sciendo

Proceedings on Privacy Enhancing Technologies ; 2020 (4):376–393

Frederik Möllers*

Energy-Efficient Dummy Traffic Generation for Home Automation Systems

Abstract: Home and Building Automation Systems are becoming more and more popular these days. While they increase the comfort of living, they may also leak private information such as user presence to passive observers. In this paper we investigate approaches for the generation of dummy traffic in Home Automation Systems (HASs). We discuss fundamental requirements and their impact as well as two concrete dummy traffic generation algorithms. We measure the impact of Constant-Rate Dummy Traffic (CRDT) on the responsiveness and energy efficiency of Home Automation Systems. As an alternative, we present the Naive Exponential Dummies (NED) generation scheme in which the balance between privacy guarantees and energy efficiency can be arbitrarily moved. We formally prove its privacy guarantees and evaluate it against realistic sample data.

Keywords: keywords, keywords

DOI 10.2478/popets-2020-0078
Received 2020-02-29; revised 2020-06-15; accepted 2020-06-16.

1 Introduction

Home and Building Automation is becoming common among new houses and existing properties. Benefits of this technology range from increased comfort over assistive technology to higher security. Tedious tasks are no longer performed by the inhabitant and systems can also monitor the property for emergencies such as fires or burglaries. Naturally, these systems are tied to the users' lives and process information about the inhabitants' private space.

Contrary to the need for privacy, however, research has shown that smart home appliances often *undermine* the privacy of their users by leaking data—voluntarily or involuntarily—to outside observers. Early HASs transmitted data in the clear and allowed for detailed analyses about user behaviour and habits, down to the exact time a person woke up and left the house. [16] Further research has shown that wired connections are as susceptible to these attacks as wireless ones [19] and that information can still be deduced even if all content and addressing metadata is stripped or concealed. [7, 17]

Arguably, eavesdropping attacks on HASs are not a widespread issue as of now. However, this may partly be due to the fact that passive attacks are virtually undetectable. Even if a burglary is observed, it is next to impossible to reliably determine whether the perpetrator has used an eavesdropping attack on the HAS prior to their break-in. Moreover, the attacks may quickly rise as smart home devices are becoming more and more prevalent. Once a majority of households is equipped with such systems, large-scale surveillance becomes feasible for professional adversaries for the following reasons:

1. Eavesdropping attacks can be fully automated. Whole blocks of buildings can be monitored by a single device and without requiring a line of sight. The adversary can receive notifications such as "all inhabitants have left the building" and can use these to plan burglaries.
2. The hardware required to mount these attacks is becoming cheaper. The hardware used in previous papers [16] costs no more than $100 as of now and similar suitable devices are likely to further decrease in price. A strong antenna can increase the attack range well beyond the bidirectional communication range of smart home equipment, increasing the range of each listening device.
3. Traffic analysis attacks do not require any a priori knowledge about the victims such as encryption keys for the network. Available hardware is able to detect and log traffic from a variety of systems using different radio frequencies and protocols.

In this paper we investigate countermeasures to the information leakage from Home Automation and similar IoT systems from the timings of messages. Our contributions are as follows:

1. We identify key techniques such as the encryption of link-layer address information and the equalization of packet sizes which are necessary for any overlying dummy traffic generation scheme.

***Corresponding Author: Frederik Möllers:** Saarland University, E-mail: frederik@die-sinlosen.de

2. We provide a quantitative evaluation of dummy traffic generation in HASs with respect to the traffic volume as well as energy consumption overhead. While many related works evaluate the traffic overhead, little is known about the impact on energy efficiency. This is especially important in HASs where most devices are battery powered. We show that traffic overhead is not proportional to an increase in power consumption and highlight how this can be leveraged to achieve privacy at moderate cost in terms of energy consumption.
3. Given the heavy impact of Constant-Rate Dummy Traffic (CRDT) on power consumption, we propose a new dummy traffic generation scheme. By relaxing the targeted privacy goals, we propose a stochastic dummy traffic generation mechanism which allows for easy tuning of privacy versus traffic overhead. We call this scheme Naive Exponential Dummies (NED).
4. We formalize inherent shortcomings of zero-latency dummy traffic generation schemes as well as NED in particular and discuss the important breakpoints in energy efficiency and privacy.
5. We compare the performance of Constant-Rate Dummy Traffic and our stochastic NED scheme using data from real-world HAS installations which include authentic user interactions. We evaluate both approaches with respect to traffic overhead and energy efficiency.

2 Related Work

In recent years, research on privacy in smart home systems has gained momentum.

2.1 Smart Home Privacy

In previous works we have investigated deduction of user behavior and user presence from unencrypted [16] as well as encrypted [17] HomeMatic traffic. Copos et al. have performed a similar analysis on encrypted IEEE 802.11 (WiFi) traffic. [7]

Based on this, we have established a model for traffic analysis attacks in HASs. [18] Our concept takes ideas from differential privacy [28] and applies them to a communication network. This analysis builds on the previous findings with some amendments and offers two concrete approaches towards solving the problem of private communication in small-scale IoT networks such as HASs. Our evaluations are based on the same data set.

Apthorpe et al. have investigated HAS traffic analysis using a similar setup. [1] They have monitored encrypted WiFi traffic of popular HA devices and identified interactions based on spikes in the traffic rate. Similar to us, Apthorpe et al. model user interaction as a stochastic process. Their work however differs from this paper in several ways, including the following:

1. Their analysis is based on traffic from single devices whereas our analysis uses the output of a complete HAS system including interactions of different devices with each other.
2. Their evaluation focuses on traffic overhead. As we show in this paper, traffic overhead and energy overhead are related, but not proportional. Our evaluation takes this into account and provides an estimation of the energy consumption overhead of dummy traffic in HASs.
3. The limitation regarding "long user activities" does not apply to our approach, because the differential-privacy-based model uses a more general notion of traffic patterns induced by user interaction. In fact, our findings apply to any system regardless of how user interaction is distributed.

Despite the differences, some ideas from Apthorpe et al.'s STP algorithm can be applied to our approach. Partitioning time into short intervals which are then viewed as constant-rate traffic sequences (either empty or padded to maximum rate) makes it harder for an adversary to identify traffic patterns belonging to individual devices. Furthermore, it reduces the temporal dimension from a continuous to a discrete one which simplifies the calculation of privacy guarantees.

2.2 Wireless Networks

Research on Wireless Sensor Networks (WSNs) focuses on location privacy.[5, 6, 14] The goal is to hide the origin rather than the existence of communication.

Approaches for private communication in WSNs assume that messages are routed over several hops before arriving at the destination. In HASs, nodes are often located closely together wrt. their maximum transmission range. Therefore, mesh networks with direct source-to-sink communication channels as well as star topology networks with at most one hop are very common.

Yang et. al. propose to use Constant-Rate Dummy Traffic on all links. [31] We analyse a similar approach in

Section 5. An approach using less traffic has been proposed by the same authors, [26] but introduces considerable delays. As we state in Section 5, delays in HASs are to be avoided. Our scheme presented in Section 6 offers comparable privacy guarantees without delaying genuine messages.

In Low-Power Wide Area Networks (LPWANs), Leu et al. have formalized information leakage and cover traffic. [13] Their work significantly overlaps with our previous findings [18], but targets a different scenario. The system and attacker model are quite similar: Essentially, both models try to capture the confidence of an attacker guessing the genuine events in a captured traffic sample. We choose to base our contribution on our model due to its verifiable privacy goals that match intuitive, desirable properties of a HAS.

2.3 MIX networks

MIX networks and derived systems such as Tor [9] or Loopix [24] provide anonymity for their users. Similar to their attacker models, we have to assume a global adversary in HASs. Using readily available hardware it is easy to capture any and all traffic from a single HAS.

Contrary to MIX networks, routing in HASs is usually not performed and messages are being broadcast (most systems are either wireless or use a bus network). Furthermore, the attacker's goal is fundamentally different in HASs: The adversary tries to identify user interaction or the absence thereof, i.e. the existence and/or pattern of genuine communication. In models for MIX networks, the adversary's goal is usually to link the sender and receiver or to estimate the average sending or receiving behavior of users. The metrics established by us [18] are more suitable for the HAS scenario.

Previous research has shown limits of dummy traffic generation in MIX networks. [8, 21] However, HASs exhibit incompatible differences: Das et al. have explicitly excluded protocols where information is contained in the absence of messages. [8] Oya et al. assume an attacker who tries to estimate the generic sender or receiver profile of users in contrast to examining specific, fixed timing patterns. [21] Furthermore, the model used in this paper is agnostic to sender behavior, notably to changes in the genuine message rate.

Despite the differences, some approaches from MIX networks like Constant-Rate Dummy Traffic can be adapted and applied. [23] Shmatikov and Wang have developed an approach which uses adaptive padding to offer privacy at a lower communication overhead. [27] Our proposed NED scheme is similar, but does not build on pre-sampled traffic patterns and is tailored towards the characteristics HASs.

