Forensic analysis of home automation systems

Thomas Mundt[1], Andreas Dähn[1], and Hans-Walter Glock[2]

[1] Department of Computer Science, University of Rostock, Germany
[2] Department Material Science and Medical Engineering, University of Rostock, Germany
thomas.mundt|andreas.daehn|hans-walter.glock@uni-rostock.de

Abstract. Modern buildings are often equipped with universal bus systems. The purpose of these bus systems is to control functions of houses such as lighting, climate control, and heating. In this paper we present a case study on investigating such a system under a forensic focus without direct physical contact (over the air). The purpose of this paper is to demonstrate the entire forensic process of gathering digital evidence, reconstructing data, analyzing digital evidence, and drawing conclusions from the data. We show in detail which data are available in a typical installation of the widely adopted bus system KNX. This paper demonstrates the impossibility of securing a KNX installation in a building with publicly accessible parts.

Keywords. Home and building automation, KNX, digital forensics, privacy, installation bus, spying.

1 Introduction

Home automation bus systems – also known as installation buses – offer large possibilities to control a variety of house functions in a comfortable manner. The purpose of these bus systems is to control functions of houses such as lighting, climate control, and heating. Instead of simply switching on a light, data has to be transferred via a network. These data allow conclusions about the behavior of residents. This may raise privacy concerns for users of such systems on one side and great curiosities for any kind of investigators on the other. This paper helps to understand privacy concerns as well as it shows how investigators could gather a variety of data from the network.

KNX is one very popular type of installation bus and deployed in many public buildings. We use it as an example, but most conclusions are valid for other installation buses as well. KNX uses a standardized network protocol to control electrical installations. KNX devices send single telegrams whenever an event occurs. In a typical installation, a large variety of sensors and actors are connected to the bus, which means the bus is easily accessible at many spots in the building. Telegrams are not encrypted while they travel through the entire network, making them comfortably available for wiretapping. Basically, sensors send telegrams to particular actors or a group of actors without any central control facility.

We have conducted a preceding study [15] showing the feasibility of tracking people who walk along corridors. We utilized data mining techniques based on sensor data originating from light switches and presence detecting sensors. We have also shown how to draw inferences from those tracking data about the social behavior of people.

Several data are relevant for a forensic analysis of home automation systems. These are:

- A list of devices and the type of device (eg. “switch”, “temperature sensor” etc.)
- Topology of the network (position of sensors in the building and cabling)
- Configuration and addresses of devices (a unique address is assigned to each device)
- Live data (eg. commands to switch on the lights or indicating presence of people)

In this case study we demonstrate step-by-step how we gathered these information in the building of the Institute of Computer Science at the University of Rostock. We are in no way privileged users with elevated rights. We use techniques available to visitors and behave like visitors to demonstrate the forensic “anatomy” of the building.

2 How KNX works

KNX is a standardized network protocol for controlling devices in electrical installations within commercial and residential buildings. The KNX standard is maintained by the KNX Association and was approved by several standardization organization, such as ISO (ISO/IEC14543-3) and ANSI (ANSI/ASHRAE 135). A deeper overview about the capabilities of KNX can be found in [8] and [14].

2.1 The physical layer

KNX supports different media. Twisted pair (TP) wiring is most frequently used. Other media are powerline (PL), radio (RF), infrared (IR), and Ethernet. On twisted pair cables the bit rate is defined to be 9600 bit/s. Devices have to be wired without building loops. Exempt this, topology is irrelevant and can be any type of tree. The following participants are the building blocks of a KNX/TP bus:

- Bus devices - any actor or sensor which is connected to the bus.
- Line couplers - a device on the network layer that connects two lines of the bus - a more detailed introduction regarding the structure of the network follows in section 2.3.
- Line repeaters - a device on the data link layer connecting two electrical segments of one line and splitting a line in separate collision domains.

- Two wire cables - used for signalling and to supply devices with a rated voltage of 24V DC.
- Power supplies

The standard defines limitations on capacitance, resistance, voltage, currents etc. This has consequences for bus line lengths, cable lengths between power supply and devices, number of devices per electrical segment, and cable types. An overview can be found in [10]. Noteworthy are the following parameters:

- A total capacitance of 100nF max. is allowed on one segment without bus devices, line coupler, line repeater
- The maximum bus line resistance between power supply and bus device, line coupler or line repeater is 25Ω, and 50Ω between two bus devices, line couplers or line repeaters.
- The maximum bus line length of a segment is 1000m, the maximum bus line length between two devices is 700m - note that longer segment lengths will also prevent collision detection.
- The maximum number of bus devices of an electrical segment is 64 - note that lines can have up to 256 devices when line repeaters are used between electrical segments.
- The cable has minimum 5 twists per meter and 0.8mm diameter. KNX certified cables carry two pairs.
- The power supply nominally supplies a rated voltage of 29V DC to the bus, allowing a 5V drop due to resistance.

Signals are transmitted symmetrically on the two wires, which limits the influence of interferences. A logical '0' is represented by a voltage drop of about 5V which will result in a back swing. A logical '1' is transmitted when the voltage remains constant - note that there are constantly '1' bits when the bus is idle. Figure 1 shows the bit representation.

