Are you (Google) Home? Detecting Users' Presence through Traffic Analysis of Smart Speakers

D. Caputo[1], L. Verderame[1], A. Merlo[1], A. Ranieri[2], and L. Caviglione[2]

[1] Computer Security Lab
Department of Informatics, Bioengineering, Robotics and Systems Engineering
University of Genoa
{davide.caputo, alessio}@dibris.unige.it, luca.verderame@unige.it
[2] Institute for Applied Mathematics and Information Technologies
National Research Council of Italy
{andrea.ranieri, luca.caviglione}@ge.imati.cnr.it

Abstract

Smart speakers and voice-based virtual assistants are core building blocks of modern smart homes. For instance, they are used to retrieve information, interact with other devices, and command a variety of Internet of Things (IoT) nodes. To this aim, smart speakers and voice-based assistants typically take advantage of cloud architectures: vocal commands of the user are sampled, sent through the Internet to be processed and transmitted back for local execution, e.g., to perform an automation task or activate an IoT device. Even if privacy and security is enforced by means of encryption, features of the traffic, such as the throughput, the size of protocol data units or the IP addresses, can leak important information about the habits of the users as well as the number and the type of IoT nodes deployed. In this perspective, the paper showcases risks of machine learning techniques to develop black-box models to automatically classify traffic and implement privacy leaking attacks. We prove that such traffic analysis allows to detect the presence of a person in a house equipped with a Google Home device, even if the same person does not interact with the smart device. Experimental results collected in a realistic scenario are presented and possible countermeasures are discussed.

1 Introduction

Smart speakers and voice-based virtual assistants are important building blocks of modern smart homes. For instance, they can ben used to retrieve information, interact with other devices, and command a wide range of Internet of Things (IoT) nodes. Moreover, they can be used as hubs for managing IoT deployments or implementing device automation services, e.g., to perform routines in smart lighting or provide remote connectivity for domestic appliances. According to [17], there are over 200 million of smart speakers installed in private properties (with the wide acceptation inside private houses and small office settings), and the trend is expected to culminate in 2030 when the number will exceed 500 million of units. In general, smart speakers and voice-based virtual assistants take advantage of cloud-based architectures: vocal commands of the user are sampled and sent through the Internet to be processed. As a result, the smart speaker or the appliance running the virtual assistant receives a textual representation as well as optional, companion multimedia data. Then, it executes the command or route it to a proper hub, e.g., to communicate via ZigBee or Bluetooth links with IoT nodes. To enforce privacy and security, the prime mechanism is the encryption of traffic (see, e.g., reference [27] and references therein). However, features of the flows such as, the throughput, the size of protocol data units or `(address, port)` tuples, can leak important information about the habits of the users [8] or the number and the type of IoT nodes [4, 22]. As a consequence, an attacker can collect traffic

from the local IEEE 802.11 wireless loop or between the home gateway and the Internet and then try to guess the type of IoT nodes and the state of sensors and actuators. With such a knowledge, the malicious entity can launch a wide array of offensive campaigns, such as profile users, plan attacks to the physical space or perform social engineering campaigns [4, 22].

Despite the underlying technology or the complexity of the deployment, there is an increasing interest in investigating risks arising from the statistical analysis of the traffic exchanged by a smart speaker and the cloud. For instance, in [4] authors showcase how passive network analysis can be used to identify devices and correlate some user activities, e.g., traffic flows produced by switches and health monitors can leak the sleep cycle of a user. In [12], the traffic produced by state transitions of home devices (i.e., a thermostat and a carbon dioxide detector) can be used to infer if a user is present in the home. Such idea is further refined in [1], where passive measurements are used to develop models of the daily routine of individuals (e.g., leaving/arriving home). Concerning works aiming at identifying devices, possibly by adopting machine learning or statistical tools, in [22] several machine learning techniques are used to identify IoT devices by exploiting "poor" information like the length of packets produced during normal operations. Additionally, in [3] the risks of HTTP-based communications are discussed, both from the perspective of inferring data about the devices (e.g., the state or the intensity of a light source) and performing session-highjacking attacks.

In addition, sensitive data contained in IoT nodes and smart speakers can be relevant for forensics investigations [18], and traffic patterns can be manipulated by malware to exfiltrate data, for instance through information hiding schemes [13] or covert channels [10, 19]. In this vein, the paper discusses risks of machine learning techniques to develop black-box models for automatically classifying traffic and to implement privacy leaking attacks. Differently from previous works [1, 3, 12, 22], we focus on understanding whether it is possible to recognize the presence of a user when no queries are performed. In fact, when a request is sent towards the Internet, the produced traffic volumes or the appearance of specific network addresses trivially leak the presence of a human operator in the house. To this aim, we empirically prove how it is possible to detect the presence of a person in a house by analysing the traffic produced by a Google Home device under the assumption that the person is not interacting with it. Nevertheless, attention will be devoted in proposing ideas to mitigate such kind of threats by acting at the traffic level. In fact, the design of suitable mitigation techniques is often neglected (see, e.g., [3] for a notable exception) or addressed at an API-permission level [2], which is definitely out of the scope of the paper.