Loopix [24] also offers a property called *Sender online unobservability* which corresponds to our goal of hiding the existence of user interaction. It does so by having the users send data through the MIX network back to themselves. In HASs, we cannot leverage this route of partially trusted MIXes, as the system does not route messages at all. However, our approach builds on some of the same principles: Inter-message timings are modelled as an exponentially distributed random variable, thus the output of a given node corresponds to a Poisson process. This allows for an intuitive and well manageable model.

2.4 Website Fingerprinting

A large body of literature is available on the topic of website traffic fingerprinting and recognition. [4, 12, 22] The general idea is that web browsers exhibit a unique traffic pattern when accessing a single website. These patterns can be learned and later recognized to match users to websites even if the traffic is encrypted and possibly routed through a MIX network.

Some ideas from this field can be adapted and used in HAS settings as well. For example, traffic fingerprinting algorithms could be used to recognize known communication patterns between particular pairs of devices. However, in our particular use case the attacker has little to no a priori information about the system, the devices or the inhabitants. The attacks presented by us in previous works [17] leak information about the users without requiring large amounts of sample data and our model [18] is abstract. In general, the models of website fingerprinting and HAS traffic analysis differ in several regards:

1. In website fingerprinting attacks, the initiator of the communication is known and the attacker tries to match the counterpart against a set of known and publicly reachable entities. In our scenario, only the HAS to which the communicating parties belong is known. The attacker tries to determine whether some particular category of communication (e.g. genuine user interaction) is happening.
2. Countermeasures against website fingerprinting attacks generally aim to be applied at the user's node and possibly at nodes along the way to the web server. Unrelated third parties and the website itself are to be protected from negative side-effects of

the countermeasure. In HASs, all nodes are under the user's control. Except for regulatory thresholds, no third parties are involved and have to be considered.

3. In computer networks, a large traffic overhead degrades the performance of the system. In HASs, this directly affects battery lifetime and can lead to system unresponsiveness if regulatory thresholds for communication bandwidth are exceeded.
4. Routing or at least direction information is available in computer networks [22]. In broadcast HASs where the destination of a packet is sent unencrypted, only this information is available to an observer. If the destination is encrypted as well, no routing or direction information is available at all.

3 System and Attacker Model

Our model is largely similar to that used in our previous works [18], but we go into detail on certain decisions and where they impact our further analysis.

3.1 System Model

We assume that the HAS consists of an arbitrary number of nodes which communicate directly, i.e. without routing. Even if the network exhibits a star topology and messages are possibly routed over a base station, this can fit into our model. Disconnecting the transfer to and from the base station yields two transmissions which can be viewed as separate packets.

Temporal links between messages can appear even in networks without routing. For example, a user might have programmed their (mesh-network) HAS to switch on the light and turn on the heating whenever a door is opened. Thus, transmissions in close succession can happen in both mesh and star topology networks. Their commonness depends on the automation rules and user habits in addition to the network topology.

We further assume that all communication links are encrypted individually, which means that message payloads do not leak information to an observer. We stress that messages must be re-encrypted in case routing does happen in order to prevent the adversary from being able to match the incoming and outgoing message. End-to-end encryption can be added on top with no loss of generality. There are numerous approaches to provide confidentiality of message contents in systems with low computational power and limited power supply. [2, p. 1–2] Challenges such as key distribution and renewal are outside the scope of this paper.

Additionally, we assume that all messages are padded to the same length. In the Section 3.2 we dive deeper into the assumptions on encryption and padding and examine the effects on our evaluation should they not hold.

Last, we assume that message headers such as the intended receiver are either encrypted together with the message payload or are otherwise hidden from outside observers. This can be achieved by a mechanism such as SlyFy [11]. While this is a strong assumption and certainly not true for all available HAS products, we keep it for the sake of simplicity. As detailed in the following section, relaxing the assumptions does not invalidate the model. It merely splits a real-world system into several *virtual* systems that can be modeled and analyzed largely individually.

To summarize, our system is modeled as an entity that generates messages which are only distinguishable by their timestamps. An output O can thus be written as a series of timestamps $O = \{t_1, t_2, t_3\}$. When relaxing the assumptions and incorporating multiple discriminating factors into each message, the output O of the system can be modeled as a series of message *fingerprints* (e.g. vectors) $O = \{f_1, f_2, f_3\}$.[18]

We do not make assumptions about the distribution of user interaction over time. Instead, we provide generic results that can be applied to any system and user.

3.2 Effects of Relaxations

It might be infeasible to pad all messages to the exact same length. While the heartbeat message of a smoke detector only requires a single bit of information to be transmitted, the measurements of a weather sensor can be several bytes long. Also, parts of the communication might not be encrypted or it might be that addressing information is not hidden.

We therefore explicitly note the possibility of extending the attributes of a message beyond our model. For our analysis, we model messages as relative timestamps. One can however annotate each timestamp with a message length, a receiver address, a wireless channel ID or any other information visible to a passive adversary. The principles of our model and analysis still apply and only the numbers will differ in practice, as it is still possible to calculate probabilities of encountering certain message sequences.

3.3 Attacker Model and Privacy Goals

We assume that the attacker is interested in learning some information about the user. According to our model for (ε-δ)-private communication [18], this corresponds to the user performing one of two possible tasks T_i, T_j and the adversary observing the output of the system O as a series of message timestamps. The attacker then has to decide, based on the observation O, whether the user has indeed performed T_i or T_j.

We recall the definitions of the privacy goals [18]:

Definition 1. *A HAS provides (ε-δ)-private communication if there are constants $\varepsilon \geq 0$ and $0 \leq \delta < 1$, such that for any possible adversary-provided tasks T_i, T_j and for all possible adversarial observations O we have that*

$$Pr(O|T_i) \leq e^{\varepsilon} \times Pr(O|T_j) + \delta$$

(ε-δ)-private communication focuses on the probability of observing any given output. Aiming for this goal means that the system is as likely to generate one output as it is to generate any other, regardless of how the user is interacting with it.

Definition 2. *A HAS provides (ε-δ)-indistinguishability for a set of tasks $\mathbb{T}$ if there are constants $\varepsilon \geq 0$ and $0 \leq \delta < 1$, such that for all tasks $T_i, T_j \in \mathbb{T}$ and for all possible adversarial observations O we have that*

$$Pr(O|T_i) \leq e^{\varepsilon} \times Pr(O|T_j) + \delta$$

(ε-δ)-indistinguishability is a relaxation of (ε-δ)-private communication where the tasks that can be performed by the user are limited. Since (ε-δ)-private communication is hard to achieve, (ε-δ)-indistinguishability aims at being practically achievable in real-world settings.

Definition 3. *A HAS provides (ε-δ)-unobservability of a set of tasks $\mathbb{T}$ if*

$$\forall T \in \mathbb{T} : \overline{T} \in \mathbb{T}$$

(where $\overline{T}$ is the complementary task of T)and the system provides (ε-δ)-indistinguishability for $\mathbb{T}$.

(ε-δ)-unobservability is a special case of (ε-δ)-indistinguishability where the system aims to generate the same output pattern whether or not the user performs a given task from a limited set.

We further refine the definition of tasks and complementary tasks: A task T is a set of possible outputs $T = O_1, O_2, \ldots, O_n$. Given the set of all possible tasks $\mathbb{T}_{all} = \bigcup T$ the complementary task $\overline{T}$ is defined as $\overline{T} := \mathbb{T}_{all} \backslash T$. This means that the complementary task is the set of all outputs that are never generated by performing T but are possibly generated by other tasks.

We have also defined the privacy goals for continuous models by substituting the probability for the probability density functions. [18] This, however does not capture the same idea: A PDF can have a value of > 1 which conflicts with intuitive understanding of the privacy goals given the bound $\delta < 1$.

Therefore, we modify the definitions for continuous models as follows: The conditions must hold not for all adversarial observations O but for all adversarial observations O and all sets of adversarial observations $\mathbb{O}$. In place of the probability $Pr(O|T)$ from the discrete case we use the probability $Pr(O \in \mathbb{O}|T)$. The definition of (ε-δ)-private communication then reads as follows:

Definition 4. *A HAS provides (ε-δ)-private communication if there are constants $\varepsilon \geq 0$ and $0 \leq \delta < 1$, such that for any possible adversary-provided tasks T_i, T_j, for all possible adversarial observations O and for all sets of adversarial observations $\mathbb{O}$ we have that*

$$Pr(O \in \mathbb{O}|T_i) \leq e^{\varepsilon} \times Pr(O \in \mathbb{O}|T_j) + \delta$$

The attacker in our scenario is global and passive. They are able to receive all messages which are transmitted by any device. We explicitly disregard the possibility of node compromise, triangulation or wireless fingerprinting attacks, as these are separate areas of research and would extend beyond the scope of this paper.

