Deducing User Presence From Inter-Message Intervals in Home Automation Systems

Frederik Möllers and Christoph Sorge

CISPA, Saarland University, 66123 Saarbrücken, Germany
{frederik.moellers,christoph.sorge}@uni-saarland.de

Abstract. Privacy in Home Automation Systems is a topic of increasing importance, as the number of installed systems constantly grows. In this paper we investigate the ability of an outside observer to link sets of message timestamps together to predict user presence and absence. The question we try to answer is: *If attacker Eve has captured 1 hour of traffic from victim Alice's HAS and knows whether Alice was present at that time, can Eve deduce Alice's state by capturing another hour of traffic?* We apply different statistical tests and show that in certain situations, the attacker can infer the user's presence state with absolute confidence.

Keywords: Traffic Analysis · Home Automation · Statistical Analysis · Privacy

1 Introduction

Home Automation Systems (HASs) are rapidly gaining popularity. They can be used to control lights, window blinds, heating etc. Wireless HASs are currently most popular for use in private homes, as the installation is easier than for their wired counterparts. However, previous research has shown privacy risks of wireless HASs [17]: A passive eavesdropper can derive information about user habits from message contents and metadata. Message encryption—the obvious countermeasure—does not prevent analysis of communication patterns: packets sent by a remote control to a door lock reveal that a user is locking or unlocking a door. Even if addresses are obfuscated, message intervals could reveal information, e.g. about absence or presence of users interacting with the HAS. In this paper, we study the extent of information leakage through message intervals in HASs. Our contribution consists of two parts: We present a new attack vector which passive adversaries can use to *infer information about the presence of users* from the timings of messages alone. Additionally, we analyse the success rates of this approach and determine conditions under which a high confidence can be achieved. While we acknowledge that certain situations are not distinguishable by software using our model (e.g. a user being asleep and not interacting with the system vs. a user being absent), the goal is to find out whether or not situations exist in which a correct statement can be reliably achieved.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-33630-5_25

The paper is structured as follows: In Sec. 2 we give an overview of existing research in similar areas. Sec. 3 contains the definition of our system model as well as our model of the attacker. In Sec. 4 we summarise the attack method whose effectivity we are investigating. Sec. 5 contains the description of our analysis procedure and is followed by its results in Sec. 6. We conclude the paper and provide an outlook on future work in Sec. 7.

2 Related Work

Several authors have pointed out the privacy risks of HASs. Jacobsson et al. provide an overview of security and privacy risks in HASs [8]. A survey by Denning et al. states "activity pattern privacy" with the sub-goals of "presence privacy" (which we investigate in the paper at hand) and protection of occupant identities as HAS security goals [5]. Privacy implications of specific systems have been studied by Mundt et al., who derived information about user habits from the communication of office building automation systems [14], and were able to eavesdrop on communication of a wired bus system from a distance of 5 cm [13]. In our own previous work, we have demonstrated the extent of information leakage from a wireless HAS that neither encrypts communication nor attempts to obfuscate sender and receiver addresses [17]. Moreover, we have studied legal aspects of HASs that use data processing in the cloud [16]. Packet inter-arrival times as a side channel have been considered by Wendzel et al. [19], but their work focuses on establishing covert channels. Our contribution instead addresses the problem of deducing information from existing timings.

As wireless HASs are a specific type of wireless sensor networks (WSNs), some general results about WSNs might apply. There is a considerable body of literature on privacy in WSNs. In their survey [10], Li et al. distinguish between data privacy (concerning both the queries and the sensed data) and context privacy, with the latter term referring to both *location privacy* and *temporal privacy*. A number of publications consider traffic analysis in WSNs as a means to breach location privacy [4][11][20], but this aspect is not very relevant in HASs. While temporal privacy (which concerns the ability of an attacker to determine the timing of an event detected by the WSN) is related to the problem we investigate, we do not consider individual events in the paper at hand.

The use of traffic analysis (i.e. analysis of traffic patterns without consideration of communication contents) is not restricted to particular networks; the distribution of message inter-arrival times is commonly considered in traffic analysis. For example, Moore and Zuev [12] use that distribution (among other discriminators) to classify internet traffic; Bissias et al. [2] use inter-arrival times to identify web sites in encrypted, proxied HTTP traffic. Čeleda et al. [18] use traffic analysis in Building Automation Networks to detect attacks.

3 System and Attacker Model

For our analysis we assume the following situation. A user Alice has installed a home automation system. The system generates messages based on automation rules and in reaction to user behaviour. Both the rules and Alice's habits are known only to Alice. As the idea of this paper is to analyse if certain information can be deduced from message timings alone, it makes sense to exclude all other possible sources of information an attacker might be able to use. Real-world observations as well as publicly known statistics (e.g. "The average user is asleep during the night and at work from 09:00 to 17:00.") are explicitly neglected here.