2.2 Data link layer

KNX devices send telegrams. There are three general types of telegrams, *Data Frames*, *Poll_Data Frames*, and *Acknowledge Frames*. *Data Frames* are utilized to send regular commands and transmit data, *Poll_Data Frames* can be used to gather information from up to 15 devices, and *Acknowledgement Frames* are used to indicate the correct transmission of telegrams by the receiver. Telegrams are split into characters. Each character starts with a startbit, contains 8 databits, a parity bit, and a stopbit. There is an inter-character space of two bit-lengths between each character during which the bus is idle, which could be read as two 'ones'. The physical broadcast domain spans one electrical segment.

Except *Acknowledgement Frames* all telegrams consist of at least 6 characters (bytes) - one control field (1 byte), two bytes for encoding the source address and two bytes for encoding the destination address, and one 8-bit check field. The structure of a telegram is shown in Figure 2.

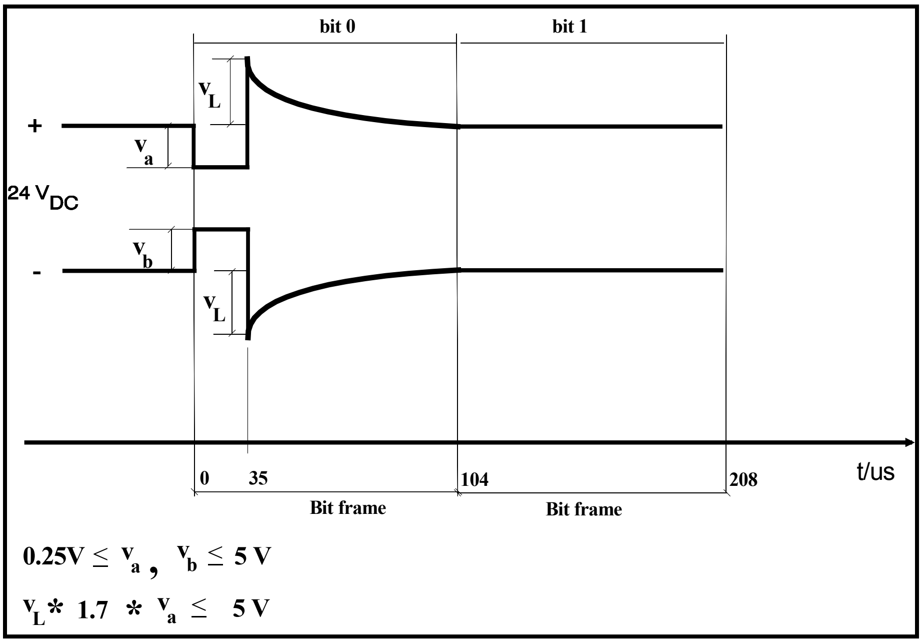
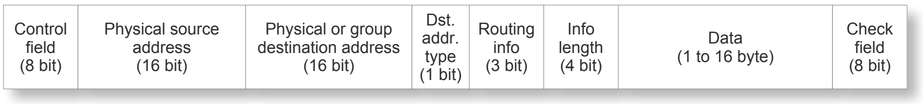


Fig. 1. KNX Twisted Pair bit representation (from [10]).



Control field (8 bit)	Physical source address (16 bit)	Physical or group destination address (16 bit)	Dst. addr. type (1 bit)	Routing info (3 bit)	Info length (4 bit)	Data (1 to 16 byte)	Check field (8 bit)

Fig. 2. Structure of an KNX data telegram.

The check field contains the bit-wise parity over all data fields (including control field and addresses). As there is also one parity bit for each character, a cross check matrix can be used for error detection.

There is no clock signal on the bus, hence a long sequence of ’ones’ could result in timing issues. All characters start with a ’0’ as startbit, the first character of a telegram ends with two ’zeros’, leading to three ’zeros’ at the beginning of each telegram on the wire – KNX TP uses “Least Significant Bit First” (LSB).

The twisted pair medium is accessed in a Carrier sense multiple access with collision detection (CDMA/CD) scheme [12], each device waits for an idle bus before beginning to transmit. If a collision is detected a randomly delayed retransmission is initiated, see also [11] and [5].

Telegrams contain a 16-bit physical address as source address and 16-bit physical or group addresses as destination address. Group addresses are indicated by a flag. Physical addresses consist of area (4 bit), line (4 bit), and device (8 bit). A physical address will be written eg. 3.2.12 (area 3, line 2, device 12). In most cases a line represents an electrical segment of the network. In some rare cases up to 4 segments are coupled by repeaters. Note that repeaters also generate *Acknowledgement Frames* when forwarding a telegram. Lines can contain up to 256 devices (64 devices per segment due to the limitation of power supply).

2.3 Network layer

In most scenarios, line couplers and area couplers filter telegrams according to configured rules in order to limit collision domains and, hence, to reduce network utilisation. Depending on the filters set in couplers, telegrams will be distributed throughout the network.

Figure 3 shows a line coupler among other devices.

