Short Paper: Extrapolation and Prediction of User Behaviour from Wireless Home Automation Communication

Frederik Möllers*, Sebastian Seitz, Andreas Hellmann† and Christoph Sorge*
*Saarland University
* {frederik.moellers|christoph.sorge}@uni-saarland.de, † kontakt@anhellmann.de

ABSTRACT

Wireless home automation systems are becoming increasingly popular. They can help users save energy and increase the comfort. However, this increased convenience also comes with new attack vectors. Many available systems provide little to no security. In this paper, we explore the possibilities of passive attacks against these systems. We exemplarily investigate two real-world installations of off-the-shelf home automation systems to see what amount of information can be obtained by a passive adversary.

Our results show that the systems provide no privacy. They leak information about the users' habits as well as their presence and can be abused to plan burglaries. Furthermore, we conclude that even encrypted communication does not fully protect against the attack presented here. In particular, it is still possible to predict user presence and absence even if individual actions cannot be identified.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Unauthorized access*

Keywords

Privacy; Home Automation; Traffic Analysis; Wireless Networks; Profile Building

1. INTRODUCTION

In recent years, home automation systems (HASs) have become affordable and thus very popular with private users [15]. While providing increased comfort, this also introduces new attack vectors, especially with the increasingly popular wireless systems [8]. Therefore it is necessary to consider the security regarding attacks against both the functionality of the system and the privacy of the inhabitants.

In this paper we present our security analysis of real-world HASs, using HomeMatic installations as an example. We focus on passive attacks targeting the system's communication to create user profiles and predict user behaviour. Our passive attack consists of 3 consecutive stages. First, data is collected by sniffing the wireless communication. Semantic information is extracted from this data, such as sensor readings or actuator commands. As a second step, the attacker identifies patterns in the data, either from automation rules or from routine user behaviour. In the third step, the attacker then predicts the user's habits.

We also examine how much information can be obtained by performing *traffic analysis*, where the adversary uses timing patterns and metadata to draw conclusions about the users of the system. The information from both kinds of attacks can be used, for example, to plan a burglary when nobody is at home.

In addition to the burglary itself, insurance companies might refuse to pay the damage, given that the information was readily available from the inhabitants' own system. Thus, unsecured HASs can also be a monetary danger.

We conducted our analysis on two real-world installations of the HomeMatic HAS. To make the scenario realistic, we did not have any additional information about the two installations prior to the attack other than the fact that they were HomeMatic HASs. In the case of the second system, we were informed that it consisted of two parts that were installed in different locations and connected via a VPN.

The remainder of this paper is organized as follows. First we will provide some background information on the topic and have a look at related work in this area. Then we explain the methods we used in our analysis in Sect. 3 and present our analysis in along with its results in Sect. 4. We conclude with Sect. 5 and provide an outlook into future work.

2. BACKGROUND

For more than a decade now, technologies exist that enable the implementation of HASs. The type of devices in a home automation (HA) network varies from simple power switches and temperature sensors to smoke detectors and door locks. Main benefits are efficiency and comfort, but HASs can also be used or extended for security purposes. Light-controlling movement detectors can detect burglars, too, and supplementary intrusion detectors can use the existing network architecture to call the police.

2.1 Fundamentals of Home Automation Systems

Local communication technologies for HASs can be coarsely divided into wired (e.g. BACnet) and wireless (e.g. BidCos). There are also standards that cover both (e.g. KNX).

WiSec'14, July 23–25, 2014, Oxford, UK.

ACM 978-1-4503-2972-9/14/07 ...$15.00.

A major difference between wired and wireless HASs is the installation effort which disqualifies the former from installation in certain (especially rented) properties. However, traffic from a wireless system can be intercepted and possibly manipulated from anywhere nearby. Thus, protection against network attacks becomes much more important.

The topic of HA is closely related to that of building automation. Since building automation systems are usually wire-based and used in public buildings, they do not focus on privacy protection and have not been analyzed by us.

Currently, there are several HASs available on the market, e.g. HomeMatic, EnOcean and Siemens Synco Living. Our choice fell on the HomeMatic system. Due to its wide availability, the potential number of volunteers was large. In addition to this, hardware and open-source software exists that allows the capture of the communication and thus could be used for our experiments.