The network topology of our model is a fully connected graph with respect to intended communication. This means that any two devices which are intended to communicate with each other can do so directly. This model is used in many available products; only few systems employ multi-hop communication. However, the research presented in this paper can be used as a base for developing dummy traffic schemes in both types of systems.

The communication is fully encrypted and packets are padded to a fixed length. Both message payloads and message headers (including source and destination addresses) are hidden from an outside observer.

In certain situations, low-level channel information can be used to try and fingerprint devices when both sender and receiver are static [1][3][6]. For the analysis at hand, we disregard this possible source of information. We argue that these kinds of attacks require a level of effort and dedication from the attacker which is unrealistic for common houses or when mounting traffic analysis attacks on a large scale against many buildings at once. Furthermore, countermeasures against these attacks have been explored in literature. We thus assume that the attacker cannot determine the source of a packet by these means.

We model our attacker—Eve—as a global passive adversary. Eve can detect any communication happening within the network, i.e. she can capture any packet being transmitted. However, Eve cannot break the packet encryption and she cannot distinguish between different devices by other means such as triangulation or wireless device fingerprinting.

Eve's goal is to determine whether or not Alice is at home at a given time. For this, we assume she has the following *a priori* information about Alice's home automation system:

1. Eve knows that Alice's HAS does not generate dummy traffic.
2. Eve has captured all communication packets during one hour of HAS operation. She also knows whether Alice was at home during this time.

The reason why we choose an interval of one hour for item 2 is twofold. On the one hand, a time frame of more than one hour allows Eve to mount sophisticated device fingerprinting attacks [6], invalidating our assumptions. However, it also makes decisions less useful: The longer the time frame, the less likely Alice is to keep this state during the next minutes or hours. On the other hand, a shorter time frame makes decisions harder, as there is less data to base an assumption

on. We performed the same experiments with time frames of half an hour and two hours, getting nearly the same results: The difference in the AUC values in Sec. 6.3 was 0.005 on average with a maximum of 0.104.

Using the available information, Eve needs to decide at a given time whether Alice is currently at home or not. Eve can capture the communication packets again for the same time frame to try and deduce Alice's presence state.

4 Attack Methodology

Our analysis works as follows: We assume the role of the attacker, Eve. Using the captured communication packets from two different time frames of one hour each, we try to find similarities in the statistical distribution of timestamps or inter-message intervals. We apply three different statistical tests to the two samples: The Kolmogorow-Smirnow Two-Sample Test [9], the Chi-Square Test of Independence [15] and the "Message Counts Test".

The statistical tests used here tackle the null hypothesis that *the two samples have the same underlying distribution function.* Instead of rejecting the null hypothesis with a certain confidence at a threshold depending on the desired confidence, we analyse the computed test statistics and try to determine suitable thresholds ourselves. The reason behind this is twofold: On the one hand, we do not have any a priori knowledge about the underlying distribution functions. On the other hand, we want to determine whether the difference in the distributions between two samples with different user states is high enough to allow a distinction based on the calculated test statistics. If this is the case, we can subsequently calculate thresholds and resulting confidence values for HASs.

4.1 Kolmogorow-Smirnow Test (KS Test)

The Kolmogorow-Smirnow Test for homogeneity [9] is based on the empirical cumulative distribution functions of the two input samples. Informally speaking, it measures the maximum vertical distance between the two curves. Formally, given to samples $X = [x_1, x_2, \ldots, x_n]$ and $Y = [y_1, y_2, \ldots, y_m]$ with respective empirical cumulative distribution functions F_X and F_Y, it computes the value

$$D = \sup_a |F_X(a) - F_Y(a)| \tag{1}$$

If the result D is high, the null hypothesis is rejected.

We use the `SciPy`[3] implementation of the KS 2-sample test from `SciPy` version 0.14.0 and apply it to the inter-message time intervals. In addition to the KS statistic D (sometimes referred to as $d_{\max}$ or $D_{a,b}$ in literature), the implementation computes a *p-value* as a function of D and the sample sizes. This accounts for the fact that large samples with the same underlying distribution are expected to show less differences than smaller samples (as per the law of large numbers). We examine both the value of D as well as the *p-value*.

[3] `http://www.scipy.org`, accessed 2015-12-18

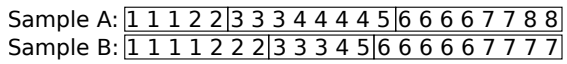

Fig. 1. Example of the approach used for binning using a minimum bin size of 5. The bounds are chosen so that at least 5 elements of each sample fall into one bin.