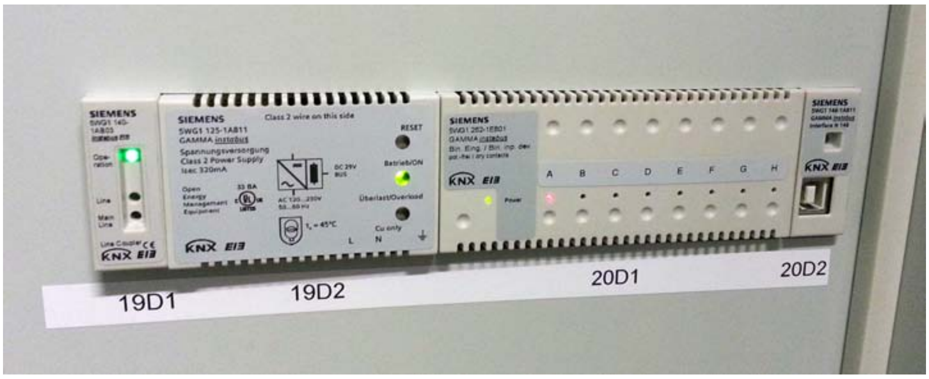


Fig. 3. A line coupler (19D1) beside a power supply (19D2), a generic binary input device (20D1), and a KNX-USB gateway (20D2).

Up to 15 lines can be assembled into an area. One additional line (main line) then works as cross link. Line couplers are also counted as devices in each line. A network can have up to 15 areas. Destinations can be addressed by physical addresses (unicast) or group addresses (multicast). Group addresses are structured as main group (4 bit), middle group (3 bit), and sub group (8 bit) and are written as 1/1/74 for example. A two layer-address is also specified. Devices can be configured to listen to certain group addresses. A broadcast can be addressed by sending a telegram to the group address 0/0.

As mentioned before, loops are not permitted in a KNX TP network. In a tree structure routing is unsurprisingly simple. Based on his filter a line coupler can decide if it forwards a telegram to the other side. A routing counter in all telegrams allows limiting the number of hops.

Figure 4 shows a part of an KNX network with physical device addresses and several group members of one specific group which spans several lines.

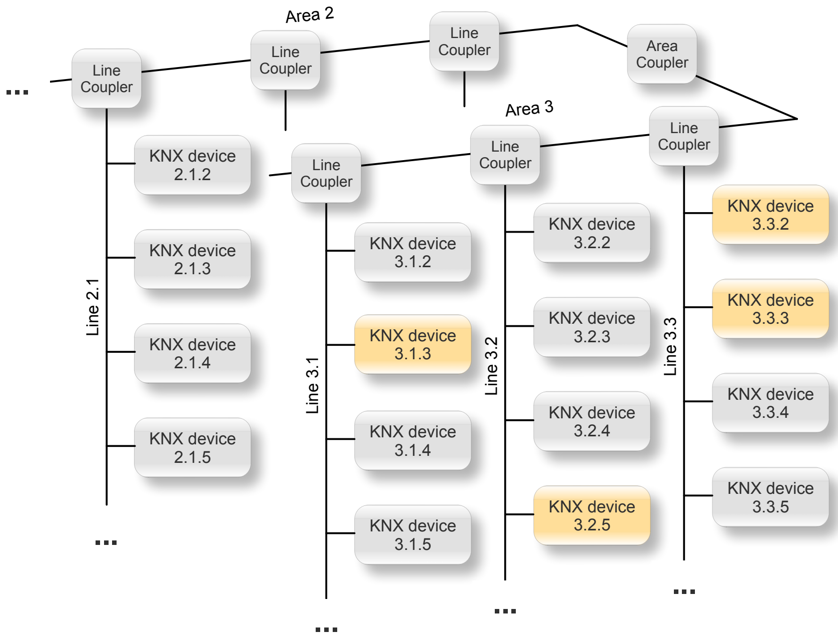


Fig. 4. Example for the structure of a KNX network showing two areas with several lines and several devices per line. As an example, colored devices are configured to listen to group address such as 1/1/123.

2.4 Transport layer

KNX supports a connection less mode and connection-oriented communication. In connection-oriented mode a sending device requires a positive acknowledgement of each telegram. On reception of a NACK message rejected telegrams (negative acknowledgement) are repeated up to three times. The connection is closed if neither an ACK nor a NACK are received before timeout [10].

2.5 Application layer

To avoid confusion about data types, formats, encodings, dimensions, ranges, and units the KNX standard defines *Datapoint Types* as a combination of a data type and a dimension. The list of standardized *Datapoint Types* can be found in [3]. Figure 5 shows how a *Datapoint Type* is generally defined.

Any *Datapoint Type* thus standardizes one combination of format, encoding, range and unit. Figure 6 gives an example and shows how a date is being encoded in 5 bytes.

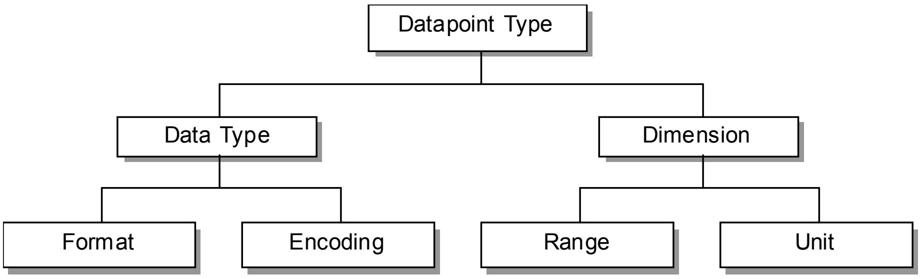


Fig. 5. *Datapoint Types* consist of *Data Type* and *Dimension*. (from [3]).

