link keys are acquired either via key-transport, key- establishment, or pre-installation (for example, during factory installation).

## 3.1 Network Layer Security

The ZigBee Network Layer ensures the integrity and encryption of the transmitted frames by applying AES encryption (AES CCM mode) with a key length of 128 bit, and by using a cipher block chaining message authentication code (CBC-MAC). (ZigBee Alliance 2008, p. 423)

### 3.2 Application Support Sublayer Security

If a frame originated at the APS layer needs to be secured, the APS layer is responsible for the proper protection of that frame. The APS layer allows frame security to be based on link keys or the network key. If the active network key should be used for frame protection, the APS layer first checks if the frame gets protected on NWK layer. If so, the frame just gets passed to the NWK layer and the frame protection is performed on the NWK layer. The APS layer is also responsible for providing applications with key establishment, key transport and device management services. (ZigBee Alliance 2008, p. 424)

The ZigBee standard states the following about the security of ZigBee installations: »The level of security provided by the ZigBee security architecture depends on the safekeeping of the symmetric keys, on the protection mechanisms employed, and on the proper implementation of the cryptographic mechanisms and associated security policies involved. Trust in the security architecture ultimately reduces to trust in the secure initialisation and installation of keying material and to trust in the secure processing and storage of keying material.« (ZigBee Alliance 2008, p. 420).

As stated above, the ZigBee Security is based on the assumption that keys are securely stored, and devices are pre-loaded with symmetric keys so they have never to be transmitted unencrypted.

But there are exceptions to this policy. If a non-preconfigured device joins a network, a single key may be sent unprotected and enable encrypted communication. This one-time transmission of the unprotected key results in a short timeframe of exploitability in which the key could be sniffed by an attacker. Since the security is dependent on the safekeeping of the encryption keys such a key interception leads to a critical security issue and compromises the security of the whole network. Even thought the timeframe seems to be narrow, an attacker could use jamming techniques to trick the user to initiate a factory reset or another way of re-joining, re-establishing that attack time-frame.

Another exception is made due to the low-cost nature of some types of devices such as light switches or temperature sensors. Because of their limited capabilities, it cannot be assumed that the hardware is built tamper-resistant. So if an attacker gets physical access to such a device, it may be possible to access the secret keying material and other privileged information, as well as access to the security software and hardware. (ZigBee Alliance 2008, p. 420)

## 4 ZigBee Application Profiles

The key to communicating between devices on a ZigBee network is the usage of application profiles. Application profiles are agreements for messages, message formats, and processing actions that enable developers to create an interoperable, distributed application employing application entities that reside on separate devices. These application profiles enable applications to send commands, request data, and process commands and requests. As one ZigBee device might be a multi-purpose-device, different profiles are created to allow devices of various vendors to properly communicate with each other using those predefined profiles.

### 4.1 ZigBee Home Automation Public Application Profile (HAPAP)

An example of a profile would be the home automation profile. This ZigBee profile permits a series of device types to exchange control messages to form a wireless home automation application. These devices are designed to exchange well-known messages to effect control such as turning a lamp on or off, sending a light sensor measurement to a lighting controller, or sending an alert message if an occupancy sensor detects movement.

This means if a manufacturer wants a device to be compatible to other certified devices from other manufacturers, the device has to implement the standard interfaces and practices of this profile. To provide this kind of interoperability all ZigBee Home Automation devices should implement so called Startup Attribute Sets (SAS). From a security standpoint, the following two specified attributes are of particular interest:

- Default Trust Center Link Key
  - 0x5A 0x69 0x67 0x42 0x65 0x65 0x41 0x6C 0x6C 0x69 0x61 0x6E 0x63 0x65 0x30 0x39
  - Note: The Link Key is listed in little-endian format.
- Default Link Key Join
  - 0x01 (True).
  - This flag enables the use of default link key join as a fallback case at startup time.

The use of the default TC link key `ZigBeeAlliance09` introduces a high risk to the secrecy of the network key. The Home Automation Public Application Profile states that: »The current network key shall be transported using the default TC link key in the case where the joining device is unknown or has no specific authorization associated with it. This allows for the case where alternative pre-configured link keys specifically associated with a device can be used as well.« (ZigBee Alliance 2013, p. 44) Since, as discussed before, the security of ZigBee is highly reliant on the secrecy of the key material and therefore on the secure initialisation and transport of the encryption keys, this default fallback mechanism has to be considered as a critical risk. If an attacker is able to sniff a device join using the default TC link key, the active network key is compromised and the confidentiality of the whole network communication can be considered as compromised. This might be a lower risk if only light bulbs are used, but as HVAC systems and door-locks

also use the Home-Automation profile, the impact on security of this profile requirement is greatly increased.