4.2 Chi-Square Test (χ^2 Test)

Pearson's Chi-Square Test [15] follows a similar approach as the KS test, but calculates the sum of squared differences between the actually measured frequencies and the expected ones. In the 2-sample form, the expected frequencies are estimated by taking the average frequencies of the two samples. Formally, the test expects categories and respective frequencies as inputs. Given two samples and m categories, these frequencies can be written as $X = [x_1, x_2, \dots x_m]$ and $Y = [y_1, y_2, \dots, y_m]$, where x_i is the number of elements in the first sample which fall into the i-th category. Using the intermediate definitions

$$n = n_x + n_y = \sum_{i=1}^{m} x_i + \sum_{i=1}^{m} y_i \tag{2}$$

$$\forall z \in \{x, y\} : E_{z,i} = \frac{n_z \times (x_i + y_i)}{n} \tag{3}$$

the test statistic is then defined as

$$\chi^2 = \sum_{i=1}^{n} \frac{(x_i - E_{x,i})^2}{(E_{x,i})} + \sum_{i=1}^{n} \frac{(y_i - E_{y,i})^2}{E_{y,i}} \tag{4}$$

If the value of χ^2 is high, the null hypothesis ("The two samples have the same underlying distribution function.") is rejected.

For the Chi-Square Test, we use a custom implementation. Similar to the Kolmogorow-Smirnow Test, it is applied to the inter-message time intervals.

As the test expects the two samples to be categorized into bins, we need to do this before calculating the actual test statistic. Literature suggests to choose bin sizes so that no bin contains less than 5 elements for any sample [7]. Thus, we adaptively choose bins of varying size. The lower bound for the first bin is the lowest value in any of the two input samples. The upper bound for a bin (which is also the lower bound for the next bin) is chosen as the smallest number which results in at least 5 elements of each sample falling into this bin. We thus guarantee that at least 5 values are in each bin for each sample. An example for the binning approach is depicted in Fig. 1. For the Chi-Square Test we calculate and examine the test statistic.

4.3 Message Counts Test (MC Test)

Our "Message Counts Test" divides the number of messages in the larger sample by the number of messages in the smaller one and subtracts 1, resulting in a

value within $[0, +\infty[$. Higher values indicate larger differences in the amounts of messages, just as higher results in the other tests indicate different distributions. The idea behind it is that if the sheer amount of activity in the system is very different to that during the reference time frame, the user state is likely to be different. For example, if the reference capture was taken while Alice is present and the capture in question shows lower activity, Alice is likely to be absent.

Formally, given two samples $X = [x_1, x_2, \ldots, x_n]$ and $Y = [y_1, y_2, \ldots, y_m]$, the test statistic is defined as

Similar to the Chi-Square test, we calculate and examine the test statistic.

$$C = \frac{\max(n, m)}{\min(n, m)} - 1 \tag{5}$$

5 Analysis Procedure

We obtained input data for our analysis by collecting packet captures from two real-world home automation systems. *System 1* is an installation using *Home-Matic* hardware, an off-the-shelf solution for consumers, which was already used as a base for our previous work on this topic [17]. The owner voluntarily captured all traffic for 36 days and provided us with the log files as well as presence and absence times. *System 2* is data from a custom system, built by combining multiple automation products from different manufacturers. Traffic was recorded for 37 days and published in a series of news articles[4].

As a first step, we annotate each message with the user state: PRESENT and ABSENT are chosen based on the available data. A third state, ASLEEP is introduced to handle the fact that during night hours (22:00 to 08:00), users are usually asleep and thus the activity of the system is reduced. Due to the vague nature of the ASLEEP state and the fact that we cannot be sure whether the users were actually asleep during this time, we exclude it from further analysis and only investigate messages whose state is either PRESENT or ABSENT.

Analysing each system by itself, we construct (non-overlapping) intervals of 1 hour each during which the user state did not change. For each interval, we gather the messages sent during this time into *Message Groups*. Each Message Group is thus identifiable by its system and the timestamp of the first message. Also, as per the construction of intervals described above, each group has a fixed user state. For System 1, we obtain 180 Message Groups with state PRESENT, 136 Message Groups with state ABSENT and 237 Message Groups with state ASLEEP. For System 2, the numbers are 223, 125 and 296, respectively.

For all non-identical combinations of Message Groups (only considering those with states PRESENT and ABSENT)—167,941 in total—we perform the 3 statistical tests mentioned in Sec. 4. We then visualize the results in boxplots, both overall per system as well as individually for each combination of user states. In a second step, we test different thresholds for all tests and plot the true and false positive rates in ROC diagrams.

[4] `http://spon.de/aeDkn`, accessed 2015-12-18.