Octet 6								Octet 7								Octet 8								Octet 9								Octet 10							
						APCI										r	r	r	Day					r	r	r	r	Month				r	Year						
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI																														
						0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0
																			18									12					2006						

Fig. 6. December 12, 2006 encoded according DPT_Date (Octets 8 to 10) in an A_GroupValue_Write-frame (from [3]).

3 Data acquisition

As we noted in the introduction, different data from many sources might become relevant for a forensic analysis. These data are basically a list of devices and the respective type of device, the network topology, configuration and addresses of devices, and live data from telegrams. In this section we exemplarily show how any of these data can be gathered. Once again, we use the KNX bus in the building providing home to the Institute of Computer Science at the University of Rostock as example.

3.1 Acquiring a list of devices

There are several possible sources for a list of devices. The easiest way to get this information would be having a file. Typically, complex KNX installations are planned by an electrician with a software called ETS [9] provided by the KNX association. With this application, devices are parametrized. Each device is assigned an address. Most devices need to be configured to send telegrams to certain destination addresses or to listen to certain group addresses. In order to avoid duplicate addresses and in order to later replace defected devices with the original parameters, ETS needs to maintain the entire configuration. We don't know an alternative software, hence, there is a very high chance, that somewhere an ETS project file exists with all configuration parameters and the structure of the entire network. So, basically, this file contains all information of interest – except live data.

If a configuration file is not available, listening to the KNX bus over a longer period of time and recording source addresses will also provide a list of devices. This list is possibly incomplete, since some devices may not transmit any telegram during recording time. Looking at destination addresses does not unveil many devices in a typical installation, since device destination addresses are very rarely used. Instead, group addresses are much more frequently used as destination address. In our experiment, we did not receive any device address as destination during several weeks of recording.

As we mentioned before, telegrams can be received at any accessible point of the line when Twisted Pair cable is used for the physical layer. Line couplers can be configured to filter telegrams by their destination addresses. In our experiment, we were able to read telegrams from an entire area, since line couplers between the lines in that area were configured not to filter any telegrams, hence, to permit all telegrams regardless of destination addresses.

3.2 Figuring out the type and configuration of devices

KNX devices can be queried for their device type. This requires sending a telegram to each device. Devices answer with their unique device identification number and a value being unique for each type of device. In most cases it is not necessary to actively query devices.

First assumption about the type of device can be made when listening to live data and interpreting the data types being used. A standard light switch, for instance, will send telegrams with values of a binary data type - see section 2.5 for a description of *Datapoint Types*.

All kind of devices, namely couplers, sensors, and actors have to be configured during the process of installation. Line couplers do not participate actively in the exchange of telegrams, but filters have to be set. For sensors, a unique device address has to be maintained, which is consequently being used as source address. Telegrams are sent by sensors and typically received by at least one actor. In most cases, a group of actors is addressed. The destination address is programmed into the sensor device.

Routing happens in a static manner, which means that all paths have to be configured at installation time. Routing is implemented in a very simple way: Basically, line couplers can be configured to allow the passing of certain telegrams according to their destination addresses. Packets with other destination addresses than those permitted will not be forwarded beyond the broadcast domain formed by a line.

Listening to live traffic for a while will certainly unveil a more or less complete list of devices and destination addresses configured in those devices. See section 3.4 for details about capturing telegrams.

3.3 Learning about addresses and positions of devices

The logical topology of a KNX network consists of areas, lines and devices (see section 2.3). This is not necessarily associated with the position of devices. When planning a KNX network, an electrician has to observe some constraints, such as possibilities to place cables, length of cable, number of devices per line, distance between devices, and estimated number of telegrams per line. Under those circumstances it is not expectable that the physical structure of the network and its logical structure are particularly similar to each other in terms of neighborhood and consistency of addresses.

In the concrete case of our experiment, we assume that the installation was mainly influenced by the idea of connecting each device to the nearest available cable. Hence, in order to later draw conclusions about the behaviour of subjects, we needed to determine the position of relevant sensors.

3.4 Listening to live traffic

KNX does neither implement authentication nor privacy measures, making it widely available for forensic research – or hackers. For our preceding study we have utilized a KNX-IP-gateway which we connected to a two wire twisted pair cable. This required at least the removal of a cover plate from a switch. All telegrams on the line connected to the KNX-IP-gateway can be easily stored in a file or database for later analysis.

As we described in the preceding sections, captured telegrams tell information about the presence of sensors – which are sending devices – and destination group addresses. Telegrams give some hints about the type of sensor device and together with visual observations tell about the position of sensors. A sample floor plan which easily could be taken from public emergency rescue displays, is depicted in Figure 7. The entries in this floor plan could be the result of listening to traffic and correlating events with visual observations.

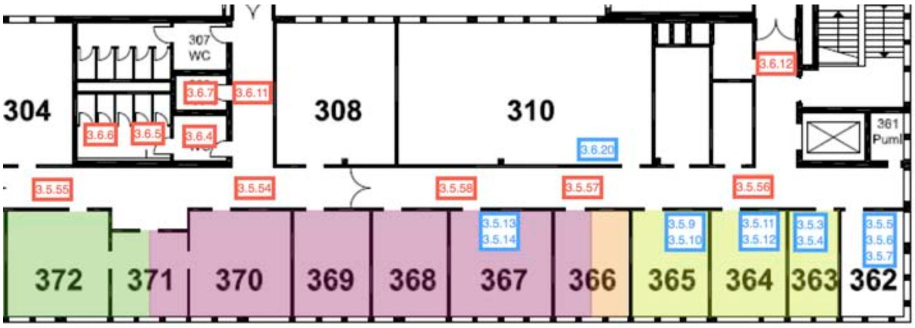


Fig. 7. Sample floor plan with addresses of some sensors. Infrared presence detectors are marked in red, switches in blue.