### 4.2 ZigBee Light Link Profile (ZLL)

The ZigBee Light Link (ZLL) profile addresses devices and functionality in the over-the-counter, consumer lighting application domain. (ZigBee Alliance 2012, p. 1)

Devices in a ZLL shall use ZigBee network layer security. During classical ZigBee commissioning where a non-ZLL device is being joined to a ZLL network without a trust center, a pre-installed link key is used to secure the transfer of the network key when authenticating. The ZLL pre-installed link key is a secret shared by all certified ZLL devices. It will be distributed only to certified manufacturers and is bound with a safekeeping contract. Additionally, if the decryption of the APS message fails with the key described above, ZLL devices shall try to decode the APS message using the known default trust center link key. Like the HAPAP, the ZLL profile also specifies »ZigBeeAlliance09« as the default Trust center link key in the SAS and requires the support of an insecure join as a fallback. This leads also to the same vulnerable initial key exchange. Even if the manufacturer implemented a secure key exchange and distributed proper key material, it would be possible for an external attacker to disturb the network join using selective jamming and then wait for the insecure join to get access to the exchanged key material.

As every ZLL device joining to a ZLL network per definition shall use the ZLL master key to derive the active network key, knowledge of the ZLL master key allows an attacker to intercept the key exchange and acquire the current active network key. This would then allow the attacker to control all devices in the ZigBee network. As the ZLL master key has supposedly been leaked in the Internet (e.g. on reddit and some online forums), the security of the ZLL devices has to be considered as compromised.

Besides the leaked key, ZLL devices support a feature called »Touchlink Commissioning« that allows devices to be paired with controllers. As the default and publicly known TC link key is used, devices can be »stolen«. Tests showed that amateur radio hardware such as a Rasperry Pi extension board with normal dipole antennas already allowed Touchlink Commission from several meters away whereas for security reasons this should only work in close proximity. Usage of professional radio equipment would allow an even higher distance for such a successful device takeover.

## 5 SecBee – A new ZigBee Security Testing Tool

Since ZigBee provides some very specific security services and attack vectors, a tool that enables security researchers, testers and developers to check the configuration and implementation of security services of their product was developed. Unlike other tools for ZigBee testing, it enables testers to check encrypted networks and automatically perform ZigBee specific tests such as network leaves / joins, resetting to factory defaults or searching for unsecure key transport.

SecBee[1] is based on scapy-radio[2] and killerbee[3], but enhances the functionality drastically and also fixes some limitations of these tools.

## 6 Real world assessments and identified vulnerabilities

To verify the implementation of ZigBee security in real world devices, a home automation system, a smart lighting solution and a ZigBee enabled door lock were assessed using the newly developed ZigBee security testing tool - SecBee. The practical security analysis of every assessed device showed that the solutions are designed for easy setup and usage but lack configuration possibilities for security and perform a vulnerable device pairing procedure that allows external parties to sniff the exchanged network key. Even if the timeframe to exploit the vulnerability is very limited, bringing the user into play can easily circumvent this. ZigBee communication can be easily jammed. Since ZigBee is designed for low power communication and energy saving this can be easily achieved by simply sending noise on the target ZigBee channel to prevent successful communication. A typical user would notice a lost connection and therefore just perform a re-pairing procedure to solve this issue. Targeting the user level allows an attacker to enforce a re-pairing and sniff the transmitted network key. This would allow an attacker to get complete control of the system as the security of the solution is solely relying on the secrecy of this key.

Furthermore, the tested home automation system is not capable of resetting or changing the applied network key, so even if a user notices unwanted behaviour in the network, there would be absolutely no possibility of locking the intruder out. Also, no automatic key rotation could be identified during a timeframe of eleven months.

The smart lighting solution is also vulnerable to a device takeover from any external party. It was possible to steal light bulbs and join them to a fake network without knowledge of the active secret keys. An

1 https://github.com/zu1na/SecBee/

2 https://bitbucket.org/cybertools/scapy-radio/

3 https://code.google.com/p/killerbee/

attacker just has to send a »reset to factory default« command to the light bulb and wait for the bulb to search for ZigBee networks to join. The bulb will connect to the first network available without any further interaction of a user. No button or similar has to be pressed. The light bulb is always sending beacon requests to look for a new network to join.