Summing up, the contributions of this work are: *i*) to review the architectural blueprint used by smart speakers and voice-based virtual assistants and elaborate an effective model to conduct privacy leaking attacks, and *ii*) to evaluate the effectiveness of using machine learning techniques for black-box modelling of traffic. We also sketch some design rules to mitigate identification attacks in Section 5.

The remainder of the paper is structured as follows. Section 2 discusses the general architecture used by smart speakers to control IoT devices, introduces the threat model and the machine learning mechanisms that can be exploited by the attacker. Section 3 deals with the testbed used to collect data, while Section 4 presents numerical results and Section 5 concludes the paper and showcases some possible future directions.

2 Smart Speakers: Architecture and Threat Model

As hinted, smart speakers and voice-based virtual assistants are a core foundation for smart homes. In essence, they provide a user interface to issue requests or commands in a natural

manner, i.e., by simply talking. Such devices can be also used as hubs for other IoT nodes and network appliances or to perform tasks like playing music and video, purchasing items, and to make recommendations. Besides, smart speakers and virtual assistants can provide a variety of information including directions and weather forecasts.

As today, the most popular smart speakers implementing the aforementioned features are Google Home[1], HomePod[2] and Amazon Echo[3], whereases virtual assistants are Amazon Alexa[4], Apple Siri[5] and Google Assistant[6]. Literature still lacks of a unified terminology for this class of devices and services. In fact, smart speakers and virtual assistants are identified as intelligent personal assistants, virtual personal assistants, home digital voice assistants, voice-enabled speakers, smart speakers, and voice-based virtual assistants, just to mention the most popular names. Therefore, in the following, we only use the terms smart speakers or Intelligent Virtual Assistant (IVA) interchangeably, except when doubt may arise.

Even if each smart speaker is characterized by specific design choices and some setups are implemented via a complex interplay of technologies and services, the core architectural blueprint is quite standard and depicted in Figure 2. The overall set of components is often defined as the *ecosystem* as to emphasize the end-to-end pipeline at the basis of such services, i.e., hardware or software entities allowing the interaction of end users, computing and communication services, and software running in IoT nodes. Even if each vendor usually implements its own blueprint, the typical one is composed of four major components:

- **Smart Speaker** or **IVA**: it is in charge of collecting vocal commands, sample them and transmit the data trough the Internet to a backend. Upon receiving a response, the smart speaker or the software IVA agent can provide a feedback to the user or directly interact with other devices. For instance, the smart speaker could start the playback of a music stream received from a Content Delivery Network (CDN) or send through a ZigBee link a command to a smart lightbulb. In some cases, it can also act as a sort of "router", thus delivering commands to the suitable hub. To avoid security and privacy threats, communications are encrypted via the Secure Socket Layer (SSL) [8, 15].

- **Client** and **IoT Devices**: they are the targets of commands of the ecosystem. Typical nodes deployed in a smart home are sensors, actuators, Bluetooth/ZigBee bridges, wireless speakers or IoT-capable appliances. As previously said, some entities belonging to this class can be colocated within the smart speaker.

- **IVA Cloud**: it is the backend in charge of processing data and delivering back text/binary representations of commands to be executed, including additional contents like multimedia streams, geographical information or JSON files containing a composite variety of information. With the advent of open ecosystems promoting the interaction among services provided by multiple vendors, the borders of the IVA cloud are blurring [8], [10], [15]. For instance, vocal stimuli could be processed in a datacenter and sent back to the IVA while contents can be delivered by a third-part CDN and some IoT nodes could establish a direct point-to-point connection with the computing infrastructure of their manufacturer.

[1] https://store.google.com/product/google_home
[2] https://www.apple.com/homepod/
[3] https://www.amazon.com/echodot
[4] https://developer.amazon.com/alexa
[5] https://developer.apple.com/siri/
[6] https://assistant.google.com/

- **Network**: it connects the smart speaker or the IVA with IoT nodes as well as the Internet. Typical deployments use a single local (wireless) network connected via a router/gateway to the Internet. However, in most complex scenarios, different networks could be present, e.g., a local access cabled network for some IoT nodes and hubs and multiple wireless loops to connect smart devices and grant access to the user via a smartphone. Concerning protocols used to exchange data between the IVA and the cloud, the TCP is the main choice, with the multipath variant to optimize performances and reduce delays [8]. A notable exception is the Google ecosystem. In fact, it exploits QUIC [6], a protocol originally engineered to improve performance issues of HTTP/2 and based on transport streams multiplexed over UDP. We point out that the presence of QUIC can represent a signature to ease the identification of the ecosystem (e.g., Apple HomeKit vs. Google). However, this requires to understand its behaviors, which can be highly influenced by the underlying network conditions (see, e.g., [7] for a sensitivity/performance analysis of the SPDY counterpart in different wireless settings).

Concerning the typical usage scenario, smart speakers rely upon a microphone to sense commands, which are processed by a vocal interpreter running locally. In fact, only wake-up commands are executed within the device, while others are transmitted remotely to the cloud. Each IVA is activated via its own phrase or keyword and the most popular are "Ok Google", "Alexa", and "Hey Siri", for the case of Google Assistant, Amazon Echo/Alexa and Apple/HomeKit ecosystem, respectively. As it will be detailed later on, a relevant fragility is due to the continuous data exchange from the IVA and the cloud. Even if several frameworks could be considered "secure" both from the architectural and technological viewpoints, still they are prone to a variety of privacy-breaking attacks targeting a composite set of features observable within the encrypted traffic flows [2, 4, 8, 27].