In this consecutive study we demonstrate step-by-step how we collected data from a KNX bus without attaching a cable to the bus.

A live forensic analysis can be performed by intercepting telegrams at any point of the line for traffic in that particular line. Depending on the filters which set on couplers, traffic from other lines and areas might be also perceptibly. Software libraries to interact with KNX networks are available for all popular programming languages. We used Calimero, an open source framework from the University of Vienna [13][6], to save telegrams in an SQL database.

4 Example investigation

We use an example for demonstrating the basic working principles of KNX and show how to gather a variety of information. In our scenario a building very, very similar to the building of the Institute for computer science of the University of Rostock is equipped with a KNX bus system. The following functions are controlled over the bus system:

- Lighting. All light switches and lamp controllers are connected to the KNX bus. Infrared presence detection sensors are installed in all corridors for automatic light control.
- Heating and air-conditioning. Several thermostats and air-condition units in laboratories are remotely controlled via KNX. Sensors measure the temperature in almost all rooms. A few ambient sensors report weather conditions.

Figure 8 shows some of the devices which are visible to visitors.

4.1 General course of action

The only precondition for intercepting data wirelessly from the bus is the ability to get close to the cable -– up to 5cm with the tested equipment — or any installed device. We tried several coils to receive suitable impulses from the cables within a wall. KNX on Twisted Pair cables uses 9600 baud (see section 2.1). As the resulting frequency is well within the range of audible frequencies, an audio pre-amplifier and a notebook for recording proofed more than suitable for our purpose (see Figure 9).

Utilizing audio equipment lowers the costs for an attacker dramatically. We were able to intercept telegrams from various devices on the line. Figure 10 shows the audio signal as it has been recorded with the described equipment. The signal noise ratio is superb, and, hence, all bits can be surely detected.

For a short first investigation, we saved the samples in a WAV-file [2] for later decoding. In a second phase we implemented an application on an embedded board and a sophisticated sound card with pre-amplifiers. We decided for a Raspberry Pi [16] embedded board with Raspbian [1], a Debian descendant, as operating system and a Focusrite Scarlett 2i2 [7] as audio interface.

A down-sampled wave file can be downloaded at http://tinyurl.com/knxless. For decoding, we use a Python program (see code snippet in Listing 1.1), which detects spikes in the audio file, recovers the bit stream and detects telegram

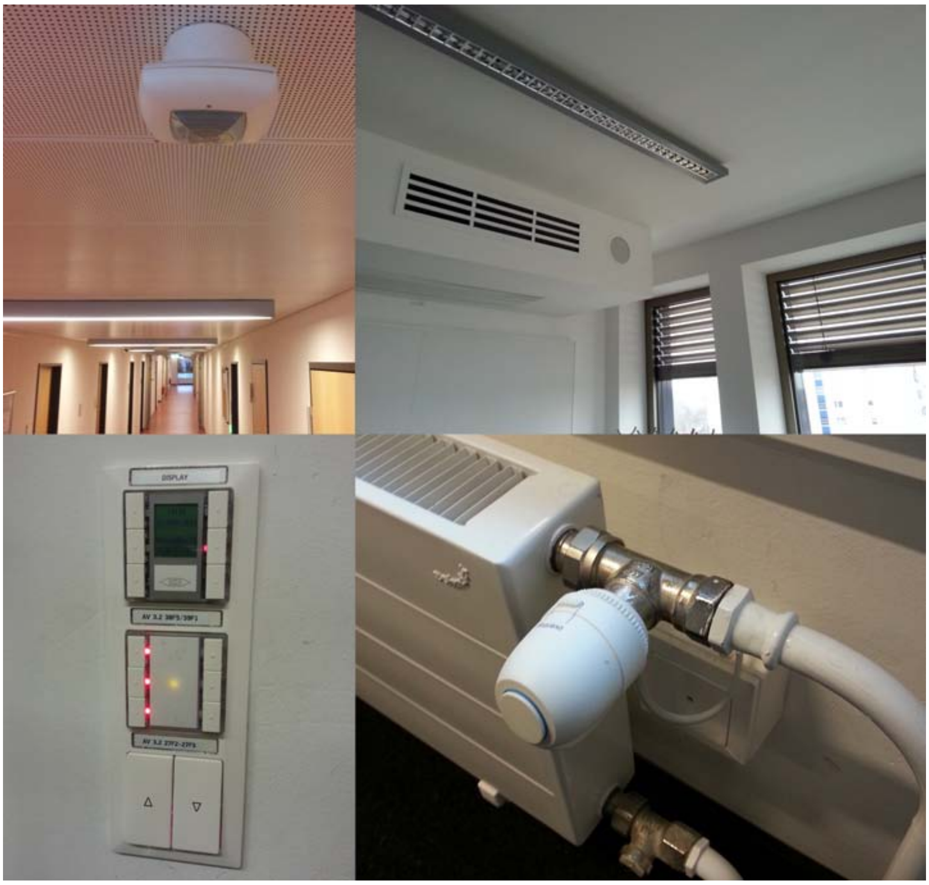


Fig. 8. Visible devices connected by KNX. Top left: Infrared presence detector. Top right: Sun shades, air conditioning and lamps. Bottom left: Control unit for heating / air-conditioning, light and sun shades. Bottom right: heater thermostat.