4 Evaluation Data

For estimating the effect of CRDT and NED on the power consumption of HASs, we run the algorithms on our existing data sets. [16, 17]

System 1 is installed in a private apartment and consists of 23 HomeMatic devices: Temperature and humidity, door and window sensors, switches, smoke detectors, remote controls and a base station. Data was captured over 35 days. The system was used for everyday tasks such as controlling the lighting and monitoring the state of doors and windows.

System 2 consists of two parts. One is installed in a private home and consists of similar components as System 1 with the addition of door locks and corresponding remote controls. The second part is installed in an office space and further includes thermostats. Due to the

different nature and use cases, we treat these two parts as distinct systems: System 2.1 and System 2.2. The capture period is 8 days for System 2.1 and 13 days for System 2.2.

System 3 is installed in a private apartment. It consists of 30 custom components and assumingly uses an existing WiFi (IEEE 802.11) network for communication. The data capture consists of events rather than raw packets. Due to the nature of the installed devices, all events captured are direct results of user interaction. The capture period is 38 days.

5 Constant-Rate Dummy Traffic (CRDT)

CRDT offers (ε-δ)-private communication. [18] To achieve Constant-Rate Traffic throughout the HAS, two steps are necessary: First, genuine traffic has to be shaped to achieve a fixed maximum traffic rate. Then, times of inactivity must be padded with dummy traffic to achieve the same traffic rate. Our evaluation is based on exactly this mechanism. We assume an ideal CRDT scheme where during each timeslot, a dummy message is sent if and only if there is no genuine message to be transmitted.

The only variable in CRDT is the rate at which traffic is generated or permitted. A lower bound for this rate can be estimated by taking an exemplary use case: The user presses a switch and expects the light to turn on (or some other action to happen). The maximum acceptable value for this response time sets the minimum rate at which traffic must be permitted and generated. This value cannot necessarily be used directly, though. Due to the fact that multiple transmission requests may arise within a single timeslot, the actual reaction to the user's input may be delayed for more than a single period. Therefore, the rate has to be adjusted depending on the user and system behaviour.

Research in usability engineering suggests that a response time of 0.1 s is acceptable in most cases and a response time of more than 1 s is not acceptable.[3, 15, 20] To have a conservative estimate, we assume a maximum delay of 1 s for any message. While delays of more than 1 s might be acceptable for devices such as thermostats or temperature sensors, splitting these from the rest of the system has no effect on data rates. Ignoring these devices leads to the same minimum data rates.

5.1 Evaluation of CRDT

To measure the effect of CRDT on the power consumption of HASs, we apply the algorithm to the available dataset. The constraint of having a maximum delay of 1 s for genuine messages results in different data rates:[1] $4\,\mathrm{P\,s^{-1}}$ for System 1, $7\,\mathrm{P\,s^{-1}}$ for System 2.1, $5\,\mathrm{P\,s^{-1}}$ for System 2.2 and $18\,\mathrm{P\,s^{-1}}$ for System 3. Table 1 contains the detailed results.

5.2 Traffic Overhead and System Responsiveness

All Systems exhibit bursts of traffic. This becomes very obvious in System 2.2, which has the lowest traffic rate before introducing CRDT. However, the nature of the system explains the traffic pattern: The HAS consists of heating actuators installed in an office. In the morning, all heatings are turned on at the same time so the temperature is comfortable when the employees arrive. In the afternoon, the heatings are turned off again. This results in short bursts of traffic during these two phases, while for the rest of the day the system is rarely interacted with. Due to the fact that messages are to be delayed at most 1 s, the data rate has to be high enough to guarantee this upper bound during the bursts. However, the high rate has to be kept during the whole day, effectively introducing a traffic overhead which is orders of magnitude larger than the amount of genuine messages.

Decreasing the rate is no option: If we fix the rate to 1 packet/s, genuine messages are delayed for up to 59 s (in System 3), while the traffic still increases by a factor of almost 80 in the same system. The other systems do not perform significantly better.

Regarding the system responsiveness, the result is satisfying. The mean delay for genuine messages is 0.2 s for System 2.2 and lower than 0.1 s for all other systems. For System 2.2, delays of more than 0.1 s would likely happen when the heating actuators are automatically configured and only rarely be observed on user interaction.

As a preliminary conclusion, we see that CRDT introduces a significant traffic overhead if the system responsiveness is not to visibly deteriorate. While this confirms the first intuition, its actual impact on energy consumption is not as strong.

1 Within this paper, packets are assigned the unit P.

5.3 Energy Consumption

It is widely perceived that in terms of required energy, communication is more expensive than e.g. computations. Wander et al. have stated that "the power required to transmit 1 bit is equivalent to roughly 2090 clock cycles of execution on the microcontroller" [30]. By this principle, many protocols for Wireless Sensor Networks (WSNs) have been developed, minimising communication as far as possible. However, the cost of communication in HASs has not been analysed thoroughly yet. It is unclear whether e.g. the idle power consumption of the devices outweighs the sporadic bursts of communication so that there is little need for further optimisations of the traffic volume. Naturally, manufacturers try to optimise their systems' battery lifetimes and thus put effort in minimising the power consumption. However, they usually do not publish specifications or information on the focus of their research.

In order to estimate the increase in power consumption by CRDT, we apply five different models to our data. Feeney et al. have conducted measurements on regular 802.11 PC cards, which can be installed in laptops. [10] Van Dam et al. have implemented a power-saving MAC protocol for WSNs and evaluated its performance on *EYES* nodes—battery-powered devices using an energy efficient microcontroller and a wireless transceiver. [29] Wander et al. have evaluated the power consumption of public-key cryptographic protocols using the *Mica2dot* platform. [30] The Mica2dot platform uses the same transceiver as HomeMatic devices, which were used in two of our three analysed systems. Polastre et al. have developed a custom wireless node called *Telos* and compared its power consumption to others, including the Mica2dot. [25] For the fifth model, we have performed our own measurements on HomeMatic hardware.

Table 1 shows the condensed results of the evaluation. A detailed table with intermediate results is supplied in Appendix A. The following sections summarise these results and draw conclusions from them.

The five models lead to significantly different numbers for the total energy consumption, which can partly be explained by taking a closer look. Feeney et al.'s measurements were performed on laptop hardware. While laptops are built with energy efficiency in mind, the batteries are usually much larger than in HAS devices and the communication behaviour is significantly different. The other models use different kinds of hardware which were developed with particular use cases in mind.

System	1	2.1	2.2	3
Traffic				
Data Rate (Packets per second, $\mathrm{P\,s^{-1}}$)				
	4	7	5	18
Increase	267.53	148.54	5812.63	1453.40
Energy Consumption				
802.11 PC Card [10]				
Increase	5×10^{-4}	6×10^{-4}	1×10^{-3}	6×10^{-3}
EYES nodes [29]				
Increase	0.31	0.83	1.36	1.14
Mica2dot [30]				
Increase	5.33	9.13	17.52	14.12
Telos [25]				
Increase	0.44	0.98	1.02	2.28
HomeMatic				
Increase	2.31	5.37	8.68	10.12

Table 1. Effects of enforcing Constant-Rate Dummy Traffic in the sample installations under different energy consumption models. Numbers larger than 10^{-2} are rounded to two digits after the decimal point. The increase (both traffic and energy consumption) is given as a factor. A factor of 0 means no increase in traffic or energy consumption, whereas a 1 means that the original value doubled. Note that for System 3 using the 802.11 PC Card, a different idle power consumption was used.

5.3.1 802.11 PC Card, Feeney et al.

For the method presented by Feeney et al., we use the numbers of the 11Mbps WaveLAN PC Card as it has a lower power consumption than the 2Mbps model. For the calculations we use a packet size of 20 B—the same size used by van Dam et al.—for comparability. We also calculated the energy consumption using a packet size of 2048 B, but the conclusion is similar. With a packet size of 2048 B, the maximum increase when using CRDT is by a factor of 0.07 (or 7 %) and thus acceptable for most scenarios.

Feeney et al. did not take into account the time it takes to send a certain amount of data. They merely calculated the idle power consumption and then the additional energy consumption required to send a packet of a certain size. Thus, we can multiply the total sample time of each system by the idle power consumption. HomeMatic components as found in Systems 1 and 2 can communicate directly and without the need for a base station. We therefore use the "ad hoc" idle power consumption for these two systems. The exact setup of System 3 is unknown, but the publications suggest that an WiFi (IEEE 802.11) network is used. We therefore use the "BSS" mode idle power consumption for this setup.

In contrast to the other models, Feeney et al. have specifically measured the energy required to discard packets that are not sent to the receiving node. We have used this value for calculating the energy consumption of CRDT. It does however not play a significant role in the result, as the idle power consumption massively outweighs any other factor.