2.1 Threat Model

We aim at investigating the class of attacks targeting the encrypted traffic in a black-box manner, i.e., without trying to decipher the payload of protocol data units. Literature abounds of works dealing with techniques against SSL flows, for instance, [11] provides an extensive survey on Man-in-the-Middle (MitM) attacks for SSL/TLS conversations as well as techniques to highjack or spoof different protocol entities and nodes (e.g., BGP routes, ARP/RARP caches, and access points). Moreover, [21] reports an MitM attack expressly crafted for the Alexa IVA. Specifically, it targets "skills", which are extensions introduced to integrate third-part devices and services in the Amazon ecosystem. An attacker can redirect the voice input of the victim to a malicious node, thus highjacking the conversation. However, such attacks are definitely outside of the scope of this paper. Rather, we consider an adversary wanting to profile the user, for instance, for reconnaissance purposes or to plan a physical attack. To this aim, the adversary can exploit the traffic to infer "behavioral" information, e.g., when the victim is not at home. Figure 3 depicts the reference threat model.

In more detail, we assume an adversary (denoted as *malicious user* in the figure) that can only perform a passive attack, i.e., he/she can observe and acquire the traffic produced by the victim but cannot alter or manipulate it. To this aim, the adversary should access the home router. However, this is not a tight constraint as he/she can abuse the IEEE 802.11 wireless loop to gather information to be sent to the IVA (see, e.g., [9] for an analysis of threats that can be done by moving throughout the attack surface). We also assume that the adversary is not able to use the contents of the packets to launch the attack: in other words, he/she is not able to attack the TLS/SSL or VPN schemes usually deployed. Therefore, by inspecting the

traffic produced by the smart speaker, the adversary can only rely on statistics and metadata of conversations. As an example, the attacker inspects (or computes by performing suitable operations) values like the throughput, the size of protocol data units, the IP address, the number of different endpoints, flags within the headers of the packets, or the behavior of the congestion control of the TCP. Finally, as it's usually done in similar works, we assume that the attacker is able to isolate and recognize traffic that comes from different IoT devices [5, 20, 24].

Even if the deployment of encryption schemes is not sufficient to prevent the leakage of important information about the habits of users [8] and the number or the type of IoT nodes deployed [1, 3, 4], this was a suitable countermeasure to mitigate a wide variety of threats. Alas, the advent of computational-efficient statistical tools brings into a feasibility zone a new wave of attacks. As a prototypal example, the work in [26] demonstrated how to leak the language of the talker by inspecting the bit rate of VoIP conversations. In essence, authors used a sort of "signatures" produced by the variable bit rate codec to feed various classifiers, such as the k-Nearest Neighbors, Hidden Markov Models, and Gaussian Mixture Models and a computational-efficient variant of the χ^2 classifiers, to identify the language with different performances (e.g., they can discriminate between English and Hungarian and from Brazilian Portuguese and English with a 66.5% and 86% of accuracy, respectively). We then review the most suitable tools that the adversary can use to extract information obtained from the gathered traffic and then unhinge the privacy of smart speakers and part of the IoT subsystem.

2.2 Machine Learning Techniques for Attacking the IoT Ecosystem

Nowadays, gathering and analyzing traffic is a core technique used during the reconnaissance phase of an attack [25], e.g., to enumerate devices or to fingerprint hosts for searching known vulnerabilities. In this work we consider the attacker wanting to infer high-level information, for instance to launch social engineering campaigns or plan physical attacks. Literature showcases different machine learning approaches and their adoption to solve networking duties is becoming a *de-facto* standard (see, [23] for a recent survey on the use of deep learning for different traffic classification problems). However, in the perspective of endowing an attacker with the suitable tools to gather information on the state of the smart speaker or the IVA, we shortlisted the following most promising algorithms.

- **Decision Tree** (DT) is a family of non-parametric supervised learning methods suitable for classification and regression problems [16]. The DT builds classification or regression models in the form of a tree structure. To this aim, it breaks down the data into smaller subsets while developing an associated decision tree. The process is iterated by further splitting the dataset and the final result is a tree with decision nodes and leaf nodes.
- **Adaptive Boosting - AdaBoost** (AB) exploits the the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules [16]. AdaBoost can be used in conjunction with many other types of learning algorithms to improve performance. In this case, the output of the other learning algorithms (defined as *weak learners*) is combined into a weighted sum that represents the final output of the boosted classifier.

3 Experimental Testbed

To prove the effectiveness of privacy threats of smart speakers leveraging machine learning techniques, we developed an experimental testbed. Due to the lack of public datasets con-

taining network traffic of smart speakers, we have also developed an automated framework for generating and collecting the relevant network traffic.

Concerning the device under investigation, we used a Google Home Mini[7] since it is one of the most popular smart appliances. Our version is equipped with an IEEE 802.11 L2 interface, an internal microphone to sense commands and the surrounding environment, and a loudspeaker for audio playback and LEDs for visual feedbacks. The configuration of the device must be done via a companion application[8]. To this aim, we provided the SSID and the password of our test network, which allowed the smart speaker to communicate remotely with the cloud running Google services and to exchange data with other devices connect to the same network (e.g., smart tv, smart light bulbs, etc.). We did not performed other tweaks as to reproduce an average installation usually accounting for the device deployed by the user in an out-of-the-box flavor.