boundaries. In a second step we extended the software program to decode the entire telegram, to check the protocol constraints and to assemble and send a KNX-IP message for each KNX telegram. With this feature the embedded board has the same functionality as the commercial KNX-IP-gateway we used in our first experiment.

```
import struct
f = open('Sample.wav', 'rb')
f.seek(46)
sample1 = sample2 = sample3 = lasti = 0
decoded = '0'
while True:
  sample1 = sample2
  sample2 = sample3
```

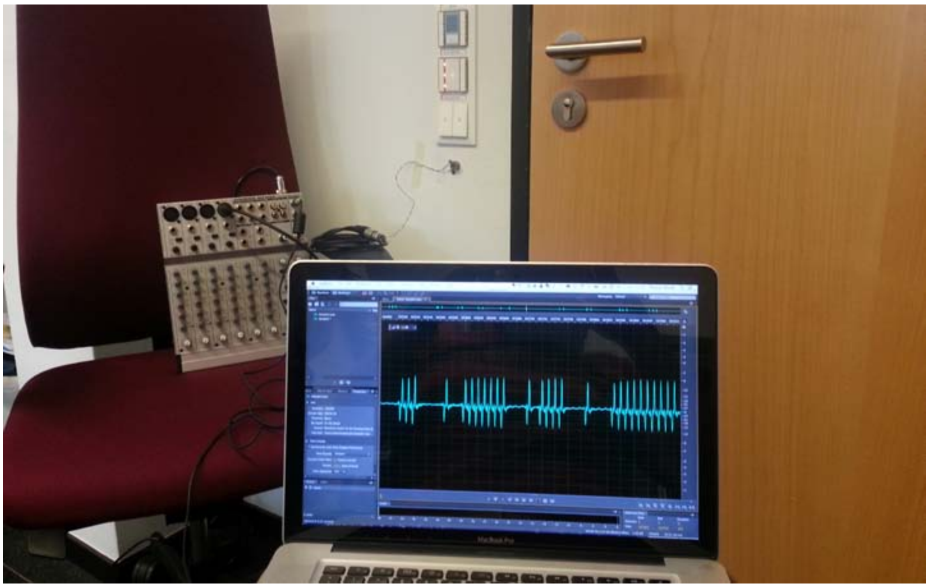


Fig. 9. Test setup with coil next to the wall switch, audio mixer used as pre-amplifier, and a notebook with audio input and recording software.

```
  buf = f.read(4)
  if not buf: break
  sample3 = struct.unpack(’<f’, buf)[0]
  if sample1 < sample2:
    if sample2 >= sample3:
      if sample2 > 0.1:
        samplessincelast = i - lasti
        if samplessincelast > 155:
          print ’Full telegram:’, decoded
          decoded = ’0’
        else:
          num = (samplessincelast+5)/10
          if num > 1:
            for j in range(1,num):
              decoded = decoded + ’1’
          decoded = decoded + ’0’
        lasti = i
f.close
```

Listing 1.1. Code snippet illustrating the principle decoding of KNX telegrams from a WAV-file.

We are able to surely decode all telegrams and collect source and destination addresses as well as data units. For testing purposes we collected data simultaneously with a commercial KNX-IP-gateway and our equipment. The commercial

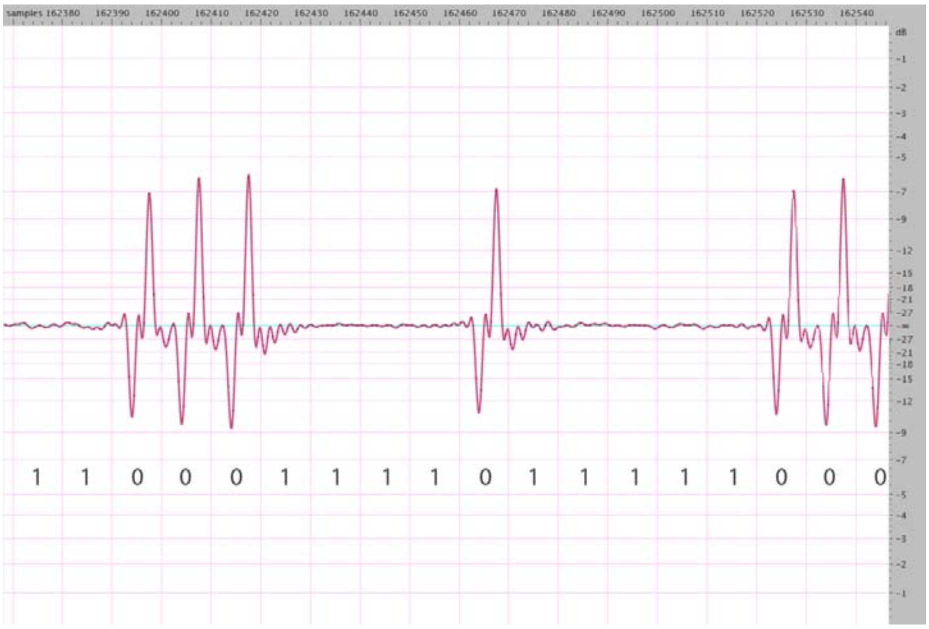


Fig. 10. Wave form as recorded and decoded bit values. Note that KNX on two wire Twisted Pair cables uses a constant voltage to indicate a logical "One" and a "peak" to indicate a logical "Zero". There is no clock signal, but the protocol guarantees frequent "zeros" for synchronization.