2.2 Legal Considerations

Gathering data for our analysis turned out to be problematic: only two volunteers gave us permission to analyze the traffic of their wireless HASs. A number of researchers have performed "War Driving" experiments, which essentially means getting information about a large number of WiFi networks without having asked for permission. There is, however, a legal difference to HASs—at least in Germany, where our research has been conducted. A WiFi network's beacon frames are generally broadcast messages, directed at anyone in the vicinity, to allow distinction between different networks. Data collected during War Driving might (partly) be considered as personal data, which is protected under European and national law. However, research purposes can justify the collection and processing of data under certain circumstances.

The messages exchanged by a wireless HAS, on the other hand, are only meant to be received within that system. From the perspective of legislation, the same limitations apply as in case of WiFi networks. In addition, collecting data from non-public transmissions is also a criminal offense according to section 202b of the German criminal code. We therefore decided against this option.

2.3 Related Work

There has been extensive research in the area of home automation in the last decade but very few have taken the aspect of secure communication into account.

Al-Muhtadi et al. proposed a very early approach [1] based on Jini and Tiny SESAME, a stripped down version of SESAME, which itself is an extension to Kerberos. Their main goals are authentication and access control and they do not consider passive attacks.

Bergstrom et al. presented an approach to secure home automation communications in 2001[2]. They assumed HA networks being controlled via the Internet using a so-called Global Home Server (GHS). The approach only focuses on securing the communication between the GHS and the individual networks, whereas the local communication is not taken into account.

Marin et al. [11] developed a middleware for home automation systems which relies on TLS for inter-node communication. They introduce different authentication procedures and encryption to secure data transmissions, but do not consider leakage of information through side channels.

Wireless sensor networks (WSNs) have similar contraints and requirements as HASs so research in this field can provide additional insight. Several surveys [9, 10, 14, 16] list known problems and solutions. De Cristofaro [5] tackles the problem of privacy protection for a user who queries a WSN. While traffic analysis and other passive attacks are a well-known threat in WSNs, the solutions rely on properties such as multi-hop routing or a local attacker. HASs differ from WSNs in exactly these aspects, so using WSN countermeasures in HASs requires extensive adaptations.

Concerns about traffic analysis in general go back as far as 1981 [3]. Numerous approaches have been developed to protect against this class of attacks [4, 6, 7, 13]. For computer networks, they have proven to even throw back the most powerful attackers [12]. However, similar to WSNs, they leverage properties that do not exist in HASs, such as routing and looser energy constraints.

3. ANALYSIS METHODS

For our passive attacks, we assumed a realistic attacker model: The attacker can observe the whole system at once over longer periods of time, but has no prior knowledge. This was implemented by putting a capture device inside the users' homes without receiving any information about the setup from the owners. While the position of the device is certainly different in a real attack, we assume the same coverage can be reached with multiple devices. The following sections provide an overview of the methods used in our analysis which have been implemented in a toolsuite. The modules are called *sniffer*, *cleaner* and *analyzer*.

3.1 Data Acquisition

A simple and way to eavesdrop on the communication is provided by the so-called CC1101 USB Lite (CUL) stick, using an open-source firmware(called `culfw`) that can decode several wireless HAS protocols. The collection of data is performed by the *sniffer* module. It reads the data and applies regular expressions which are used to preliminarily identify the type and function of each node. They are found in the FHEM (`http://fhem.de`) software. For our studies we attached the stick to a Raspberry Pi which served as a host computer and data storage.

3.2 Data Interpretation

The approach of applying regular expressions from the FHEM software gave us a basic idea of the device categories involved in the communication. In order to achieve maximum clarity about the packet contents, we processed the collected transmissions in 4 steps.

1. The BidCos packet structure allows the distinction of different devices based on their addresses. Counting the number of distinct addresses in the collected packets also gives the number of devices in the network.
2. To clean the collected data, we discarded all packets that were resent due to transmission errors. The remaining packets were saved in a database table whose attributes (columns) correspond to the packet fields (e.g. source address, length). The *cleaner* module performs this task.
3. The classification of devices using regular expressions is not error-free. FHEM relies on packets captured from an initial pairing procedure, which the attacker does not necessarily observe. In order to correct these

errors, we supplemented the preliminary classification with plausibility checks. These checks are explained in the next paragraph.