Since we are focusing on privacy leakages related to the behavior of the microphone when disabled or when sensing various situations, i.e, the presence of humans or a quiet condition, we performed three different measurement campaigns, each one lasting 3 days. In particular, for the first round of tests, the microphone of the smart speaker was manually set off as to investigate the traffic exchanged between the device and the remote cloud datacenter. Then, for the second round, the microphone was manually set on and the device put in a quiet condition, i.e., the microphone did not receive any stimuli from the surrounding environment, which was completely without noise or voices. For the last round of tests, we set the microphone on and we simulated the presence of humans speaking each others or background noise. We underline that human talkers will not issue the "Ok Google" phrase or will not inadvertently activate the smart speaker. In the following, we denote the different tests as `mic_off` for the case when the microphone is disabled, `mic_on` and `mic_on_noise` for tests with the microphone active and the smart speaker placed in a silent or noisy environment, respectively. To the aim of having proper audio patterns, we selected videos from YouTube in order to stimulate the smart speaker with a wide variety of talkers and settings (e.g., female and male speakers of different ages).

To capture data, we prepared a standard computer to act as the IEEE 802.11 access point and we deployed ad-hoc scripts for running `tshark`[9], i.e., the command line interface provided by the Wireshark tool. To process the dataset and perform computations, we used a computer with an Intel Core i7-3770 processor, with 16 GB of RAM running the Ubuntu 16.04 LTS operating system.

To implement the machine learning algorithms presented in Section 2.2, we used the scikit-learn[10] library. In essence, it is an open-source library developed in Python that contains the implementation of the most popular machine learning algorithms.

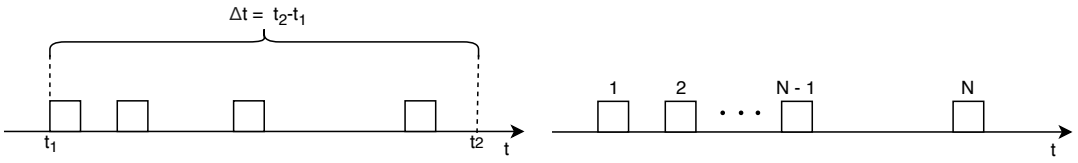
3.1 Data Handling

As said, we only collected traffic without performing any operation aimed at breaking the encryption scheme. In other words, we consider a worst-case scenario where the attacker is not able to perform deep packet inspection or more sophisticated actions (e.g., pinning of SSL certificates). Instead, the threat model we investigate deals with a malicious entity wanting to infer the smart speaker state by only using statistical information observable within the encrypted network traffic. To this aim, the attacker can extract/compute indicators by using

[7] https://store.google.com/it/product/google_home_mini
[8] https://play.google.com/store/apps/details?id=com.google.android.apps.chromecast.app
[9] https://www.wireshark.org/docs/man-pages/tshark.html
[10] http://scikit-learn.org/

two different "grouping" schemes, as depicted in Figure 1. In more detail, we computed the desired metrics by considering a suitable amount of packets obtained according to the windowing mechanisms considered as follows:

- time spans of length Δt (see Figure 1a);
- bursts of a fixed length of N (see Figure 1b).



(a) Packets grouped in a window of Δt seconds (b) Packets grouped in a window of N data units

Figure 1: Different policies for grouping packets used for the computation of statistical information.

We point out that the size of the windows affect the amount of information to be processed by the machine learning algorithm. In fact, even if the dataset sill remains unchanged, the number of windows is directly proportional to the volume of information offered to the statistical tool (i.e., for each window a statistical indicator is computed). Concerning the statistical indicators that an attacker can obtain from the traffic exchanged between the IVA and the cloud, we consider:

- Number of TCP, UDP and ICMP packets: allow to quantify the composition of the traffic in terms of observed protocols. For instance, UDP datagrams indicate the presence of signaling carried by the QUIC protocol, whereas TCP segments can represent the exchange of additional data such as multimedia material.
- Number of different IP addresses and TCP/UDP ports: the presence of different endpoints could be used to spot interaction between the smart speaker and the IVA cloud, including actions requiring to contact third-part entities or providers, IoT nodes, private datacenters or CDN facilities.
- *per*-window Inter packet time (IPT) or packet count: allow to consider how traffic distributes within the two windows used to group packets described in Figure 1. Aggressiveness of the source could be used to reveal user activity or stimuli triggered by a vocal input.
- Average value and standard deviation of the TCP window: describe the behavior of the flow in terms of burstiness and bandwidth usage. Such information could lead to indications about how the IVA and its cloud exchange data.
- Average value and standard deviation of the IPT: similarly to the previous case, they can be used to complete information inferred from the packet rate. For instance, the IPT could be used to recognize whether a flow is generated by an application with some real-time constraints.

- Average value and standard deviation of the packet length: hint at the type of the application layer, for instance, small packets can suggest the presence of voice-based activities requiring a low (bounded) packetization delay.
- Average value and standard deviation of the TTL: can be used to mark flow belonging to different portions of the network and possibly indicating that the smart speaker has been activated for a task also requiring the interaction with additional providers or actuators (e.g., IoT nodes).