KNX-IP-gateway was directly connected to the bus via an attached cable. Our own hardware was connected only electromagnetically via a coil and was running our own software. Both devices delivered the same KNX telegrams for the entire testing period of a week.

A coil taken from a small size computer speaker produced the best results as an electromagnetic loop antenna. We dismantled the magnetic core as an air loop antenna promised better results. A loop antenna is sensitive to the magnetic field and not the electric field. It induces a voltage which depends on the strength of the magnetic field and which – in our concrete case – can be read by a computer's sound card interface. Refer to [4] for more information about antenna theory. Figure 11 shows the antenna we used throughout our experiments.

We repeated the experiment at many different spots where we assumed the KNX cable to be flush-mounted underneath the drywall. Good chances are below or above light switches. We found the cable at many locations and were able to read the data in most cases. The exact positioning of the coil is dispensable as long as we could get in 10cm proximity. We used headphones to detect good spots.

We can conclude that intercepting KNX traffic without physically interfering with any installed devices is possible and can be performed with very limited

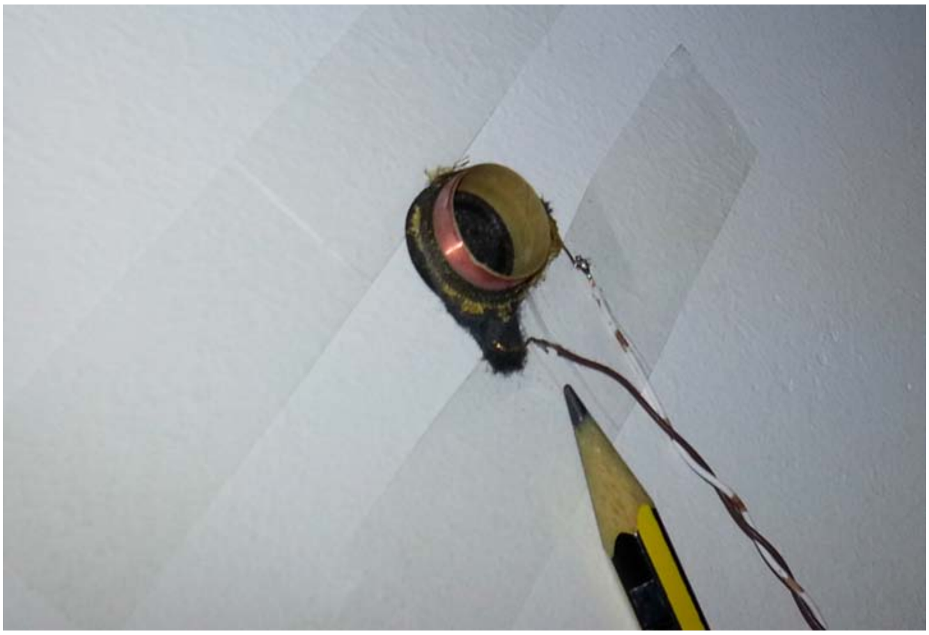


Fig. 11. A coil taken from a loudspeaker. The magnetic core has been dismantled to build an air loop antenna.

efforts. This imposes a high risk for both, privacy of users and security of buildings, as an attacker could utilize information he gathered from the bus for his own purposes, such as spying, getting warned before someone arrives or learning about the security procedures in a building.

In order to estimate the risk imposed by the attack shown in this paper we summarize the costs as follows:

- A loop antenna taken from a computer speaker - 0 to 5 EUR.
- A Raspberry Pi embedded board - 25 EUR.
- A sound card of sound interface - 5 EUR to 139 EUR, the latter including a pre-amplifier.
- About a week to learn about the position of sensors.
- About a day to understand the KNX protocol and to write a decoding program. Can be reduced to no time by finding this paper.

4.2 Specific characteristic in our example

While the preceding section 4.1 introduces general ways on how to intercept live traffic from a KNX bus this section gives more details about the concrete situation in our example. Although, not all information given in this section might be generalizable, it still illustrates obvious security and privacy aspects.

In our case the KNX bus was easily accessible in several ways. We could open the cover plate of a light switch and attach a KNX-IP-gateway to the two wires of Twisted Pair KNX. This would leave at least some traces.

As shown in Figure 12 sometimes the installation crew provides "free home delivery" of KNX telegrams by leaving USB gateways.