4. Lastly, we interpreted the messages according to their context. Messages from a temperature sensor, for example, were translated into decimal numbers. The base station, when sending messages, assumed the behaviour of a different kind of device, e.g. it would act as a remote control when sending commands to a light switch. Choosing the method of interpretation according to the destination let us correctly translate this data as well.

Plausibility Checks

As mentioned in item 3 above, the mere layout of a certain message may not clearly indicate whether it is a temperature sensor status response or a command to a window opener. To determine which of the two was correct, we would examine the possible interpretations of the data. A temperature interpretation, for example, which results in a value outside the range from $-25°$C to $50°$C would be unlikely to be correct, since the systems were installed in German homes. These checks can also be automated, but have only been performed manually by us due to lower overall effort.

Another type of check is the inspection of communication links. For example, a remote control is much more likely to communicate with a window opener than with a temperature sensor. Examining the communications partners of a node thus helps to find plausible classifications. However, this would introduce dependencies and probabilities into the identification process. Thus, these checks require either manual intervention or complex logic and the gain of additional information is questionable. As a result, we have not implemented them and have used only minor manual corrections to help in the classification.

3.3 Profile Building

After successfully identifying the device types, an attacker can now build a profile of the inhabitants using the information he gained from the above steps. We performed this analysis in 3 different steps. All 3 steps are included in the *analyzer* module.

3.3.1 *Visualizing the Communication*

As a first step, we displayed the collected data in formats that allow a manual inspection. Two particular visualization formats have proven to be useful for our analysis. On the one hand we created a directed graph out of the collected device data. Each device corresponds to a node and an edge is created between each two devices that ever communicated with each other. The width of the edge is determined by the amount of messages on this link. On the other hand we projected the messages to and from a single device in relation to the time onto a 2-dimensional graph. This graph type helps identify temporal structures and periodic events.

3.3.2 *Correlation Analysis*

In addition to the manual identification of correlated events, we performed an automated correlation analysis using a sliding window approach. We defined an event as a 4-tuple of sender address, receiver address, message type and message content. For each event e, we examined other events e^* that occured in a time frame after e. We then paired e with each of these other events e^* and for each pair (e, e^*) calculated the number of occurences over the whole observation time. 3 parameters allow filtering out events: The minimum total number of occurences of the event e, the minimum chance of e being followed by e^* and the length of the time frame in which e^* has to follow e in order to be counted.

3.3.3 *Filtering Automation Rules*

With a similar approach we tried to filter out programmed automation rules. We assumed that automated events occur at a fixed time which differs only marginally. For each event we collected all occurences over the observation period. We then stripped the date so only the time of day remained. As a last step, we sorted the occurences in chronological order. The sorted list allows for an easy identification of events that often occured at similar times during a day.

4. RESULTS

The following sections present the analysis results.

4.1 Candidate 1 (C1)

The first installation is a simple single home installation. We recorded 45,679 messages over a period of 36 days. Only a few devices could not be identified with acceptable certainty. As we found out after a debriefing with the owner, this was the case for the smoke detectors that only send heartbeat messages to the central unit as well as one of the tri-state sensors that did not send any state changes during the observation period.

4.1.1 *Communication Overview*

Fig. 1 provides a graphical overview of the communication. Expectedly, the central unit *Z 1.1* communicates with most sensors and actuators so it could be easily identified. The graph also allows us to identify which components are directly paired with each other and do not exclusively communicate over the central unit. This information might be useful, for example for later active attacks against the system, and might also be an indicator for manual interaction.

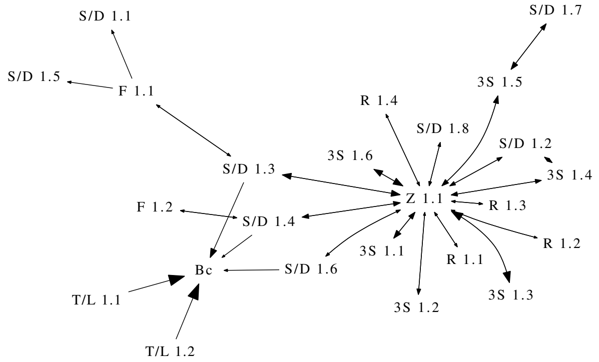

Figure 1: Directed communication graph for candidate 1. The abbreviations used for the device types are explained in Tab. 1.