We point out that many indicators are intrinsically "privacy leaking" as they allow a malicious observer to infer some information about the smart home hosting the device [4]. For instance, counting different conversations and the number of protocol data units in a timeframe could reveal the presence of specific IoT nodes or the type of the requested operation, e.g., retrieving a summary of the news. At the same time, considering such values could impact on the performance of the classification framework owing to the exploitation of interactions among the different architectural components, which are difficult to forecast.

4 Preliminar Results

In this section, we showcase numerical results obtained in our trials. First, we provide an overview of the collected dataset, then we present the performances of machine learning algorithms used to leak privacy of users with particular attention on the time needed for the training phase.

4.1 Dataset Overview

As presented in Section 3, the dataset has been generated in a 9 day long measurement campaign composed of three trials of 3 days with different conditions of the microphone of the smart speaker. Specifically, for the `mic_off` case, we collected $203,596$ packets for a total size of 69 Mbytes. Instead, when the microphone is active, we collected $216,456$ packets in the `mic_on` scenario and $282,656$ packets `mic_on_noise` one, for a total size of 74 and 173 Mbytes, respectively. The overall dataset has been processed with the StandardScaler, thus leading to a statistical population with average equal to 0 and standard deviation equal to 1.

Figure 4 depicts the average values characterizing the dataset in each scenario. It is worth noting that the average packet length and the average size of the TCP window for the `mic_off` and `mic_on` cases are very similar. Instead, for the `mic_on_noise` case, the average packet length doubles, whereas the average TCP window size halves.

4.2 Classifying the State of the Smart Speaker

We now show the results obtained when trying to classify the state of the smart speaker to conduct a privacy leaking attack.

The first experiment aimed at investigating whether it is possible to identify if the microphone of the smart speaker or the device hosting the IVA is turned on or off. We point out that this can be also viewed as a sort of side-channel, where the attacker can identify if users are in the proximity of the device. In this perspective, Figure 5 shows the accuracy of the classifiers adopted to infer from the traffic whether the microphone is ON or OFF, i.e., discriminate among `mic_on` or `mic_off` cases. To better understand the performances, we also investigated

when the different "grouping" strategies presented in Section 2.2 are used to feed the machine learning algorithms.

As shown, best results are achieved by using the AdaBoost algorithm (denoted as AB in the figure). However, it is important to note that, for identifying the state of the microphone with an acceptable level of accuracy, the attacker has to collect about 500 s of traffic or 500 packets. Therefore, a real-time classification could be not possible in the sense that the attacker has to wait a non-negligible amount of time before he/she has the knowledge to launch the attack (e.g., force the physical perimeter where the smart speaker is deployed).

The second experiment aimed at discriminating between the two different behaviors of the surrounding environment, i.e., the `mic_on` and `mic_on_noise` states. We recall that such states can be used by the attacker to infer if the smart speaker operates in a silent environment or in the presence of noise, e.g., people are talking to each other or the television is turned on. In both cases, there is not a direct interaction, that is, in the case of Google Home, any user did not issue the "Ok Google" phrase. Then, the malicious user cannot exploit "macro" features of the traffic, such as the number of TCP connections, the IP range or the traffic volume [4, 22].

Figure 6 depicts the obtained results. Compared to the previous experiment, to reach a good level of accuracy, it is sufficient to use a reduced amount of packets. As an example, for the case of the Decision Tree, good degrees of accuracy to decide whether the smart speaker is in the `mic_on` or `mic_on_noise` states are achieved by using time-windows with $\Delta t = 15$ seconds or a burst of $N = 20$ packets. From the perspective of understanding the security and privacy of voice-based appliances, this result reveals a potential exploitable hazard. In fact, when the user does not directly interact with the smart speaker (e.g., the "Ok Google" phrase is not issued), the traffic generated towards the remote cloud should be the same for both the `mic_on` and `mic_on_noise` conditions. In other words, it is expected that the network traffic does not exhibit any signature. Even if we did not have access to the internals of the Google Home Mini used in our testbed, the different traffic behaviors could be due to the fact that the smart speaker is always in an "awake" mode and selected stimuli are sent to the cloud as to identify activation phrases like "Ok Google" or "Hey Siri". However, this could partially contradict the believing that such phrases are completely handled locally by the smart speaker or the IVA.

To assess the performances of the different classifiers in a comprehensive manner, Figure 7 shows the confusion matrices of the AdaBoost and Decision Tree classifiers when used to discriminate between the `mic_on` - `mic_on_noise` cases. It is possible to notice how the confusion matrices show the goodness of the chosen algorithms having the highest values distributed on the diagonal. Similar considerations can be done for the other techniques but they have been omitted here for the sake of brevity.

5 Conclusions and Future Works

In this paper, we investigated the feasibility of adopting machine learning techniques to breach the privacy of users interacting with smart speakers or voice assistants. Different from other works discovering the presence of the user via intrinsically privacy-leaking activities (e.g., the activation of a IoT node and the related traffic flow), we concentrated on discriminating how the internal microphone is used. Results indicate the effectiveness of our approach, thus making the management of silence and noise *époque* as major privacy concerns. To increase the user's privacy a possible countermeasure could be the insertion of suitable padding inside the packets to normalize the average length, as well as the standard deviation moreover, using a unique protocol for the transport, could add another layer of privacy. Another possible countermeasure

could be an of appropriate "noise", for instance by exploiting some form of traffic camouflage or morphing [14]. Therefore, suitable traffic morphing or protocol manipulation techniques should be put in place within the device or, at least, in-home routers as to reduce the attack surface that can be exploited by malicious entities.