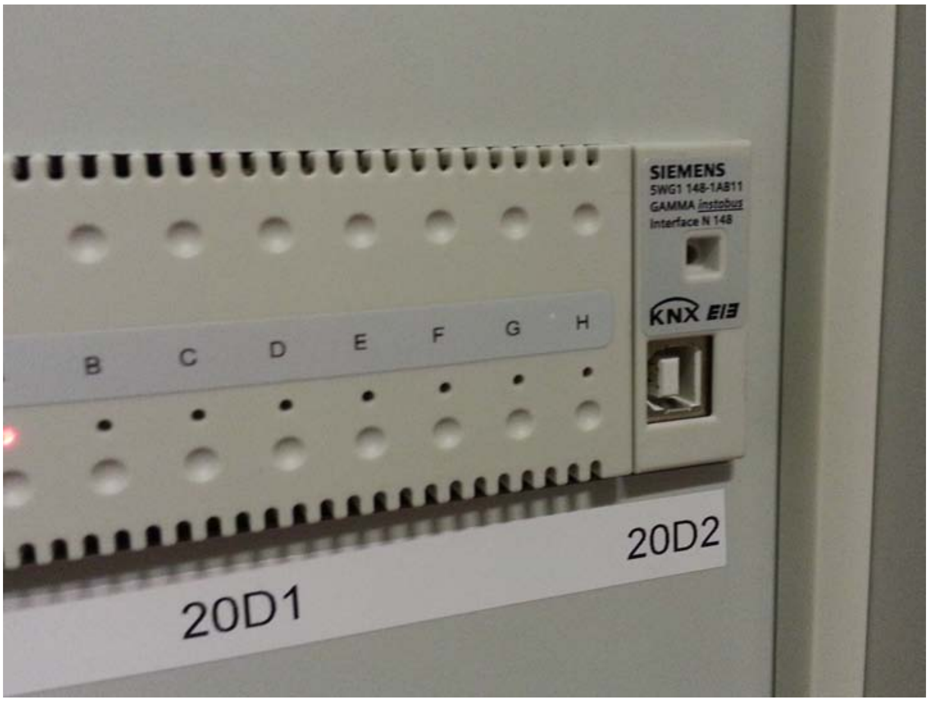


Fig. 12. KNX-USB-gateway installed in a switchboard. Obviously, this gateway was once installed for the ease of initial configuration and later maintenance.

In our concrete case it was argued by the University's administration, that these USB gateways are installed in rooms only accessible to a few authorized employees. This also proved to be wrong, as shown in Figure 13. The electrical cabinet shown in this figure is installed in a room which is accessible by several "ordinary" employees.

And finally, as a special service for brave researchers, the standardized key for the electrical cabinet is provided using a highly sophisticated attaching device which "definitely" guarantees access by authorizes personnel only. The cabinet case itself also contained schematic diagrams with all installed devices in a larger part of the KNX network.

5 Conclusion and outlook

In a preceding study we have already shown how data mining techniques can be used to easily infer information about the social behaviour of people – in that particular case we used the hand washing behaviour of people as an admittedly dramatic example to emphasize on privacy aspects.

In this paper we have shown that KNX is insecure by design. We have also shown that preventing physical access to installations does not help maintaining a minimum level of security. We demonstrated that a complete forensic analysis of an entire KNX network can be done at a very limited price with devices and know how available to the general public.

Beside our own privacy concerns, other attacks seem to be possible without much effort. A thief, for instance, could reconnoiter hotel rooms for absent guests.

We are currently planning another experiment which aims on sending telegrams using an electromagnetic loop instead of only receiving.

References

1. The Raspbian homepage. http://www.raspbian.org, http://www.raspbian.org
2. Multimedia programming interface and data specifications 1.0 (August 1991)
3. Knx standard - system specifications - interworking - datapoint types (04 2010)
4. Balanis, C.A.: Antenna theory: analysis and design. John Wiley & Sons (2012)
5. Cavalieri, S., Cutuli, G.: Proposal and evaluation of deterministic access in knx standard. In: Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on. pp. 1674–1679. IEEE (2008)
6. Erb, B., Neugschwandtner, G., Kastner, W., Kögler, M.: Open-source foundations for pc based knx/eib access and management. In: Konnex Scientific Conference (2005)
7. Gopman, D.B., Bedau, D., Kent, A.D.: Audio cards for high-resolution and economical electronic transport studies. arXiv preprint arXiv:1202.5601 (2012)
8. Kastner, W., Neugschwandtner, G., Soucek, S., Newmann, H.: Communication systems for building automation and control. Proceedings of the IEEE 93(6), 1178–1203 (2005)
9. KNX Association: ETS4 Professional, http://www.knx.org/knx-tools/ets4/ets4-professional/
10. KNX Association: Serial Data Transmission and KNX Protocol, knx tutor seminar edn.
11. Kyselytsya, Y., Weinzierl, T.: Implementation of the knx standard. In: Tagungsband KNX Scientific Conference November (2006)
12. Lam, S.S.: A carrier sense multiple access protocol for local networks. Computer Networks (1976) 4(1), 21–32 (1980)
13. Malinowsky, B., Neugschwandtner, G., Kastner, W.: Calimero: Next generation. In: Proc. KNX Scientific Conference 2007 (2007)
14. Merz, H., Hansemann, T., Hübner, C.: Building automation: communication systems with EIB/KNX, LON und BACnet. Springer Verlag (2009)
15. Mundt, T., Kruger, F., Wollenberg, T.: Who refuses to wash hands? privacy issues in modern house installation networks. In: Proceedings of the 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications. pp. 271–277. IEEE Computer Society (2012)

16. Upton, E., Halfacree, G.: Raspberry Pi User Guide. John Wiley & Sons (2013)

Fig. 13. Electrical cabinet containing the switchboard shown in Figure 12.