4.1.2 *Manual Examination of Message Graphs*

Fig. 2 shows the temperature status messages of two temperature/humidity sensors *T/L 1.1* and *T/L 1.2*. What immediately leaps to the eye is the different ranges that the values lie in. We thus concluded that *T/L 1.1* is located outside the house whereas *T/L 1.2* is located on the inside.

Abbr.	Device
3S	Tri-state Sensor
Bc	Broadcast Address
F	Remote Control
KF	KeyMatic Remote Control
KS	KeyMatic Lock
R	Smoke Detector
S/D	Switch / Dimmer
ST	Heating Actuator / Thermostat
T/L	Temperature / Humidity Sensor
Z	Central Unit

Table 1: Abbreviations for the different devices.

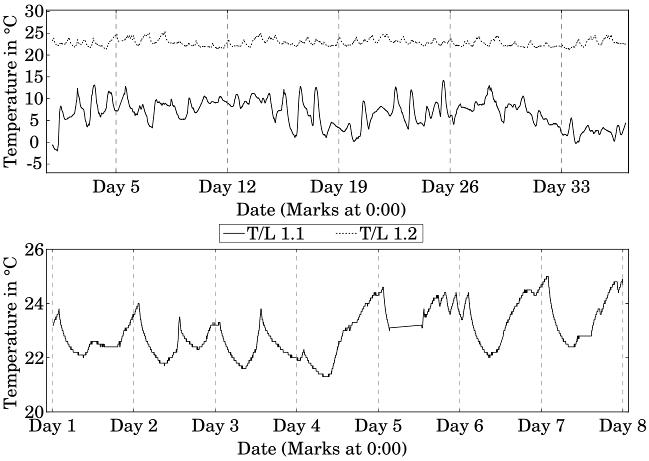

Figure 2: Temperature values of *T/L 1.1* and *T/L 1.2* over the course of 6 weeks (upper) and values of *T/L 1.2* over the course of 7 days (lower).

We confirmed this by comparing the recorded values with weather reports from the area.

Values of the in-house sensor *T/L 1.2* consistently lie in the range between 20°C and 25°C. The not perfectly regular rise and fall suggests that the heating is controlled manually and indicates a user habit. Furthermore we deduced from the low outside temperature and the slow temperature drop inside that the room is seldomly ventilated by widely opening the windows for at least 10 minutes.

The tri-state sensors[1] can be coarsely divided into two groups. The first group consists of only two sensors, *3S 1.2* and *3S 1.4*. Both send only very few messages with usually the same content. No conclusions could be drawn about their role. The second group consists of the remaining tri-state sensors whose traffic mainly consists of *open* and *close* state announcements.

Examining the protocol data for *3S 1.3* and *3S 1.6* reveals that they frequently switch the state. The *open* state is never held for more than 1.5 minutes and usually lies in the order of seconds, suggesting that the sensors are placed on doors rather than windows. The activity over longer time spans shows gaps during the nights and early mornings. Since tri-state sensors notify about state changes usually caused by user interaction, these gaps are good indicators for the inhabitants' sleep cycles. *3S 1.6* changes its state to *closed* some time before the gaps, which supports the assumption that it is installed on the front door. This is a major discovery for an attacker, because he can now tell when the first inhabitant leaves the flat/house in the morning. If there is only one inhabitant, this knowledge is already enough to plan a burglary during the owner's absence.

Similar to the tri-state sensors, the switches/dimmers can be divided into two groups. *S/D 1.2*, *S/D 1.6* and *S/D 1.8* showed very little activity over the observation period. *S/D 1.3* and *S/D 1.4* regularly alternate between *on* and *off* states. The activities of *S/D 1.4* (Fig. 3) revealed a strong regularity in the afternoon between 16:30[2] and 17:00 when the actuactor is switched *on* and at 1:00 when it is switched *off* again. Each day the former action is performed 1.5 minutes earlier than the day before. This is a very strong indicator for an automation rule which compensates for the sunset times. This assumption is supported by the fact that the respective commands come directly from the base station rather than a remote control.

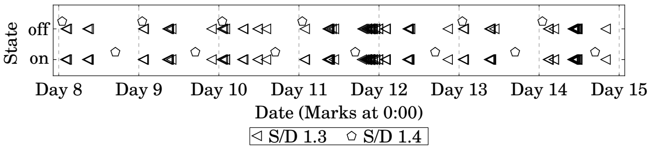

Figure 3: Data sent to and from switches/dimmers *S/D 1.3* and *S/D 1.4*.