The transmission of broadcast messages has a lower energy consumption than the transmission of point-to-point messages. We therefore use the broadcast energy consumption for our calculations. For the calculations it does not matter which device is sending which packet. However, when one device is sending data, at least one other device is receiving it and the remaining (non-receiving) devices need to discard the packet. This model is the only one with precise data available on the energy required to discard a packet.

The results of the application of this model are decisive: The energy consumed during idle phases is orders of magnitude higher than the energy consumption of message transmissions—both genuine and dummy. Thus, the negative effect of CRDT is likely unobservable. If a Home Automation System is therefore implemented using similar hardware, CRDT can be implemented with a negligible impairment of battery lifetime and system responsiveness.

5.3.2 EYES Nodes, Van Dam et al.

Van Dam et al. have developed and implemented a novel power-saving MAC protocol using so-called EYES nodes. The published data on power consumption is brief, but can be used to get a rough estimate of the energy consumption of a Home Automation System. Based on the graphs in the paper, we estimate that the maximum transmission rate of EYES nodes using the T-MAC protocol is about $58.17\,\mathrm{P\,s^{-1}}$. Since the relation between transmission rate and power consumption does not appear to be linear, we interpolated the missing data using a second-degree polynomial based on the power consumption of $1\,\mathrm{P\,s^{-1}}$, $10\,\mathrm{P\,s^{-1}}$ and $58.17\,\mathrm{P\,s^{-1}}$.

We then calculated the original transmission rates for every device in the evaluated systems during each 1 s-timeslot. Here we assume that all transmissions originating from a single sender and being sent during the same timeslot are targeted towards the same receiver. Furthermore, we assume that no two devices transmit data to the same receiver during the same timeslot. While this might not be entirely accurate, we assume the error introduced by this to be negligible. This assumption is supported by the results of the other models.

Applying the consumption values from the model to this data results in an estimate of the energy consumed by the original system. We then introduce dummy traffic into the system at the given rates. While doing so, we distribute the dummy traffic evenly among all idle nodes, so that each node only sends at most one dummy packet during each 1 s-timeslot. This minimises the overall energy consumption and results in a more conservative estimate on the impact of CRDT.

The resulting total energy consumption of the systems is smaller than that of Feeney's model by a factor of almost 2000. However, due to the smaller idle power consumption, the impact of transmissions on the total consumption is much higher. The energy consumption of the system is increased by at least 31 % (System 1) and up to 136 % (System 2.2). The large discrepancy between the different systems in this model can be explained by examining the typical use cases. While System 1 is installed in an apartment and is frequently triggered by both user interaction and automation rules, System 2.2 almost solely handles bursts of automation rules after long periods of inactivity.

The *relative* difference between the different systems is similar to the one in Feeney et al.'s model. System 3 being an outlier can be explained by the different model variation used in the first calculation.

If hardware similar to EYES nodes is is used to implement a Home Automation System, it becomes necessary to develop alternative approaches to CRDT in order to protect the users' privacy.

5.3.3 Mica2dot, Wander et al.

In order to measure the energy-efficiency of different public-key cryptographic protocols, Wander et al. have performed measurements using the Mica2dot platform. The platform uses the same CC1100 transceiver as HomeMatic devices which make up 3 of our 4 analysed systems. When applying this data to our scenario, we make several assumptions: We assume the packet length is 20 bytes—similar to the other models. Furthermore, we have to use the different power consumptions for active and inactive microcontrollers. Therefore, we differentiate between two categories of devices:

Senders (e.g. light switches) only initiate communication themselves. They react to events such as the pressing of a button and then begin communicating with other devices. The processors of Senders can therefore

lie dormant for most of the time and only wake up when there is an event to be processed. This matches our observations during our own measurements which are described in Section 5.3.5.

Receivers (e.g. door locks or thermostats) react to messages from other devices. They therefore have to wake up periodically to check if there is a transmission to be processed and reacted upon. The exact wake-up strategy depends on the communication protocol and design decisions of the manufacturer. In our experiments we found out that HomeMatic Receivers wake up approximately once every 350 ms. Their processors are active about 0.5 % of the time. We apply this *duty cycle* to our calculations.

To calculate the energy consumption in the CRDT scenario, we assume that all dummy traffic is transmitted by Receiver nodes. This leads to a more conservative estimate than assuming Senders transmit all dummy traffic: Receivers exhibit a 0.5 % duty cycle anyway, so transmitting additional packets during this active time does not pose a large impact on power consumption. Sender devices, on the other hand, would have to wake up in order to be able to send packets, which further increases the energy consumption.

Calculating intermediate results for the use of CRDT reveals a peculiar effect: Transmitting 4 packets of 20 B each at a speed of 12.8 kbit s^{-1} takes about 0.05 s. This means that even if the dummy traffic generation is evenly distributed among all 9 Receivers and the genuine messages transmitted by Senders are subtracted, the Receivers have to spend more than 0.5 % of the total time transmitting data. As an intermediate conclusion, this means that a duty cycle of 0.5 % is not maintainable when applying CRDT. In practice, other tasks such as reading sensor values require further processor time, adding to the required duty cycle.

The impact of CRDT on the energy consumption according to Wander et al.'s model is enormous: For System 1, the energy consumption is five times that of an unmodified system. For System 2.2, it the factor is as high as 17. Increasing the length of the packets further increases the impact of CRDT. A packet size of 200 B leads to an increase factor of at least 45. In conclusion, CRDT is infeasible for systems using similar hardware.

5.3.4 Telos, Polastre et al.

Polastre et al. have developed a new type of wireless node and compared it to previous hardware such as the Mica2dot platform. Their goal was to create a more energy-efficient device. In order to apply the model to our data, we made similar assumptions as for the other models. We assumed a packet size of 20 B and—similarly to the application of Wander et al.'s model—split the devices into Senders and Receivers.

The results match the goals the Telos project: The energy consumption less than half of the consumption of the Mica2dot nodes. Surprisingly, the effect of CRDT on the overall energy consumption is also lower. However, the overall energy consumption when using CRDT is still 44 % to 228 % *higher* than the original energy consumption of the systems. This supports our thesis that CRDT is not feasible for use in specialised HASs.

The results also suggest that Telos is a viable technology for use in HASs in general. Among our evaluation, it exhibits the lowest energy consumption both before and after applying CRDT across all four tested systems.

5.3.5 HomeMatic Hardware

To check the models against hardware of an existing HAS, we performed our own measurements on HomeMatic hardware. We were able to confirm the most important aspects of power consumption by measuring and comparing the data with the other models. We could confirm our classification of devices as Senders and Receivers. Senders lie dormant most of the time with an idle power consumption of around 0.4 mW. Sending a (genuine) packet requires about 17.22 mJ of energy, including sensing the carrier before transmitting and listening for replies or collisions after the transmission. The energy consumption of a pressed switch where the different states are highlighted is supplied in Appendix A. Receivers, on the other hand, periodically wake up to listen for incoming transmissions. The duty cycle is 0.5 % and each spike requires about 80.79 µJ of energy in addition to the idle power consumption.

According to our measurements, the HAS systems consumed energy among the order of several kJ. This is not particularly high: A single alkaline AA battery can supply around 10 kJ of energy.

The impact of CRDT on the overall energy consumption is significant. In a "busy" setting such as System 1, the energy consumption is more than tripled. In a system handling bursts and long spans of inactivity such as System 2.2, the increase is nearly tenfold. While the impact on System 3 is the highest, this value has to be interpreted with care: The system is not built

from HomeMatic components and therefore actual values might differ from the model.

5.4 Conclusion of CRDT

From the analysis of CRDT power consumption we can draw two conclusions. The impact of CRDT on the power consumption strongly depends on the hardware used. On the one hand, there are systems where CRDT places a negligible strain on batteries and is therefore well suited to guarantee privacy. This is a strong contrast to the first intuition, which is that CRDT introduces too much traffic overhead to be feasible. On the other hand, many specialized systems with low idle power consumption are heavily impacted by CRDT. For those systems, different approaches need to be developed and implemented. The following section deals with one such approach.

6 Naive Exponential Dummies (NED)

The power consumption of systems using CRDT has made it clear that low latency and constant-rate traffic are incompatible for most HASs. We therefore relax our requirements on the privacy goals and present an approach using a probabilistic generation of dummy traffic. This approach introduces significantly less traffic overhead while introducing no latency to user interaction and offering (ε-δ)-unobservability for certain interactions. We call this approach *Naive Exponential Dummies (NED)*.

The approach works as follows: Genuine traffic is untouched by the system and is transmitted without delays. After every message (genuine or dummy), the system generates a random duration d from an exponential distribution with rate λ. If no genuine message is transmitted after this time d, a dummy message is generated and transmitted. If a genuine message does appear before, a new number is sampled from the same distribution. In reality, a system cannot choose the transmission time of a message with arbitrary precision (it is limited by the maximum transmission rate and other physical properties) and the precision with which the adversary can determine the timestamp of a captured message is also limited by the equipment. Therefore, a discrete model is more suitable than a continuous one. To convert the continuous exponential distribution into a discrete one, the system rounds down the drawn number to the nearest possible transmission time. Alternatively, it can draw a number directly from a geometric distribution with success probability $p = 1 - e^{-\lambda}$. Note that the algorithm does not necessarily exclude 0 as a possible outcome. If the system model does not allow multiple messages to be transmitted at the same time or in the same timeslot, the value 0 can be interpreted as the *next possible* transmission time rather than the current time.