Future work will aim at refining our framework by considering smart speakers from other vendors. Besides, we are working towards the implementation of a sort of "warden" able to normalize traffic generated towards the IVA cloud.

References

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and A. Selcuk Uluagac. Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted! 2018.

[2] Efthimios Alepis and Constantinos Patsakis. Monkey Says, Monkey Does: Security and Privacy on Voice Assistants. *IEEE Access*, 5:17841–17851, 2017.

[3] Yousef Amar, Hamed Haddadi, Richard Mortier, Anthony Brown, James Colley, and Andy Crabtree. An Analysis of Home IoT Network Traffic and Behaviour. 2018.

[4] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic. 2017.

[5] Lei Bai, Lina Yao, Salil S Kanhere, Xianzhi Wang, and Zheng Yang. Automatic Device Classification From Network Traffic Streams of Internet of Things. In *43rd Conference on Local Computer Networks*, pages 1–9. IEEE, 2018.

[6] Prasenjeet Biswal and Omprakash Gnawali. Does QUIC Make the Web Faster? In *2016 IEEE Global Communications Conference*, pages 1–6. IEEE, 2016.

[7] Andrea Cardaci, Luca Caviglione, Alberto Gotta, and Nicola Tonellotto. Performance Evaluation of SPDY Over High Latency Satellite Channels. In *International Conference on Personal Satellite Services*, pages 123–134. Springer, 2013.

[8] Luca Caviglione. A First Look at Traffic Patterns of Siri. *Transactions on Emerging Telecommunications Technologies*, 26(April):664–669, 2015.

[9] Luca Caviglione, Mauro Coccoli, and Alessio Merlo. A Taxonomy-based Model of Security and Privacy in Online Social Networks. *International Journal of Computer Sciences and Engineering*, 9(4):325–338, 2014.

[10] Luca Caviglione, Maciej Podolski, Wojciech Mazurczyk, and Massimo Ianigro. Covert Channels in Personal Cloud Storage Services: The case of Dropbox. *IEEE Transactions on Industrial Informatics*, 13(4):1921–1931, 2016.

[11] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. A Survey of Man in the Middle Attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051, 2016.

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

[13] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. Your Voice Assistant is Mine: How to Abuse Speakers to Steal Information and Control Your Phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, pages 63–74. ACM, 2014.

[14] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-Bboo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *IEEE Symposium on Security and Privacy*, pages 332–346. IEEE, 2012.

[15] Marcia Ford and William Palmer. Alexa, Are You Listening to Me? An Analysis of Alexa Voice Service Network Traffic. *Personal and Ubiquitous Computing*, 23(1):67–79, 2019.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2009.

[17] B. Kinsella. Smart Speaker Prevision. https://voicebot.ai/2019/04/15/smart-speaker-installed-base-to-surpass-200-million-in-2019-grow-to-500-million-in-2023-canalys. Last Accessed: Sept. 2019.

[18] Shancang Li, Shancang Li, Kim-Kwang Raymond Choo, Qindong Sun, William J. Buchanan, and Jiuxin Cao. IoT Forensics: Amazon Echo as a Use Case. *IEEE Internet of Things Journal*, 14(8):1–1, 2015.

[19] W. Mazurczyk and L. Caviglione. Information Hiding as a Challenge for Malware Detection. *IEEE Security Privacy*, 13(2):89–93, 2015.

[20] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In *Proceedings of the Symposium on Applied Computing*, pages 506–509. ACM, 2017.

[21] Richard Mitev, Markus Miettinen, and Ahmad-Reza Sadeghi. Alexa Lied to Me: Skill-based Man-in-the-Middle Attacks on Virtual Assistants. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 465–478. ACM, 2019.

[22] Antônio J Pinheiro, Jeandro de M Bezerra, Caio AP Burgardt, and Divanilson R Campelo. Identifying IoT Devices and Events Based on Packet Length from Encrypted Traffic. *Computer Communications*, 144:8–17, 2019.

[23] S. Rezaei and X. Liu. Deep Learning for Encrypted Traffic Classification: An Overview. *IEEE Communications Magazine*, 57(5):76–81, May 2019.

[24] Mustafizur R Shahid, Gregory Blanc, Zonghua Zhang, and Hervé Debar. IoT Devices Recognition Through Network Traffic Analysis. In *EEE International Conference on Big Data*, pages 5187–5192. IEEE, 2018.

[25] Siraj A Shaikh, Howard Chivers, Philip Nobles, John A Clark, and Hao Chen. Network Reconnaissance. *Network Security*, 2008(11):12–16, 2008.

[26] Charles V Wright, Lucas Ballard, Fabian Monrose, and Gerald M Masson. Language Identification of Encrypted VoIP Traffic: Alejandra y Roberto or Alice and Bob? In *USENIX Security Symposium*, volume 3, pages 43–54, 2007.