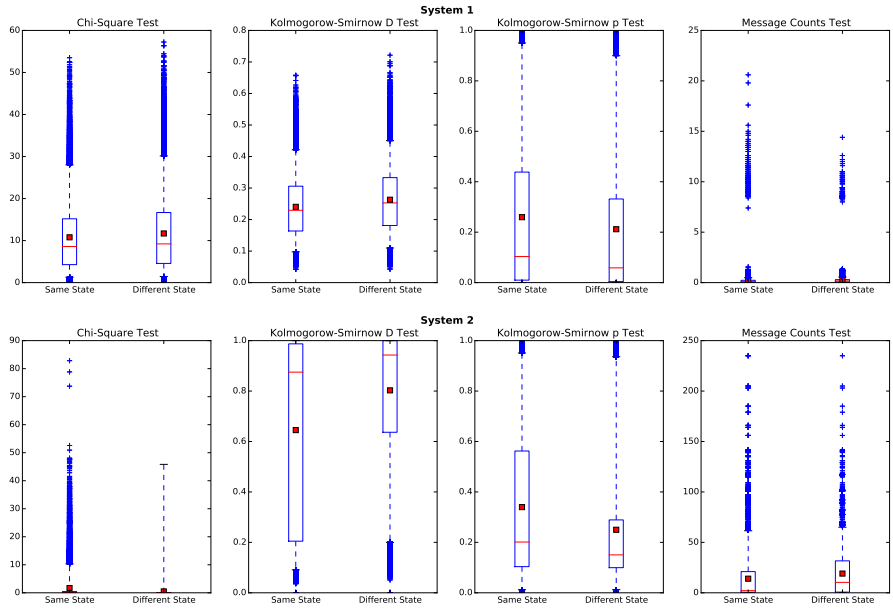

Fig. 2. General Test Results for both systems. The boxes extend from the first to the third quartile. The whiskers extend up to $1.5 \times IQR$ past the boxes, where IQR is the interquartile range. If $IQR = 0$ (as with the χ^2 Test for different states in System 2), the whiskers extend up to the minimum and maximum values. Red lines mark the medians while red squares mark the arithmetic means. Blue plus signs show outliers beyond the whiskers.

6 Analysis Results

6.1 Test Suitability in the General Case

At first, we plot all test results by system and test and only distinguish between the two cases whether or not the samples have different user states. This section gives a general and quick overview over the suitability of the tests for our purposes. If the plots of the two cases differ significantly, the test results carry a high amount of information and if they are largely the same, the information immediately available from the test result is limited. The plots are visualised in Fig. 2 for both systems. The boxplots do not show any immediately obvious peculiarities. For both systems and all tests, the boxes overlap and thus suggest that the tests cannot be used as a universal oracle telling Eve whether the 2 compared samples have been taken with the same user state.

System 1. For System 1, the χ^2 *Test* values are broadly spread. Comparing samples with the same user state yields values from 0 to 53.5, samples from different states lead to values from 0 to 57.2. This suggests that there may be an

upper bound to the value for samples of the same state and that values above this limit indicate a different state of the two compared samples.

The *KS Test* statistic D ranges from 0.04 to 0.66 for the same state and from 0.04 to 0.72 for different states. Like the Chi-Square test, this suggests an upper bound for the value in the same-state case.

The *KS p-values* again provide similar information. For the same state, the values range from 6.41×10^{-12} to $1 - 10^{-11}$, for different states they range from 4.76×10^{-15} to $1 - 10^{-12}$. The null hypothesis ("The two samples originate from the same distribution [=the same state].") is rejected for p-values lower than a threshold. The lower minimum value for different results shows that the default thresholds are not useful in our scenario.

The *MC Test* provides the least useful results. The values are in fact misleading: While they range from 0 to 20.6 for samples with the same state and the minimum is the same for different states, the maximum value in the latter case is only 14.4. This shows that while the user state does not change, the number of messages being generated in a given time frame can differ significantly.

System 2. The results for System 2 offer much less information than those for System 1. The χ^2 *Test* values range from 0 to 82.8 for samples with the same state and from 0 to 45.8 for samples with different states. As shown in Fig. 2, 75% (the lower three quartiles) of the tests with different states had the result 0. These values are misleading if interpreted in the same way as those of System 1. Intuitively, the values should be higher for different states (and they are for System 1). We conclude that either the test's usefulness depends on the type of the HAS or that the previous results were not representative.

The *KS Test statistic D* yields values in the full range $[0, 1]$ for samples with the same state. While this already indicates that the test is not useful for this system, the same range of values for samples with different states support this.

Consequently, the *KS Test p-values* are inconclusive as well: They range from 1.13×10^{-16} to 1 for the same state and from 2.28×10^{-8} to 1 for different states.

The *MC Test* surprisingly yields the exact same range of values for both cases: The results range from 0 to 235 in both cases.