If the timestamps are to be modelled as continuous or if the timestamp precision is too high to provide meaningful results, they can be transformed into a discrete model nevertheless. By using Apthorpe et al.'s approach [1], genuine traffic can be shaped into following a fixed maximum transmission rate without affecting system responsiveness, similar to the functioning of CRDT.

In this section we focus on the discrete model as it captures the properties of real systems better than the continuous one. However, since the goals are also defined for continuous models, similar analyses can be performed for those cases or for approaches which require an underlying continuous model. We assume that for the sending probability p of the geometric distribution, it holds that $0 < p < 1$. Furthermore we assume a geometric distribution with the possible outcomes $\mathbb{N} = \{1, 2, 3, \ldots\}$[2].

We can immediately deduce an important property of NED:

Theorem 1. *Running the algorithm on n different nodes simultaneously with sending probability $1 - \sqrt[n]{p}$ leads to the same distribution of inter-arrival times (time between subsequent messages) for dummy traffic as running it on one node with sending probability p.*

Proof. Let $X_1, \ldots, X_n$ be stochastically independent random variables following geometrical distributions with success probabilities $p_1, \ldots, p_n$ and let $X = \min\{X_1, \ldots, X_n\}$ be the combined random variable describing the overall inter-arrival times.

Then, since the geometric distribution is memoryless,

$$Pr(X = k) = Pr(X \geq k) \times Pr(X = k | X \geq k)$$

2 For the theoretical analysis in this paper we define the natural numbers to exclude 0.

$$
\begin{aligned}
&= Pr(X \geq k) \times (1 - Pr(X > k | X \geq k)) \\
&= Pr(X \geq k) \times (1 - Pr(X_1 > k \wedge \ldots \wedge X_n > k | X \geq k)) \\
&= Pr(X \geq k) \times (1 - \prod_{i=1}^{n} Pr(X_i > k | X_i \geq k)) \\
&= Pr(X \geq k) \times (1 - \prod_{i=1}^{n} (1 - p_i)) \\
&= Pr(X_1 \geq k \wedge \ldots \wedge X_n \geq k) \times (1 - \prod_{i=1}^{n} (1 - p_i)) \\
&= \prod_{i=1}^{n} (Pr(X_i \geq k)) \times (1 - \prod_{i=1}^{n} (1 - p_i)) \\
&= \prod_{i=1}^{n} (1 - p_i)^{k-1} \times (1 - \prod_{i=1}^{n} (1 - p_i)) \\
&= \left(\prod_{i=1}^{n} (1 - p_i) \right)^{k-1} \times (1 - \prod_{i=1}^{n} (1 - p_i))
\end{aligned}
$$

Therefore, X follows a geometric distribution with success probability $1 - \prod_{i=1}^{n} (1 - p_i)$. □

While this property is not groundbreaking, it serves an important purpose for practical system design: The sending probability p is the only parameter that needs to be synchronized between devices. Aside from this, the nodes can take decisions about the generation of dummy traffic locally and do not need to coordinate every transmission.

The sending probability p (or mean time between dummy messages λ) can be adjusted to reach a balance between privacy and energy efficiency. For $p = 0$, no dummy messages are generated so there is no impact on power consumption and none on privacy. For $p = 1$ NED generates CRDT at the maximum possible rate. We analyse the impact of $0 < p < 1$ on privacy guarantees and energy efficiency in the following sections.

6.1 Privacy Guarantees of NED

In this section we analyse NED with respect to its privacy guarantees. Note that while NED uses a geometric distribution for the generation of dummy traffic, no assumption is made about the distribution of genuine user interaction. The proofs in this section hold for any distribution of genuine events.

6.1.1 ε-δ-private Communication

As a first step, we prove that NED and, more generally, any approach which neither uses CRDT nor delays genuine messages cannot offer ε-private communication.

Theorem 2. *NED and any approach which neither uses CRDT nor delays genuine messages does not provide ε-private communication.*

Proof. Let S be the set of all possible genuine message timestamps. For each element $s \in S$, we define a task T_s = "interact with the system in any way that invokes a message with timestamp s". Such a task must exist because genuine messages are not delayed randomly. We also define a complementary task $T_{\bar{s}}$ = "interact with the system in any way so that no genuine message is generated at timestamp s".

Let $x \in S$ be a time at which a dummy packet is not necessarily generated (i.e. the probability of generating a dummy packet at x is less than 1). Such a time must exist since the dummy traffic generation scheme is not CRDT. Let $O_x = \emptyset$ now be an empty observation covering only the instant at time x. Then $Pr(O_x|T_x) = 0$ because executing T_x by definition invokes a message with timestamp x. However, $Pr(O_x|T_{\bar{x}}) > 0$ because executing $T_{\bar{x}}$ does not invoke a genuine message at time x. The latter probability is not necessarily 1, because the dummy traffic generation scheme *may* generate a dummy packet at time x.

There is no constant $\varepsilon > 0$ which satisfies $0 < Pr(O_x|T_{\bar{x}}) \leq e^{\varepsilon} \times Pr(O_x|T_x) = 0$. Consequently, the system does not offer ε-private communication.

For a continuous model, the same holds. Let $\mathbb{O}_x = O | x \notin \mathbb{O}$ be the set of all possible adversarial observations where no message appears at time x. Then by the same reason as above, $Pr(O \in \mathbb{O}_x|T_x) = 0$ and $Pr(O \in \mathbb{O}_x|T_{\bar{x}}) > 0$. Therefore, the system does not offer ε-private communication in the continuous model. □

The proof formally describes an intuitive but important property of NED and other latency-free approaches: If an attacker captures no messages within a certain time frame, they know that no task was executed that would have invoked a packet within this time frame. It also shows that in order to provide ε-private communication, any approach has to introduce artificial delays to genuine traffic, affecting the responsiveness of the HAS.

This is especially important if messages are not padded to a uniform length. If the goal is to provide ε-privacy for all classes of messages that an adversary

is able to distinguish, then CRDT has to be applied to each. For our test data, this would introduce an overhead larger than if all messages were padded to the maximum length.

For the discrete version of NED specifically we can also prove that it does not offer (ε-δ)-private communication. The proof follows the idea that it is possible to generate arbitrarily improbable adversarial observations (e.g. long chains of messages) so that any constant δ will eventually be surpassed. Intuitively, this corresponds to a scenario where a user continuously presses a light switch over and over again.

Theorem 3. *NED does not provide (ε-δ)-private communication under a discrete time model.*

Proof. For all natural numbers $n \in \mathbb{N}$ we define a task T_n ="interact with the system in a way so that n consecutive messages are generated" and a corresponding observation $O_n = [1, n] \cap \mathbb{N}$ comprising n consecutive messages and covering a duration of exactly n. By construction, it holds that $\forall n \in \mathbb{N} : \Pr(O_n|T_n) = 1$ We also define a task T_0 ="interact with the system in a way so that no genuine messages are generated". For this task it holds that $\forall n \in \mathbb{N} : \Pr(O_n|T_0) = p^n$ where p is the sending probability of NED.

Assuming that NED does offer (ε-δ)-private communications, there must be two constants $\varepsilon > 0, \delta < 1$ so that for all tasks T_i, T_j and for all observations O it holds that $Pr(O|T_i) \leq e^{\varepsilon} \times Pr(O|T_j) + \delta$.

For $m := \left\lceil \log_p \left(\frac{1-\delta}{e^{\varepsilon}} \right) \right\rceil$ it then holds that $Pr(O_m|T_m) = 1$ by the construction above. Due to the privacy guarantee, it holds that

$$\begin{aligned} 1 = Pr(O_m|T_m) \\ \leq \\ e^{\varepsilon} \times Pr(O_m|T_0) + \delta = e^{\varepsilon} \times p^m + \delta \end{aligned}$$

Due to the construction of m and since $p < 1$, it further holds that

$$e^{\varepsilon} \times p^m + \delta \leq e^{\varepsilon} \times \frac{1-\delta}{e^{\varepsilon}} + \delta = 1$$

However, since $p < 1$, we can come up with another sample O_{m+1} for which it holds that

$$\begin{aligned} e^{\varepsilon} \times Pr(O_{m+1}|T_0) + \delta \\ = e^{\varepsilon} \times p^{m+1} + \delta < e^{\varepsilon} \times p^m + \delta \end{aligned}$$

And since $e^{\varepsilon} \times p^m + \delta = Pr(O_m|T_0)$ as shown above, it holds that

$$Pr(O_m|T_m) > e^{\varepsilon} \times Pr(O_{m+1}|T_0) + \delta$$

This violates the condition of (ε-δ)-private communication. □

However, it is not always necessary for a HAS to offer (ε-δ)-private communication. If the user wants to e.g. only guarantee that an attacker is unable to find out whether they are at home, it might be sufficient to provide (ε-δ)-unobservability for a set which comprises tasks involving the user directly, such as opening doors and pressing switches (a reasonable number of times). While the traffic patterns triggered by these tasks might differ in practice depending on the system hardware and setup, we analyse some generalised use cases to demonstrate the feasibility of NED.