Considering *S/D 1.3*, we found regular activity on weekdays between 1:00 and 2:00 as well as between 8:30 and 9:30. The slight variations support the conclusion that this indicates a user habit rather than an automation rule. The payloads of the recorded packets revealed that on weekday mornings, the base station would send timer commands to the switch between 8:00 and 9:15. These commands would turn the switch on for an hour after which it would turn itself off again. We attributed this behaviour to either a habit of the user after waking up or to an alarm function actually waking the user by e.g. turning on the lights.

Another regularity is the absence of activity of *S/D 1.3* between 13:00 and 17:30. The fact that this coincides with a lack of activity of *S/D 1.7* (12:00 to 18:30) lead us to the conclusion that the user is absent during this time of day.

4.1.3 Correlation Analysis

We started the correlation analysis by trying out possible parameter values. Since we had no prior knowledge about the systems, the only approach was to manually determine suitable parameters.

The results of the correlation analysis largely support the previous findings which could already be observed in the graphical analysis. Additional conclusions about the system and the user are elaborated here.

In 72.5% of all cases where sensor *3S 1.6* was turned on, it would be turned off again within 10 seconds. A similar behaviour was observed for sensor *3S 1.4*, which was closed within the 10-second interval in 58% of all cases. In accordance with our reasoning above, we concluded that the sensors are installed on doors rather than windows.

When selectively analyzing the behaviour of *3S 1.3* and *3S 1.5*, we found them to act very much alike. In most cases the state *open* did not hold for longer than 90 seconds. In the case of *3S 1.3*, this was especially interesting when considering the timer commands sent to it by the base station in the mornings. The commands would turn on the switch

[1] The family of tri-state sensors includes different devices: Window sensors that distinguish between *open*, *closed* and *tilted* and door sensors which only distinguish between *open* and *closed*. Technically, they are the same kind of device.

[2] Times in this paper are expressed in the 24-hour notation.

for 300 seconds, but in 96% of these cases, the switch would be manually turned off within the first 90 seconds after reception. This supports our theory that the switch is part of an alarm function.

4.1.4 Filtering Automated Events

In order to filter out automated events, we initially started the analysis with very strict parameters: The minimum number of occurences of an event were set to 120, the maximum overall deviation of events possibly originating from the same automation rule was set to 60 seconds and the maximum deviation of two consecutive events from the same rule was set to 30 seconds. The only event to match at first was a command from the base station which turns off *S/D 1.4* at precisely 1:00, confirming our assumptions. We then proceeded to loosen the parameters to search for other rules. The command coming from the base station and turning on *S/D 1.4* in the afternoon between 16:25 and 17:10 came out next. Although the maximum distance between the different occurences is 38 minutes, we concluded that this event indicates the presence of an automation rule. The distance between two consecutive occurences is about 90 seconds and each event occured later than the one on the day before. Rather than a user habit, we attributed this regularity to an automation rule that incorporates sunset times.

4.2 Candidate 2 (C2)

The installation of C2 was somewhat special since it was split up in two parts that are interconnected via a VPN. One part was the user's private flat and the other part was his office. For this reason, we performed the data collection in two parts. We first installed the sniffer at the office, then moved it to the user's home. During this period we recorded 34,707 messages sent from 20 devices.

4.2.1 Communication Overview

Fig. 4 shows the communication graph for candidate 2. It can easily be seen that there is no single center of communication as opposed to the system of C1. Many devices are paired directly with each other and only 6 of 19 available sensors communicate with *Z 2.1*. Furthermore, neither the remote control *F 2.1* nor any of the three switches paired with it communicate with *Z 2.1*. Thus, we can almost certainly rule out any automation rules for these devices, which gives us more insight into the habits of the inhabitants. Nevertheless, the segmentation of this installation and the VPN connection between the segments make it quite difficult to derive information about the physical presence of the inhabitants from the automated events alone. The keymatic remote controls *KF 2.1* and *KF 2.2* are paired with many actuators in addition to the keymatic door locks and both keymativ remote controls are paired with both door locks.

4.2.2 Manual Examination of Message Graphs

The temperature values given by the sensors *T/L 2.1 – 2.3* and corresponding actuators *ST 2.1 – 2.3* are strong indicators for an automated heating concept. Over the weekends, the temperatures drop gradually and then rise again sharply at the start of the week. The different temperature and humidity curves and the temperature differences of up to 10°C between the sensors lead us to the assumption that they are installed in 2-3 different rooms.