6.2 Test Suitability Per State Pair

In the next step we take a closer look at the different combinations of user states. Our hypothesis is that the tests may give useful results for certain combinations of states and less useful results for others. This section deals with the performance of the tests for a given pair of user states. Fig. 3 summarizes the results for both systems.

System 1. Results of the χ^2 *Test* for System 1 do not yield much more information than what we could already see from the general evaluation. The two cases in which the user states are the same largely overlap and the ranges of values are almost the same. The same holds for both statistics of the *KS Test*.

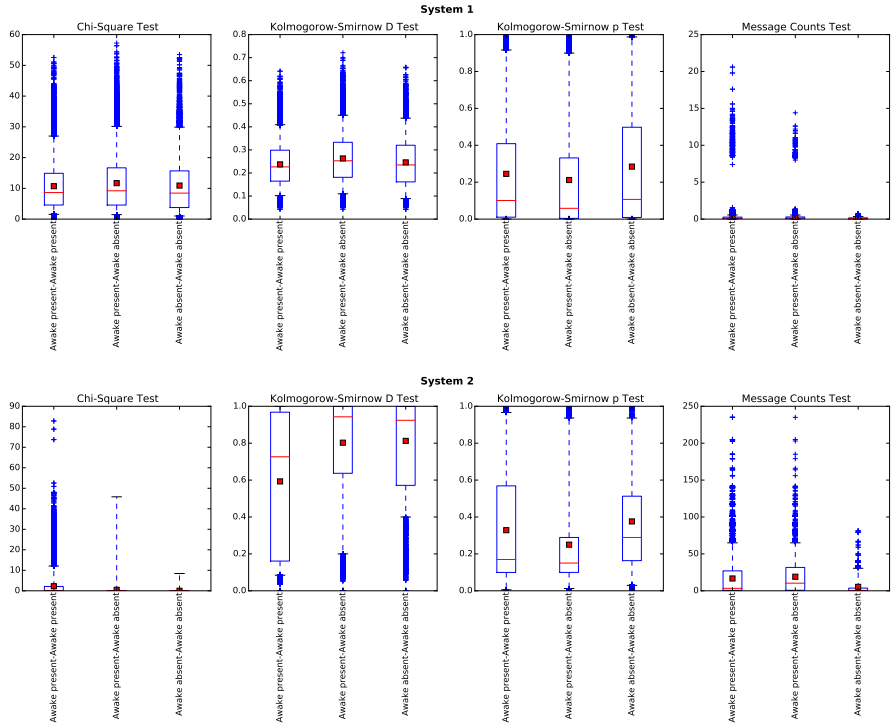

Fig. 3. Per state pair test results for both systems. The plot parameters are the same as for Fig. 2. The reason why the combination PRESENT-ABSENT does not appear is the symmetry of all tests: $T(a, b) = T(b, a)$.

The *MC Test* provides some new results: If both samples have the state ABSENT, the values do not go above 0.71. This means that if Eve obtains a sample known to have the state ABSENT, and gets higher value when comparing it to a second (unknown) sample, she can be sure that Alice was present during the time frame of the second sample. However, this is only the case for 2.4% of the tested ABSENT-PRESENT sample pairs.

System 2. In contrast to System 1, the boxplot of the χ^2 *Test* for System 2 exhibits obvious differences between the state pairs. If one of the samples has the state ABSENT, 75% of the tests evaluate to 0. Similarly to the MC Test for System 1, the plots show that there is a threshold above which Eve can be sure that Alice is PRESENT if her first (known) sample has the state ABSENT. This threshold is at 8.45 and 2.12% of the ABSENT-PRESENT pairs reach a higher value. However, also similar to System 1, Eve cannot make such a confident decision if her known sample has state PRESENT.

The *KS Test* does not show such features; this is consistent with System 1.

The *MC Test* confirms the observation from the χ^2 Test and yields another threshold. The threshold value is 81 and 1.77% (494 out of 27,875) of the tests with different states result in higher values. Surprisingly, though, none of these 494 Message Group pairs gave a result above the threshold for the χ^2 Test. In fact, some pairs even evaluated to 0 in the χ^2 Test. This is highly interesting, as it suggests that a combination of different tests with the same input data can provide significantly more information than one test alone. Using the thresholds of both tests, Eve can identify 3.89% or 1084 of 27,875 Message Group pairs as having different states if one of the samples is known to have the state ABSENT.

6.3 The Effect of Different Thresholds on Classification Rates

As shown in the previous section, some tests exhibit maximum values for certain state combinations, and knowing such values may enable Eve to infer Alice's user state at a given time with absolute confidence. Below these, however, statements about presence and absence are more difficult to make. In this section we examine the effect of different chosen threshold values on the classification rates.