6.1.2 ε-indistinguishability

If a system does not equalise message sequences with regard to the number of messages and their inter-arrival times, approaches like NED that are not bounded cannot offer ε-indistinguishability for tasks invoking different message sequences. The theorem and its proof can be visualised using the following example: Suppose that pressing a light switch makes the system transmit 3 consecutive messages and that opening a door makes it transmit 4 messages. If the system now uses NED for the generation of dummy traffic, it is possible that after pressing a light switch, still only 3 messages are transmitted within the observed time frame. If an attacker captures this output, they can be certain that the light switch was pressed rather than the door being opened. Since ε-indistinguishability requires one of the tasks to be performed, this information is leaked to the attacker.

For the proof, we assume that performing a task T invokes a message sequence $S = \{s_1, s_2, \ldots, s_n\}$ where s_i is a random variable following a (usually, but not necessarily, bounded) probability distribution and describing the time difference between the i-th message and the point at which the task was performed. Note that using slightly different models (e.g. s_i describing the inter-arrival time between messages i and $i-1$) requires minor adaptions, but does not invalidate the proof.

Theorem 4. *Let T_A, T_B be two tasks and let $A = \{a_1, a_2, \ldots, a_m\}, B = \{b_1, b_2, \ldots, b_n\}$ be the message sequences invoked by performing T_A or T_B, respectively.*

If the tasks do not fully overlap, or formally if

$$\exists i \in [1,m] \forall x \in \mathbb{N}:$$
$$Pr(a_i = x) > 0 \Rightarrow \forall j \in [1,n] : Pr(b_j = x) = 0$$

then NED (or any other unbounded probabilistic approach) does not provide ε-indistinguishability for the set $\{T_A, T_B\}$.

Proof. Let i be such that $\forall x \in \mathbb{N} : (Pr(a_i = x) > 0 \Rightarrow \forall j \in [1,n] : Pr(b_j = x) = 0)$ (cf. the precondition). Let $Q \subseteq \mathbb{N}$ be the set for which $\forall q \in Q : Pr(a_i = q) > 0$, therefore $\forall q \in Q, j \in [1,n] : Pr(b_j = q) = 0$.

Then for all observations O where $Pr(O|T_B) > 0$ and $O \cap Q = \emptyset$ it holds that $Pr(O|T_A) = 0$. Such observations must exist because elements from Q are not generated by performing T_B and if the dummy traffic generation algorithm is NED or another unbounded probabilistic scheme, then there are observations O with $O \cap Q = \emptyset$ which may occur when performing T_B.

Consequently, there is no constant $\varepsilon \geq 0$ which satisfies $0 < Pr(O|T_B) \leq e^{\varepsilon} \times Pr(O|T_A) = 0$. The system therefore does not offer ε-indistinguishability for the set $\{T_A, T_B\}$. □

We can conclude that for strict ε-indistinguishability, the message sequences invoked by the tasks have to be modified so that no possible sequence of one tasks is impossible for another. In practice, this may be achieved by equalising the length of message sequences, e.g. by ensuring that after every user interaction, a fixed number of messages is transmitted in the same fixed or equally distributed intervals. Apthorpe et al. follow this idea in their approach named STP. [1]

6.1.3 ε-δ-unobservability

As a last step we analyse NED with regard to (ε-δ)-unobservability. We show that NED achieves this goal and calculate the values of ε and δ for a given scenario. For the proof we assume the following case: As in the previous section, performing a given task T invokes a number of messages $S = \{s_1, s_2, \ldots, s_n\}$ where each s_i is a random variable following any probability distribution and describing the timing of message i relative to the task's execution time. Our S matches the set E of interesting events or messages from our previous work [18].

Theorem 5. *Let T be a task invoking a sequence of messages $S = \{s_1, s_2, \ldots, s_n\}$ and let $\overline{T}$ be the complementary task invoking no genuine message. Then NED offers (ε-δ)-unobservability of $\{T, \overline{T}\}$.*

Proof. Let O be any adversarial observation of duration l (measured in possible transmission slots). We distinguish between the following (possibly overlapping) cases:

1. $Pr(O|T) > 0$ Then O will unconditionally include a number $m, 0 \leq m \leq \min(n, l)$ of genuine messages generated by executing T and a number $d = |O \cap D|$, $0 \leq d \leq l$ of dummy messages. An upper bound for this probability $Pr(O|T)$ can be computed by calculating the probability of observing exactly the same d dummy messages in an observation of length l which already includes the m genuine messages:

$$Pr(O|T) \leq p^d \times (1-p)^{l-m-d}$$

Then, the probability of observing the same pattern with no genuine messages is

$$Pr(O|\overline{T}) = p^m \times p^d \times (1-p)^{l-m-d}$$

If we plug these two terms into the equation for (ε-δ)-unobservability, we get

$$\begin{aligned} &Pr(O|T) \leq p^d \times (1-p)^{l-m-d} \\ \leq &e^{\varepsilon} \times Pr(O|\overline{T}) + \delta \\ = &e^{\varepsilon} \times p^m \times p^d \times (1-p)^{l-m-d} + \delta \end{aligned}$$

We see that for $\varepsilon = -\ln(p^m) > 0$ (since $0 < p < 1$ and therefore $p^m < e$) and $\delta = 0$ the two sides become the same, satisfying the condition. Since $m \leq n$, the value $\varepsilon = -\ln(p^n)$ is a lower bound.
For the opposite direction, we need a lower bound for the probability $Pr(O|T)$. Since the probability of any timestamp being yielded by NED is larger than 0, the probability of observing any message cannot be lower than the probability of this message being a dummy generated by NED. This value is the sending probability p. Therefore, a lower bound is $Pr(O|T) \geq p^{m+d} \times (1-p)^{l-m-d}$. Inserting this into the equation we get

$$\begin{aligned} &Pr(O|\overline{T}) = p^m \times p^d \times (1-p)^{l-m-d} \\ \leq &e^{\varepsilon} \times Pr(O|T) + \delta \geq p^{m+d} \times (1-p)^{l-m-d} \end{aligned}$$

We immediately see that the condition is satisfied for $\varepsilon = \delta = 0$ and $\delta = 0$.

2. $Pr(O|T) = 0$ Then the equation $Pr(O|T) \leq e^{\varepsilon} \times Pr(O|\overline{T}) + \delta$ is satisfied for any value of $\varepsilon > 0$ and $\delta < 1$.
   As for the opposite direction, it is obvious that $Pr(O|\overline{T}) > 0$ since NED can yield any set of timestamps. The maximum possible value of this probability is therefore an upper bound for the constant δ. This maximum is trivial to compute: Since $Pr(O|T) = 0$, O must not have a packet over a period where T would generate one. An upper bound for observing this "silence" is $1 - p$. Hence, we get that $\delta \leq 1 - p$.

□

7 Evaluating NED

In order to evaluate NED against real HAS data, we implemented the algorithm and ran it against our evaluation data. During the implementation we made the following design decisions:

– The algorithm generates samples from an exponential distribution and rounds the result down to the nearest possible transmission time.
– Since our sample data contains messages with the same timestamp due to limitations of the capturing hardware, we allowed 0 as a possible result for the next dummy message.

The sample data of each system was used as a single trace of genuine traffic. The generated dummy traffic was added on top of the genuine messages similar to how NED could be implemented in practice. As stated in Section 6, we placed no assumption on the distribution of genuine traffic. Instead, we used the realistic sample data as-is.

For each system, we generated dummy traffic using six different values for the mean inter-arrival time λ of the exponential distribution. We also performed an analysis using no dummy traffic. We generated up to 1000 observations from the resulting system output, making sure that at least 40 of them included user interaction. The duration of these observations was set to 10 s. We performed the evaluation with longer samples as well, but reached the same conclusion. For each sample, we counted the number of times it occured in the system output including dummy messages. The occurences were split between those where the same kind of user interaction happened as in the sample and those where different or no user interaction was recorded. We ran the complete simulation multiple times to see if the calculated values are stable. Since NED can generate extreme traffic patterns (no or very high dummy traffic), it is possible to see high variations in the results, although with a low probability. The results below are the average of multiple runs; the values did not differ significantly.