[27] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal*, 4(5):1250–1258, 2017.

A Appendix

A.1 General smart home scenario and threat model

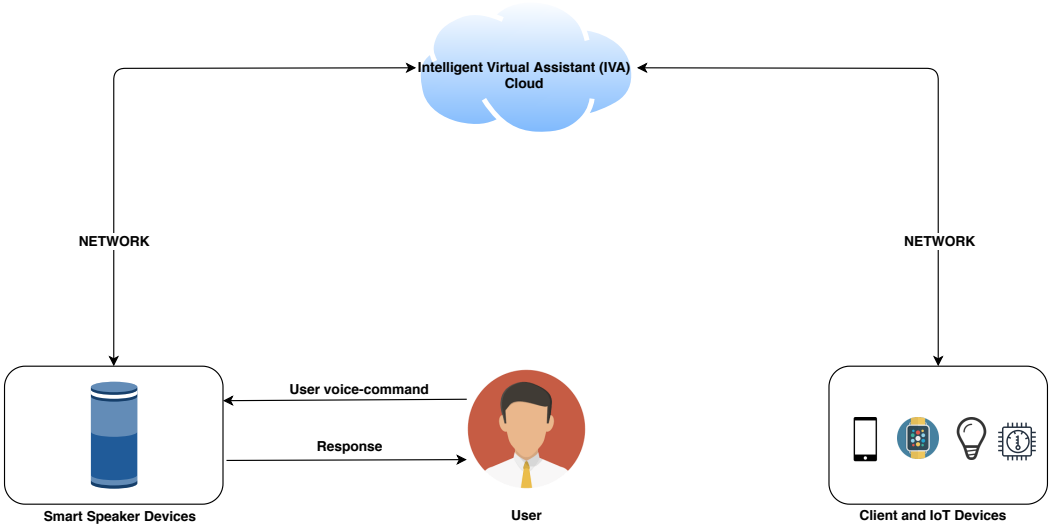


Figure 2: General system architecture used by smart speakers to control nodes in smart home scenarios.

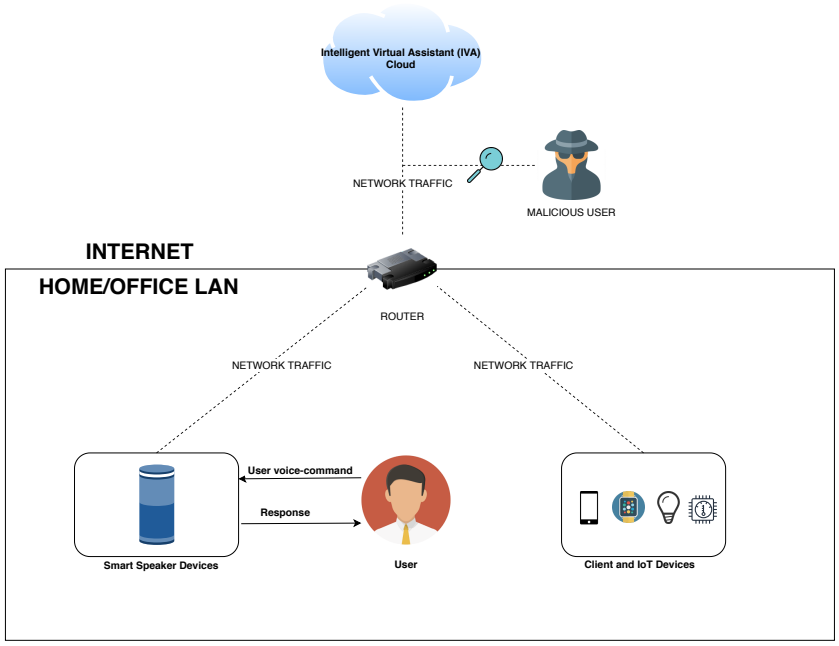
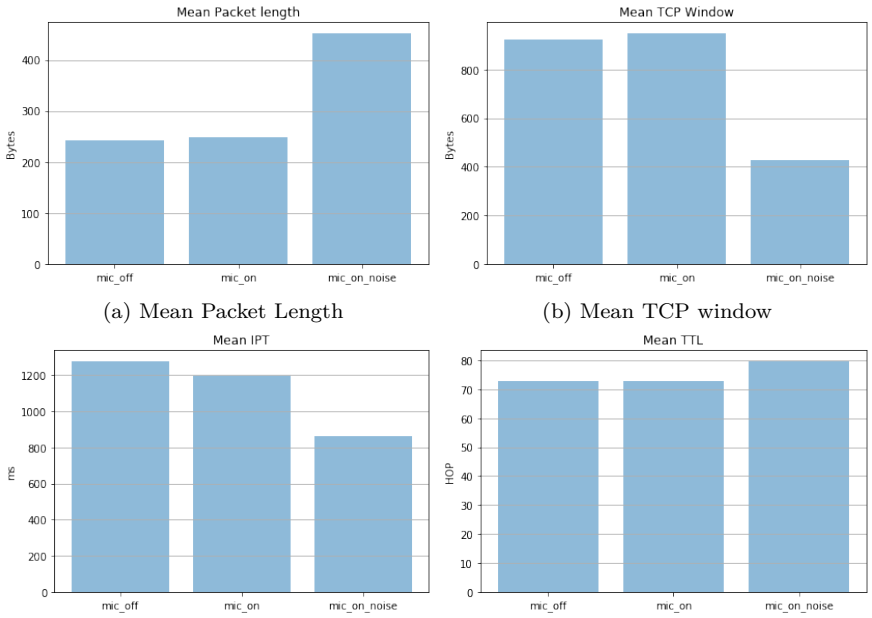


Figure 3: Reference threat model targeting the encrypted traffic for privacy-breaching attacks.