We compute True and False Positive Rates TPR and FPR for all possible threshold levels using the data from the tests previously conducted. In our case, the rates are defined as follows:

If $s(a)$ is the state of a sample a, $T(a,b)$ is the test result of the pair (a,b), t is the threshold value below which sample pairs are classified as having the same state and $N_{a,b}(cond)$ is the number of sample pairs a,b which satisfy a condition $cond$, then

$$TPR = \frac{N_{a,b}(s(a) = s(b) \wedge T(a,b) < t)}{N_{a,b}(s(a) = s(b))} \tag{6}$$

$$FPR = \frac{N_{a,b}(s(a) \neq s(b) \wedge T(a,b) < t)}{N_{a,b}(s(a) \neq s(b))} \tag{7}$$

TPR is the number of correctly classified same-state pairs divided by the total number of same-state pairs and FPR is the number of different-state-pairs which were incorrectly classified as having the same state divided by the total number of different-state pairs. TPR is a measure for how well the test can identify samples with the same state as the source and FPR is a measure for how often the test falsely reports two samples for having the same state.

In order to visualise the rates, we plot ROC (Receiver Operating Characteristics) curves and calculate the AUC (Area Under Curve) for all of them. ROC curves illustrate how fast the test performance drops (i.e. how fast the False Positive Rate increases) when raising the threshold to get a higher True Positive Rate. The AUC is a numerical measure for this quality: In the ideal case (the test has a TPR of 1.0 and a FPR of 0.0) the value is 1 and in the worst case (the test does not perform better than randomly guessing), the value is 0.5. Values below 0.5 are similar to values above, since the test result interpretation can be inverted to invert the ROC curve (i.e. values *above* the threshold are interpreted as indicators for a same-state pair).

A selection of ROC curves is depicted in Fig. 4. Some tests (most notably the χ^2 Test for System 2) yield high values for both rates with the lowest possible threshold, which is why the curves do not start at the origin $[0, 0]$. To calculate the AUC for these cases, we use the *line of no-discrimination*—the values obtained by randomly guessing—up to the FPR of the lowest threshold (the X coordinate). From there on, we proceed with the regular estimation and calculate the area below the straight line between two subsequent data points.

Most curves do not exhibit large deviations from the mean line. For System 1, both the χ^2 Test and the two KS Tests yield an AUC between 0.52 and 0.57. Only the MC Test performs slightly better, the AUC is 0.525 for a source sample with state PRESENT and 0.688 for an ABSENT source sample (shown in Fig. 4).

Overall, the results for System 1 suggest that statistical tests are only of limited use in deducing user states from inter-message intervals.

System 2 mostly confirms this observation, although the performance of the different tests varies drastically.

The χ^2 Test performs badly: For a PRESENT source sample, the minimum obtainable False Positive rate is 91.6% at a True Positive Rate of 61.3% (the threshold value in this case is 0). For an ABSENT source sample, the minimum False Positive rate is consequently the same, but the minimum True Positive rate is 98.0%. The KS Test and the MC Test perform much better. Their AUC values are relatively high and significant True Positive rates can be obtained while keeping the False Positive rates below 50%.

From the analysis of the ROC curves we draw two conclusions. Firstly some tests exhibit a significant deviation from the line of no-discrimination. Combining multiple tests could further improve the results and yield more information. Secondly we can confirm our previous observation that extreme threshold values lead to absolute certainty in the classification.

6.4 Feasibility of Detection in Practice

The statistical tests do not yield clear results in all cases we examined. However, upper or lower bounds can be determined in some cases, which then allow Eve to make statements with absolute confidence. The requirements for these thresholds to be useful for Eve are not hard to meet: She needs a source sample which—when tested in conjunction with samples of a different state—yields values above or below the thresholds.

To verify the practicability of this attack we divide our traffic data into a training set and a test set. For training, we use the first 70% of our data (221 Message Groups from System 1, 244 Message Groups from System 2).

We perform all aforementioned tests on the training data and calculate thresholds for Message Group pairs with the same state. Using these threshold, we choose one Message Group for every system and state where the amount of correct classifications among the training data is maximized—i.e. the Group with the highest TPR among the training data. We then check each of these Groups against the test data and calculate True and False Positive Rates using the thresholds calculated from the training data before.