When examining the activity of the remote control *F 2.1* we found events only in the first part of the observation period. This means that the user only uses the remote control within the office itself and does not control any devices at home. The observed activity is thus a very reliable indicator of when the user is definitely present at the office and thus, not at home.

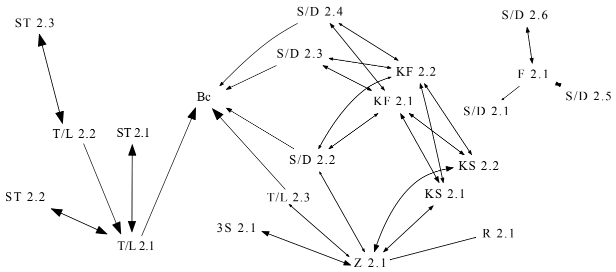

Figure 4: Directed communication graph for C2. The abbreviations are again those from Tab. 1

Keymatic Door Lock System

The most interesting data for Candidate 2 came from the automatic door lock system. Every day at about 9:15 as well as between 20:00 and 22:00 the door locks report their status to the central unit Z 2.1. We observed that there are always two messages shortly after one another, first *3S 2.1* sends the state *open* and after a maximum of 60 seconds the state *close*. Correlating the states of *3S 2.1* and *KS 2.2*, we concluded that *3S 2.1* is installed on the same door as *KS 2.2*. Due to this combination, the presence of the inhabitants can be easily predicted. Usually there is nobody at home between 9:30 and 20:30, except for mondays, where the time of absence lies between 13:00 and 21:00.

4.2.3 Correlation Analysis

Our correlation analysis found strong colleations between the thermostats and the heating actuators as well as between the remote control *F 2.1* and the actuators *S/D 2.1*, *S/D 2.5* and *S/D 2.6*. The thermostats show a consecutive acknowledgement of the heating actuators new position in 98.8% of all cases. The switch and dimmmer actuators even send their status as a reaction to a previous command from the remote control in 100% of all cases.

We see similar clear results for the reactions of the Keymatic door lock systems and the switches/dimmers *S/D 2.2*, *S/D 2.3* and *S/D 2.4* to the Keymatic remote controls *KF 2.1* and *KF 2.2*, where we recorded a reaction in 90% of all cases. In addition to what we already knew, the correlation analysis revealed that the tri-state sensor *3S 2.1* seems to have a relation to *KS 2.2*, since over 60% of all status changes of *KS 2.2* result in a status change of *3S 2.1*.

4.2.4 Filtering Automated Events

To filter automated events we used the same approach as for the first candidate. We generally found the results to support our findings from the manual analysis.

Using the automatic filtering method we could confirm our assumption that the unlocking command sent from the base station *Z 2.1* to *KS 2.2* at 8:30 in the morning does belong to an automation rule. The same holds for the command that locks *KS 2.2* again at 22:30.

Furthermore we found that the the heating actuators are automatically turned off at night. They regularly receive a *Pos.: 0%* command from the temperature/humidity sensors.

4.3 Confirmation of Results

After our experiments, we interviewed both candidates and discussed our findings with them. We were able to confirm our conclusions about the locations and purposes of the different devices. The candidates also confirmed our assumptions about automation rules and user habits.

4.4 Encrypted Communication

Applying encryption to all traffic in the HAS would make the aforementioned attacks harder to some degree. Both our approach to determine device types as well as our definition of an event for the correlation analysis and automation rule filtering incorporated the message payloads. They could not be applied in the same way if these payloads were encrypted.

However, encryption alone would not prevent an attacker from learning some information about the user. If the packet's source and destination address are unencrypted, an attacker can still try to identify devices by using heuristics. For example, devices that, from time to time, send two messages in short succession are usually door state sensors. Devices that only send one message each day can be assumed to be smoke detectors and devices that receive one message each morning and another one each evening can be assumed to be door locks. The devices that communicate the most are base stations, followed by temperature and humidity sensors.

Even if the complete packet, including the full header, is encrypted, some information leaks to an eavesdropping adversary. Activity in C1's HAS between 12:00 and 18:00 was 8.6% higher on weekends and holidays than it was on working days. HAS activity in C2's office was 21.3% lower during these times, strongly indicating presence and absence.