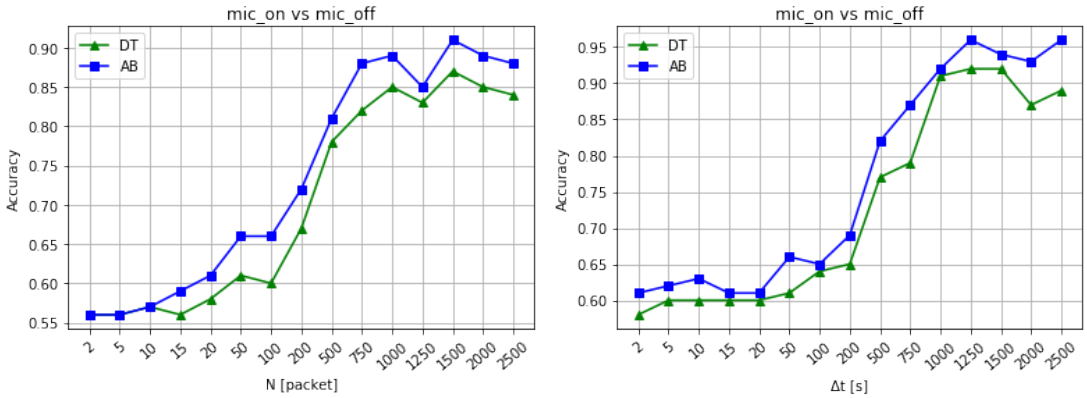
A.2 Dataset Overview and Classifiers Results



(a) Mean Packet Length (b) Mean TCP window

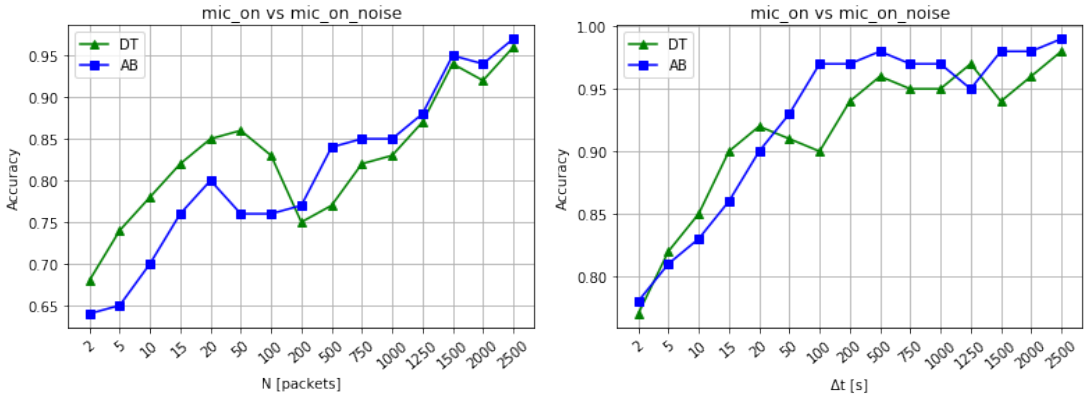
(c) Mean IPT (d) Mean TTL

Figure 4: Average values for the Packet Length, TCP Window, IPT and TTL computed over the entire dataset.



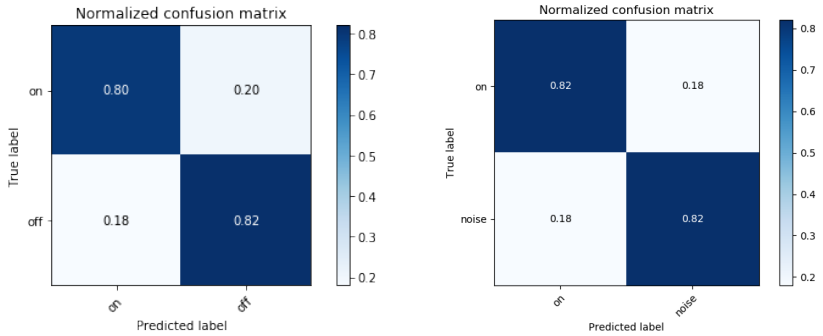
(a) Grouping in a window of N packets (b) Grouping in a window of Δt seconds

Figure 5: Accuracy of the classifiers for the `mic_off` and `mic_on` cases.



(a) Grouping in a window of N packets (b) Grouping in window of Δt seconds

Figure 6: Accuracy of the classifiers for the `mic_on` and `mic_on_noise` cases.



(a) AdaBoost: $\Delta t = 500$ - `mic_off` vs `mic_on` (b) Decision Tree: $N = 15$ - `mic_on` vs `mic_on_noise`

Figure 7: Confusion matrix showing the best results obtained in different use-cases.