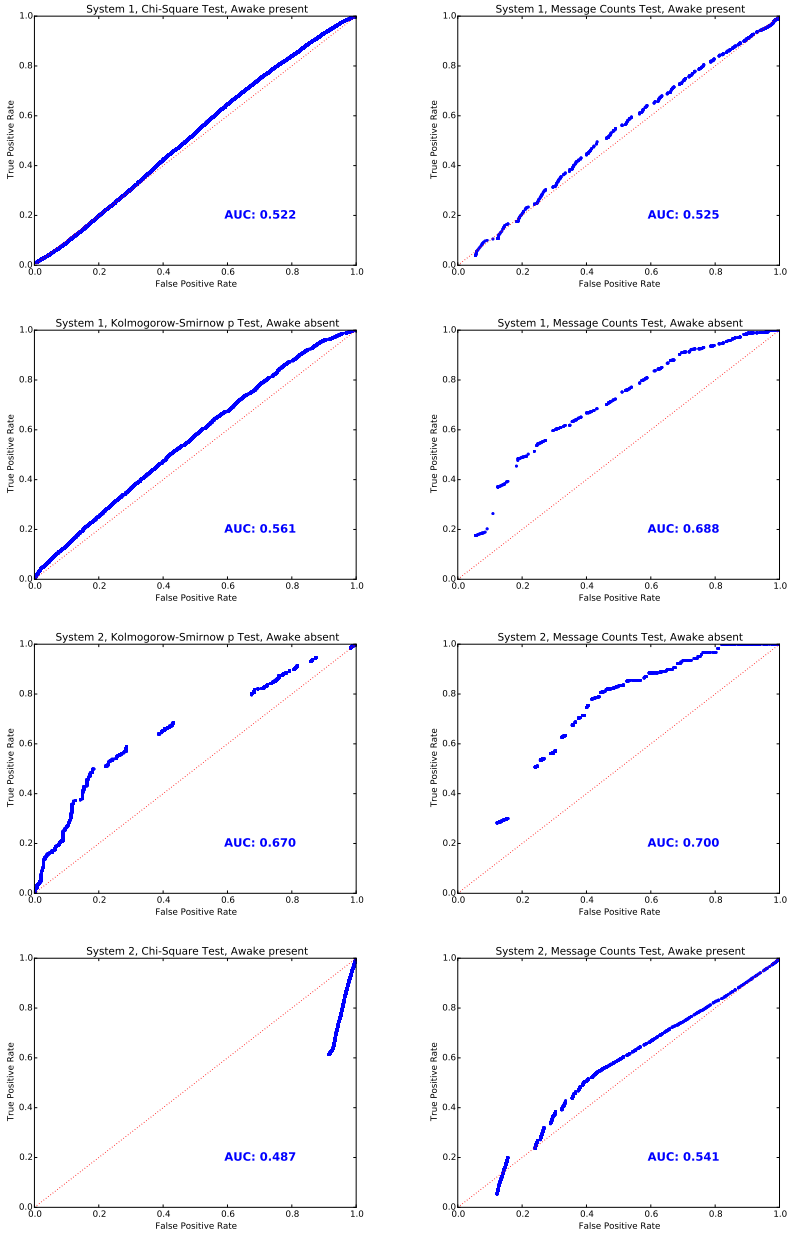

Fig. 4. ROC curves for different tests and source states. Blue points show the actual values, dotted red *lines of no-discrimination* show linear ascension from $[0, 0]$ to $[1, 1]$—the values obtained by randomly guessing. The graphics indicate that the test performance strongly depends on the system and the source sample. As noted in Sec. 6.1, the χ^2 Test for System 2 produces counterintuitive results.

For System 1 using an ABSENT source sample, we reach a TPR of 5.3% and a FPR of 1.1%. This suggests that the attack is not useful in practice. Using a PRESENT source sample, however, the FPR is at 0 while the TPR reaches 1%. It is thus only a matter of time until Eve can successfully identify an ABSENT sample if she has a suitable PRESENT source sample. For System 2, the best ABSENT source sample achieves a TPR of 5.8% while the FPR also stays at 0. However, in the data for this System no suitable PRESENT source sample exists. The tests do not yield thresholds which allow for an unanimous classification.

This particular attack is not likely to be encountered in reality: Eve would have to manually observe Alice's home for several hours or even days, annotating the captured traffic with the user states for every one-hour sample. However, the experiment shows that under the right circumstances, unanimous classification is possible. The experiment supports the theory that system-wide thresholds exist which allow for a classification of states with absolute certainty. The follow-up question whether such thresholds exist for a manufacturer or production series remains to be answered.

7 Conclusion and Outlook

In this paper we have performed the first analysis of inter-message intervals in Home Automation using statistical goodness of fit tests. We have used sample data from two real world installations to measure the ability of an attacker in deducing user states. In particular, we tried to answer the question:

If Eve has captured 1 hour of traffic from the Alice's HAS and knows whether Alice was present at that time, can Eve deduce Alice's state by capturing another hour of traffic?

Comparing and combining various tests, we were able to identify conditions under which the question above could be confidently answered with *yes*.

The χ^2 Test provides little information with regard to the question. However, the MC Test and, in some cases, the KS Test reveal identifiable discrepancies between samples with different states. A combination of all three tests allow an attacker to mount a practical attack on the system and infer the user state by passively listening after obtaining a suitable source sample.