In addition, it should be noted that the usage of wireless communication systems for security applications like surveillance is not recommended as the communication can easily be disturbed with simple jamming and no tested device implemented measures like a heartbeat message to provide the central device with information about the actual status. This attack scenario becomes increasingly likely as the prices for radio hardware are getting lower, the hardware is publicly available and open source tools exist that provide the necessary features to perform attacks on wireless networks. It is just a matter of time till the first real world incident will become public.

# 7 Conclusion

The security features provided by the ZigBee standard can be considered as very strong and robust. ZigBee encryption is based on the well known AES algorithm for data encryption and data authentication. The security is dependent on the secrecy of the encryption keys as well as their secure initialisation and distribution of the encryption keys. However, the actual specifications of application profiles such as the Home Automation Public Application Profile introduced failures and shortfalls and therefore security risks. Also, among the main constraints in implementing security features in a ZigBee wireless network, the limited resources are a challenge. The nodes are mainly battery powered and have limited computational power and memory size. Therefore, it is essential for security to fulfil some preconditions on implementation side, which are the following:

- Device Tampering: ZigBee is targeted for low-cost applications, and the nodes hardware may not be tamper resistant. If an intruder acquires a node from an operating network that has no anti-tamper measures, the actual key could be obtained simply from the device memory. A tamper- resistant node could erase the sensitive information including the security keys if tampering is detected.
- Key Transport: The default TC link key should not be used since this key is considered as public knowledge and therefore provides the same level of security as unencrypted key transport.
- Key Establishment: The master keys used during key establishment shall be distributed via out-of-band channels. For example a sticker with a preconfigured master key could be attached to a device and entered by the user during device setup.
- Key Rotation: The security of the communication is dependent on the secrecy of the network key and of the link keys. The network key shall be changed periodically. Key management in form of changing the network key in a meaningful time period or after a certain number of messages should be introduced. Otherwise known plaintext or other attacks on the security of AES may be possible.

Tests with light bulbs and even door locks have shown that the vendors of the tested devices implemented the minimum of the features required to be certified, including the default TC fallback key. No other options were implemented and available to the end-user.

Also relying on the secrecy of keys distributed only among a limited group of people, as the ZLL profile requires, is a security method known to have failed before. Travis Goodspeed showed successful attacks on ZigBee hardware to extract keys (Goodspeed 2009 p. 1f), and thus without appropriate hardware, key secrecy should not be the foundation of the ZigBee product's security architecture.

# 8 About the Author

Tobias Zillner runs his own security consulting company and works as independent researcher on several security projects. He conducts information systems audits in order to assess compliance to relevant internal and external requirements and to provide a customers management with an independent opinion regarding the effectiveness, and efficiency of IT systems. Furthermore, Tobias evaluates and assures security of Information Technology by performing webapplication and web service penetration tests, source code analysis as well as network and infrastructure penetration tests. He has a Bachelor degree in Computer and Media Security, a Master degree in IT Security and a Master degree in Information Systems Management. Tobias expertise also applies to the IT Governance, Risk and Compliance domains. He also holds a wide range of certifications, like CISSP, CISA, QSA, CEH, ITIL or COBIT and is a frequent speaker at industry leading security conferences, such as Black Hat, DeepSec, BSides, CRESTcon or Defcon.

# 9 Bibliography


- Goodspeed, T. (2009), Extracting Keys from Second Generation Zigbee Chips. Black Hat USA, Las Vegas.
- Santucci, G., (2010). Vision and Challenges for Realising the Internet of Things. Brussels: Publications Office of the European Union.
- ZigBee Alliance, (2008). ZIGBEE SPECIFICATION San Ramon, United States. ZigBee Document 053474r17.[2028?]ZigBee Alliance (2012).

ZigBee Light Link Standard. San Ramon, United States. Version 1.0, ZigBee Document 11-0037-10.

- ZigBee Alliance (2012). ZigBee Light Link Standard. San Ramon, United States. Version 1.0, ZigBee Document 11-0037-10.
- ZigBee Alliance, (2013a). ZIGBEE HOME AUTOMATION PUBLIC APPLICATION PROFILE. San Ramon, United States. Revision 29, Version 1.2, ZigBee Document 05-3520-29.