Using this data we estimate values for ε and δ. We also estimate the effect on the energy consumption according to the HomeMatic model from Section 5.3.5. The results are summarised in Table 2.

System	1	2.1	2.2	3
No dummy traffic				
TI	0.00	0.00	0.00	0.00
ε	8.77	7.62	7.83	7.63
δ	0.87	0.70	0.99	1.00
$\lambda = 0.5$ (∼1 P every 2 s)				
TI	43.55	13.84	754.83	40.40
ε	3.91	$<10^{-10}$	$<10^{-10}$	3.97
δ	5.93×10^{-3}	4.78×10^{-3}	7.11×10^{-3}	6.71×10^{-3}
ECI	0.38	0.50	1.13	0.28
$\lambda = 1$ (∼1 P every second)				
TI	115.39	36.69	1995.63	80.85
ε	$<10^{-10}$	$<10^{-10}$	$<10^{-10}$	$<10^{-10}$
δ	5.56×10^{-5}	1.46×10^{-3}	8.55×10^{-4}	7.50×10^{-9}
ECI	1.00	1.33	2.98	0.56
CRDT (for reference, rates from Sec. 5.1)				
TI	267.53	148.54	5812.63	1453.40
ε	0	0	0	0
δ	0	0	0	0
ECI	2.31	5.37	8.68	10.12

Table 2. Results for using NED in Home Automation Systems. TI stands for traffic increase and ECI means Energy Consumption Increase according to the HomeMatic model. Both are given as a factor, where a value 0 means no increase and a value of 1 means that the original value doubled. TI and ECI for CRDT is supplied for comparison. Since CRDT offers (0-0)-private communication [18], the values for ε and δ are 0.

NED offers significant privacy improvements for moderate increases of overall traffic. For higher values of λ, the traffic overhead increases quickly and so do the privacy guarantees. When NED is configured to send approximately one packet every second, the parameter ε approaches 0 in all four systems. The constant privacy leakage δ also becomes diminishingly small. However, the energy consumption is merely doubled for the first two systems. For System 3, the increase is only 50 %.

Even low values of λ alread provide significant improvements over non-anonymised systems: When transmitting one dummy every 10 seconds, the constant leakage δ is below 0.5 in all systems, while the increase in energy consumption is below 20 % for all four systems and below 10 % for three of them.

A suitable compromise between privacy and energy consumption is at $\lambda = 0.5$. ε drops to at most half the value it has in the unmodified systems. Due to it being in the exponent when comparing probabilities, the factor e^{ε} drops by more than 97 %. Furthermore, the constant leakage δ drops below 0.01 in all systems, which means that the chance of an attacker being able to learn a definitive piece of information is close to zero. The increase in energy consumption is between 6 and 50 % for all systems except System 2.2.

7.1 Behaviour of ε and δ over Time

We have investigated how ε and δ develop over different stretches of time. First we consider a simple theoretic case: A user performs an interaction at the same time every day. The interaction invokes a chain s of n consecutive messages. The probability of observing s at this time each day is 1. The probability of observing n consecutive dummies in absence of the interaction is p^n. The probability of observing a sample of length n with at least one gap in the absence of interaction is $1 - p^n$. Thus, for one day the privacy parameters are $\delta = 1 - p^n$ and $\varepsilon = -n \times \ln p$ ((ε-δ)-unobservability).

When monitoring the system for multiple days (or task execution periods), however, the adversary observes the same pattern at the same time. Thus, after k days of monitoring, the probability of observing the same pattern of n consecutive dummy messages at the same time each day is p^{nk} and the probability of observing a single gap is $1 - p^{nk}$. This means that the privacy parameter $\varepsilon = -nk \times \ln p$ linearly rises with k while $\delta = 1 - p^{nk}$ converges to 1. As an example, for $\lambda = 0.5 \Rightarrow p \approx 0.39$, $n = 4$, ε starts at $\varepsilon \approx 3.73$ for $k = 1$ and rises to $\varepsilon \approx 26.12$ for $k = 7$. δ starts at $\delta \approx 0.61$ and decreases to $\delta \approx 0.03$ for $k = 7$.

We have also tried to extract the behavior of ε and δ over time from the sample data. However, we observe that tasks are not as regular as in our theoretical example: With $\lambda = 0.1$, the parameter ε for System 3 starts at 2.97 for a timeframe of 6 hours and rises to 9.45 when taking a week's traffic data. For the full dataset it then decreases back to 7.50. This can be explained when examining the parameter calculation: For a given interaction in a short timeframe, a particular sample is unique. When observing longer stretches of time however, the same interaction results in different samples. Furthermore, short timeframes result in a high variance of the privacy parameters due to the limited data. In order to properly analyze real behavior of ε and δ over time, larger data sets are needed. The theoretical and practical values have been plotted in Figure 1, although the explanatory power of this graph is limited due to the aforementioned constraints.

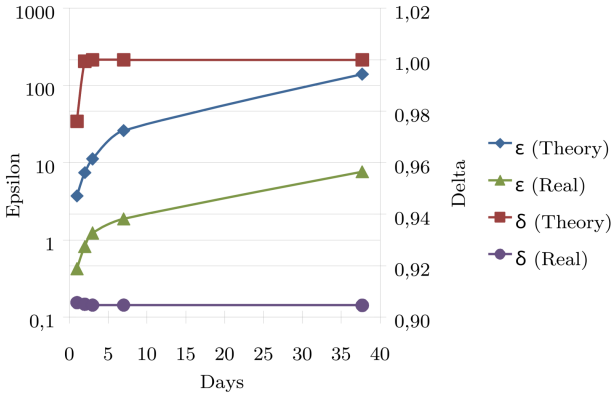

Fig. 1. Simulated and measured development of ε and δ over time ((ε-δ)-unobservability). The Scale for ε is logarithmic for increased readability. Values are calulcated for $\lambda = 0.5$ and real values are estimated from the data of System 3. Note that the real values are subject to limited data and

8 Conclusion and Outlook

In this paper we have performed a qualitative and quantitative analysis of countermeasures against traffic analysis in Home Automation Systems. We have shown that Constant-Rate Dummy Traffic is a feasible countermeasure under certain conditions and using suitable hardware. As an alternative for systems where CRDT does not perform well, we have introduced NED as a robust algorithm. It features a single, tunable parameter that can be configured to provide any amount of privacy or energy-efficiency.

On one hand, we have proven fundamental limits of latency-free privacy mechanisms using NED as an example. On the other hand, we have proven that NED offers (ε-δ)-unobservability for single user interactions and have evaluated the approach using sample data from four different HASs. We have shown that NED offers

reasonable privacy guarantees at moderate energy consumption overhead.

In order to further reduce the impact on overall power consumption, systems could offer to change the parameters of CRDT and NED according to time. This would strongly affect settings such as System 2.2, where bursts of activity are followed by long periods of little to no traffic. An adversary could then see an overall increase and decrease in activity, but could not identify if a burst was artificial or when there was user interaction during a period of higher activity.

Furthermore, insight into behavioral patterns of HAS users can shed more light onto the performance of NED and similar approaches in a realistic setting. While a study of user behavior is beyond the scope of this paper, it would enable the development of more optimized dummy traffic generation schemes. While NED introduces no latency to user interaction, some minor latencies might be acceptable depending on the use case.

Given the fact that NED builds on a very general and intuitive system model, the principle can be applied to other settings as well. The model allows for formal proofs of privacy guarantees and different algorithms can be evaluated and compared to NED.

9 Acknowledgements

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the Smart Home Private with Smart (er) IoT Traffic Shaping.

[2] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A Block Cipher for Low Energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015. ASIACRYPT 2015*, pages 411–436. Springer, Berlin, Heidelberg, nov 2015.

[3] Calum Benson, Adam Elman, and Seth Nickell. GNOME Human Interface Guidelines 2.2.3, 2012.

[4] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a Distance: Website Fingerprinting Attacksand Defenses. In *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*, page 605, New York, New York, USA, 2012. ACM Press.

[5] Haowen Chan and Adrian Perrig. Security and privacy in sensor networks. *Computer*, 36(10):103–105, oct 2003.

[6] Mauro Conti, Jeroen Willemsen, and Bruno Crispo. Providing Source Location Privacy in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 15(3):1238–1280, 2013.

[7] Bogdan Copos, Karl Levitt, Matt Bishop, and Jeff Rowe. Is Anybody Home? Inferring Activity From Smart Home Network Traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*, pages 245–251. IEEE, may 2016.

[8] Debajyoti Das, Sebastian Meiser, Esfandiar Mohammadi, and Aniket Kate. Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency - Choose Two. In *Proceedings - IEEE Symposium on Security and Privacy*, 2018.

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[10] Laura Marie Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings IEEE INFOCOM 2001 – Conference on Computer Communications – Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1548–1557, Anchorage, Alaska, 2001. IEEE.