For future work, we will work on new tests and combine them with those applied in this paper in order to obtain more information. At the same time, we will study the different properties of HASs to find out if there are filtering techniques which can be applied to the samples in order to make the tests more effective. Since this increases the abilities of an attacker to predict user presence and absence without physical labour, we will also develop dummy traffic schemes for use in HASs. These offer users the ability to mask their traffic and hide their state from unauthorized observers.

References

1. Bagci, I.E., Roedig, U., Schulz, M., Hollick, M.: Gathering Tamper Evidence in Wi-Fi Networks Based on Channel State Information. In: Proc. ACM WiSec '14. pp. 183–188. ACM, New York, NY, USA (2014)
2. Bissias, G.D., Liberatore, M., Jensen, D., Levine, B.N.: Privacy Vulnerabilities in Encrypted HTTP Streams. In: Danezis, G., Martin, D. (eds.) PET 2005. LNCS, vol. 3856, pp. 1–11. Springer, Berlin, Heidelberg (2005)
3. Brik, V., Banerjee, S., Gruteser, M., Oh, S.: Wireless Device Identification with Radiometric Signatures. In: Proc. ACM MobiCom '08. pp. 116–127. ACM, New York, NY, USA (2008)
4. Deng, J., Han, R., Mishra, S.: Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks. In: Proc. IEEE/CreateNet SecureComm '05. pp. 113–126 (2005)
5. Denning, T., Kohno, T., Levy, H.M.: Computer Security and the Modern Home. CACM 56(1), 94–103 (2013)
6. Desmond, L.C.C., Yuan, C.C., Pheng, T.C., Lee, R.S.: Identifying Unique Devices through Wireless Fingerprinting. In: Proc. ACM WiSec '08. pp. 46–55. ACM, New York, NY, USA (2008)
7. Fisher, R.A., Yates, F.: Statistical tables for biological, agricultural and medical research. Oliver and Boyd, Edinburgh, 6 edn. (1963)
8. Jacobsson, A., Boldt, M., Carlsson, B.: A risk analysis of a smart home automation system. Future Generation Computer Systems 56, 719–733 (2016)
9. Kolmogorow, A.N.: Sulla Determinazione Empirica di una Legge di Distributione. Giornale dell'Istituto Italiano degli Attuari 4, 1–11 (1933)
10. Li, N., Zhang, N., Das, S.K., Thuraisingham, B.: Privacy preservation in wireless sensor networks: A state-of-the-art survey. Ad Hoc Networks 7(8), 1501–1514 (2009)
11. Li, Y., Ren, J.: Source-Location Privacy through Dynamic Routing in Wireless Sensor Networks. In: Proc. IEEE INFOCOM. pp. 1–9 (2010)
12. Moore, A.W., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In: Proc. ACM SIGMETRICS 2005. pp. 50–60. ACM, New York, NY, USA (2005)
13. Mundt, T., Dähn, A., Glock, H.W.: Forensic analysis of home automation systems. In: HotPETs '14 (2014)
14. Mundt, T., Kruger, F., Wollenberg, T.: Who Refuses to Wash Hands? Privacy Issues in Modern House Installation Networks. In: IEEE BWCCA '12. pp. 271–277 (2012)
15. Pearson, K.: On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can be Reasonably Supposed to have Arisen from Random Sampling. In: Kotz, S., Johnson, N. (eds.) Breakthroughs in Statistics, pp. 11–28. Springer Series in Statistics, Springer New York (1992)
16. Sorge, F.M., Christoph: Hausautomationssysteme im Datenschutzrecht. In: Cooperation: Proceedings of the 18th Legal Informatics Symposium IRIS 2015. pp. 553–558. Österreichische Computer Gesellschaft (2015), in German.
17. Sorge, F.M., Seitz, S., Hellmann, A., Christoph: Extrapolation and Prediction of User Behaviour from Wireless Home Automation Communication. In: Proc. ACM WiSec '14. pp. 195–200. ACM, New York, NY, USA (2014)

18. Čeleda, P., Krejčí, R., Krmíček, V.: Flow-Based Security Issue Detection in Building Automation and Control Networks. In: Szabó, R., Vidács, A. (eds.) EUNICE 2012. LNCS, vol. 7479, pp. 64–75. Springer, Berlin, Heidelberg (Aug 2012)
19. Wendzel, S., Kahler, B., Rist, T.: Covert Channels and Their Prevention in Building Automation Protocols: A Prototype Exemplified Using BACnet. In: Proc. IEEE GreenCom 2012. pp. 731–736 (Nov 2012)
20. Yao, L., Kang, L., Shang, P., Wu, G.: Protecting the sink location privacy in wireless sensor networks. Personal and Ubiquitous Computing 17(5), 883–893 (2013)