5. CONCLUSIONS

We have analyzed of two installations of the HomeMatic system and we have shown that this kind of system poses a significant threat to the privacy of the users. In general, systems that do not apply any kind of encryption leak a large amount of information to any observer keen enough to look for it. No prior knowledge about the installation or the victim is necessary to perform this kind of attack.

Furthermore we have gained knowledge about the traits of a HAS, such as possible communication links and how frequently a device usually sends messages. This information can be used to attack systems which apply encryption and thus at the very least identify when users are at home. As long as the systems do not provide a systematic protection against traffic analysis attacks, they should be considered vulnerable. To the best of our knowledge, no publicly available system provides this sort of protection as of now.

While we performed many tasks and checks manually during our experiments, most of them can be automated with the knowledge from our findings. Our parameters for finding automated events proved useful, as long as the automation rules did not change times themselves. Possible communication links, the usual frequencies with which the different devices send status messages and the number of messages being exchanged for each action of one device can be used to program heuristics which can then in turn identify devices in a system where packet payloads are encrypted.

5.1 Future Work

Considering how easy it is to attack current HASs, it is essential to protect against the attacks mentioned in this paper. While encryption schemes are available and can be readily applied, protection against traffic analysis attacks in HA networks is yet to be developed. Generating dummy traffic in an effective and efficient manner can help tackle this problem. Focus here can be put on the fact that the energy consumption of sensors and actuators has to stay as low as possible, but the base station is usually connected to a power line and can thus send dummy traffic without considerably decreasing battery lifetimes.

6. REFERENCES

[1] J. Al-Muhtadi, M. Anand, M.D. Mickunas, and R. Campbell, *Secure smart homes using Jini and UIUC SESAME*, ACSAC '00, ACM, pp. 77–85.

[2] P. Bergstrom, K. Driscoll, and J. Kimball, *Making home automation communications secure*, Computer **34** (2001), no. 10, 50–56.

[3] D. L. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, CACM **24** (1981), no. 2, 84–90.

[4] G. Danezis, R. Dingledine, and N. Mathewson, *Mixminion: design of a type III anonymous remailer protocol*, Symposium on Security and Privacy, 2003. Proceedings., IEEE, pp. 2–15.

[5] E. De Cristofaro, X. Ding, and G. Tsudik, *Privacy-Preserving Querying in Sensor Networks*, ICCCN, '09, IEEE, pp. 1–6.

[6] M. J. Freedman and R. Morris, *Tarzan: A Peer-to-peer Anonymizing Network Layer*, CCS '02, ACM, pp. 193–206.

[7] D. Goldschlag, M. Reed, and P. Syverson, *Onion Routing*, CACM **42** (1999), no. 2, 39–41.

[8] M. Hatler, D. Gurganious, C. Chi, and J. Kreegar, *Smart Home Sensor Networks*, Tech. report, ON World Inc., 2011.

[9] Y.-X. Li, L. Qin, and Q. Liang, *Research on Wireless Sensor Network Security*, CIS '10, IEEE, pp. 493–496.

[10] Z. Li and G. Gong, *A Survey on Security in Wireless Sensor Networks*, (2011).

[11] A. Marin, W. Mueller, R. Schaefer, F. Almenarez, D. Diaz, and M. Ziegler, *Middleware for Secure Home Access and Control*, PerCom Workshops '07., IEEE, pp. 489–494.

[12] NSA, *Tor Stinks*, Presentation, January 2007.

[13] A. Pfitzmann, B. Pfitzmann, and M. Waidner, *ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead*, KiVS '91, Springer, pp. 451–463.

[14] J. Sen, *A Survey on Wireless Sensor Network Security*, IJCNIS **1** (2009), no. 2, 55–78.

[15] H. Strese, U. Seidel, T. Knape, and A. Botthof, *Smart Home in Deutschland — Untersuchung im Rahmen der wissenschaftlichen Begleitung zum Programm Next Generation Media (NGM) des Bundesministeriums für Wirtschaft und Technologie*, Tech. report, Institut für Innovation und Technik (iit) in der VDI/VDE-IT, Berlin, May 2010.

[16] Y. Wang, G. Attebury, and B. Ramamurthy, *A survey of security issues in wireless sensor networks*, Communications Surveys & Tutorials, IEEE **8** (2006), no. 2, 2–23.