[11] Ben Greenstein, Tadayoshi Kohno, Damon McCoy, Srinivasan Seshan, Jeffrey Pang, and David Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *MobiSys'08 - Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, 2008.

[12] Csaba Kiraly, Simone Teofili, Giuseppe Bianchi, Renato Lo Cigno, Matteo Nardelli, and Emanuele Delzeri. Traffic Flow Confidentiality in IPsec: Protocol and Implementation. In *The Future of Identity in the Information Society*, pages 311–324. Springer US, Boston, MA, 2008.

[13] Patrick Leu, Aanjhan Ranganathan, Ivan Puddu, and Srdjan Čapkun. I Send, Therefore I Leak: Information Leakage in Low-Power Wide Area Networks. In *Proceedings of the 11th ACM Conference on Security \& Privacy in Wireless and Mobile Networks*, pages 23–33, 2018.

[14] Alfredo Matos, Rui L. Aguiar, João Girao, and Frederik Armnecht. Toward dependable networking: secure location and privacy at the link layer. *IEEE Wireless Communications*, 15(5):30–36, oct 2008.

[15] Robert B. Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 267–277, San Francisco, 1968. ACM New York.

[16] Frederik Möllers, Sebastian Seitz, Andreas Hellmann, and Christoph Sorge. Extrapolation and Prediction of User Behaviour from Wireless Home Automation Communication. In *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks - WiSec '14*, pages 195–200, New York, New York, jul 2014. ACM Press.

[17] Frederik Möllers and Christoph Sorge. Deducing User Presence from Inter-Message Intervals in Home Automation Systems. In Jaap-Henk Hoepman and Stefan Katzenbeisser, editors, *ICT Systems Security and Privacy Protection: 31st IFIP TC 11 International Conference, SEC 2016, Ghent, Belgium, May 30 - June 1, 2016, Proceedings*, pages 369–383,

Cham, 2016. Springer International Publishing.

[18] Frederik Möllers, Stephanie Vogelgesang, Jochen Krüger, Isao Echizen, and Christoph Sorge. Modelling Traffic Analysis in Home Automation Systems. In Srdjan Capkun and Sherman S. M. Chow, editors, *Cryptology and Network Security: 16th International Conference, CANS 2017, Hong Kong, China, November 30—December 2, 2017, Revised Selected Papers*, pages 526–536. Springer International Publishing, 2017.

[19] Thomas Mundt, Andreas Dähn, and Hans-Walter Glock. Forensic analysis of home automation systems. In *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*, 2014.

[20] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1 edition, 1993.

[21] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. Do Dummies Pay Off? Limits of Dummy Traffic Protection in Anonymous Communications. In Emiliano De Cristofaro and StevenJ. Murdoch, editors, *Privacy Enhancing Technologies SE - 11*, volume 8555 of *Lecture Notes in Computer Science*, pages 204–223. Springer International Publishing, 2014.

[22] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website Fingerprinting in Onion Routing BasedAnonymization Networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society - WPES '11*, page 103, New York, New York, USA, 2011. ACM Press.

[23] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead. In Wolfgang Effelsberg, Hans W. Meuer, and Günter Müller, editors, *Kommunikation in verteilten Systemen – Grundlagen, Anwendungen, Betrieb GI/ITG-Fachtagung, Mannheim, 20.–22. Februar 1991, Proceedings*, volume 267 of *Informatik-Fachberichte*, pages 451–463. Springer-Verlag Berlin Heidelberg, 1991.

[24] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. The Loopix Anonymity System. In *Proceedings of the 26th USENIX Security Symposium*, pages 1199–1216, Vancouver, BC, Canada, 2017. USENIX Association.

[25] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, pages 364–369, Boise, ID, 2005. IEEE.

[26] Min Shao, Yi Yang, Sencun Zhu, and Guohong Cao. Towards Statistically Strong Source Anonymity for Sensor Networks. *ACM Transactions on Sensor Networks (TOSN)*, 9(3):34:1–34:23, apr 2008.

[27] Vitaly Shmatikov and Ming-Hsiu Wang. Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In Dieter Gollmann, Jan Meier, and Andrei Sabelfeld, editors, *Computer Security – ESORICS 2006 – 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006, Proceedings*, volume 4189 of *Security and Cryptology*, pages 18–33, Hamburg, 2006. Springer-Verlag Berlin Heidelberg.

[28] Raphael R. Toledo, George Danezis, and Ian Goldberg. Lower-Cost ε-Private Information Retrieval. *Proceedings on Privacy Enhancing Technologies*, 2016(4):184–201, 2016.

[29] Tijs van Dam and Koen Langendoen. An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pages 171–180, Los Angeles, California, 2003. ACM.

[30] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheueling Chang Shantz. Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks. In *Third IEEE International Conference on Pervasive Computing and Communications*, pages 324–328. IEEE, 2005.

[31] Yi Yang, Min Shao, Sencun Zhu, Bhuvan Urgaonkar, and Guohong Cao. Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks. In *Proceedings of the first ACM conference on Wireless network security - WiSec '08*, pages 77–88, New York, New York, 2008. ACM Press.

A Energy Efficiency of CRDT

Table 3 summarizes the evaluation of our CRDT analysis.

Figure 2 shows the energy consumption of a HomeMatic switch. The different states which we could identify are highlighted.

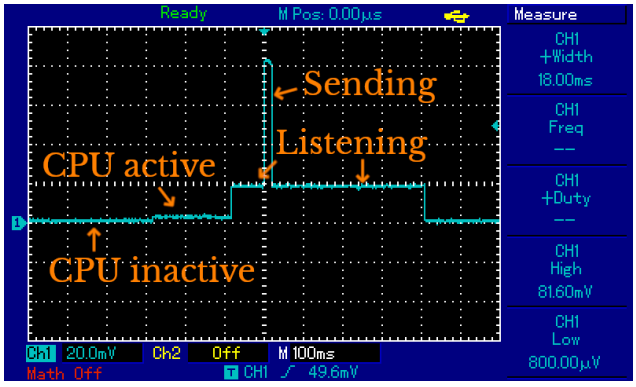

Fig. 2. Measurement of the voltage dropping across a $2\,\Omega$ shunt resistor connected in series to a HomeMatic light switch which was pressed. One can clearly identify the different states of the hardware. The switch is powered by two AAA batteries or—in this case—a laboratory power supply serving $3\,\mathrm{V}$.

B Evaluation Reproducibility

In order to allow other researchers to reproduce the results obtained in this work, we have published the

System	1	2.1	2.2	3
Unmodified Data				
Timespan (days)	35.49	8.33	13.44	37.72
Messages	45,679	33,708	999	40,336
Inter-Arrival Times				
Minimum	0 s	0 s	0 s	0 s
Maximum	3668 s	153 s	61,645 s	25,050 s
Mean	67.14 s	21.36 s	1163.89 s	80.80 s
Median	64 s	14 s	0 s	0 s
Standard Deviation	49.67 s	27.49 s	5875.77 s	469.68 s
After applying CRDT				
Data Rate ($\mathrm{P\,s^{-1}}$)	4	7	5	18
Genuine Messages Delayed	4.95 %	26.09 %	59.46 %	51.90 %
Traffic Increase (Factor)	267.53	148.54	5812.63	1453.40
802.11 PC Card [10] (20 B, 11 $\mathrm{Mbit\,s^{-1}}$)				
Idle	52.26 MJ	8.54 MJ	6.03 MJ	4.69 MJ
Increase by CRDT (Factor)	4.55×10^{-4}	6.11×10^{-4}	1.29×10^{-3}	5.71×10^{-3}
EYES nodes [29]				
Unmodified	29.68 kJ	4.81 kJ	3.37 kJ	4.05 kJ
Increase by CRDT (Factor)	0.31	0.83	1.36	1.14
Mica2dot [30] (20 B, 0.5 % Duty Cycle for Receivers, 0 % for Senders)				
Unmodified System	2600.36 J	634.89 J	383.55 J	5179.13 J
Increase by CRDT (Factor)	5.33	9.13	17.52	14.12
Telos [25] (20 B, 0.5 % Duty Cycle for Receivers, 0 % for Senders)				
Unmodified System	1473.50 J	272.22 J	191.37 J	2395.58 J
Increase by CRDT (Factor)	0.44	0.98	1.02	2.28
HomeMatic				
Unmodified System	37.93 kJ	6.69 kJ	4.29 kJ	52.21 kJ
Increase by CRDT (Factor)	2.31	5.37	8.68	10.12

Table 3. Effects of enforcing Constant-Rate (Dummy) Traffic in the sample installations under different energy consumption models. Numbers are rounded to two digits after the decimal point. A factor of 0 means no increase in traffic or energy consumption, whereas a 1 means that the original value doubled. Note that for System 3 using the 802.11 PC Card, a different idle power consumption was used.

source code of our NED implementation as well as the HAS sample data. The data can be accessed at https://github.com/frederikmoellers/ned-eval/.