Fine-hearing Google Home: why silence will not protect your privacy

Davide Caputo[1*], Luca Verderame[1], Andrea Ranieri[2], Alessio Merlo[1], and Luca Caviglione[2]
[1]*Computer Security Lab, Department of Informatics, Bioengineering, Robotics and Systems Engineering*
University of Genova, Genova, Italy
{davide.caputo, luca.verderame, alessio}@dibris.unige.it
[2]*Institute for Applied Mathematics and Information Technologies*
National Research Council of Italy, Rome, Italy
{andrea.ranieri, luca.caviglione}@ge.imati.cnr.it

Received: January 17, 2020; Accepted: March 6, 2020; Published: March 31, 2020

Abstract

Smart speakers and voice-based virtual assistants 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 activate an IoT device. Unfortunately, even if privacy and security are enforced through state-of-the-art encryption mechanisms, the features of the encrypted traffic, such as the throughput, the size of protocol data units or the IP addresses can leak critical information about the habits of the users. In this perspective, in this paper we showcase this kind of risks by exploiting machine learning techniques to develop black-box models to classify traffic and implement privacy leaking attacks automatically. 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. We also present a set of experimental results collected in a realistic scenario, and propose possible countermeasures.

Keywords: smart speakers, IoT security, machine learning and traffic analysis

1 Introduction

Modern smart homes are composed of several devices connected and organized in complex ecosystem. Among them, smart speakers and voice-based virtual assistants are becoming an essential building block of modern smart homes environments. For instance, they can be used to retrieve information, interact with other devices, and command a wide range of Internet of Things (IoT) nodes. Moreover, smart speakers 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 [23, 12], there are over 200 million of smart speakers installed in private properties, 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 speaker or the virtual assistant receives a textual representation as well as optional, companion

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 11(1):35-53, Mar. 2020
DOI:*10.22667/JOWUA.2020.03.31.035*

*Corresponding author: Computer Security Lab, Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genova, Via Dodecaneso 35, Genova, Genova, 16146, Italy, Tel: +39-010-353-2344

multimedia data. Then, it executes the command or routes it to a proper hub using a variety of wireless technologies (e.g., ZigBee, Bluetooth or Wi-Fi) to communicate with IoT nodes.

From a privacy and security standpoint, the prime mechanism to protect the transmitted data is the encryption of traffic (see, e.g., reference [41] and references therein). However, features of the flows such as the throughput, the size of protocol data units or `(address, port)` tuples, can leak valuable information about the habits of the users [15] or the number and the type of IoT nodes [6, 34]. 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. By exploiting this knowledge, a malicious user can launch a vast collection of attacks, such as, users profiling, physical attacks or social engineering attacks [6, 34].

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 [6], authors demonstrate how to infer some user activities monitoring the network traffic, e.g., traffic flows produced by switches and health monitors can leak the sleep cycle of a user. In [18], the traffic produced by state transitions of home devices (i.e., a thermostat and a carbon dioxide detector) can be used as an indicator to understand if a user is present in the home. Following the same approach, the authors of [1] use passive measurements 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 [34] several machine learning techniques are used to identify IoT devices by exploiting "poor" information like the length of packets produced during normal operations.

Besides, in [4] the risks of HTTP-based communications are discussed, both from the perspective of inferring data on the devices (e.g., the state or intensity of a light source) and from the perspective of performing session-hijacking attacks. Furthermore, [24] highlights the importance of the sensitive data stored in IoT nodes and smart speakers for forensics investigations, as traffic patterns, may enable the exfiltration of data by malware owing to information hiding schemes [19] or hidden channels.

In this vein, this paper investigates the use of machine learning techniques to develop black-box models for the automatic classification of network traffic and the implementation of privacy leaking attacks in smarthome IoT deployments. Unlike previous works [1, 4, 18, 34], we focus on understanding whether the presence of the user can be recognised even if no queries to the device are made. In fact, we argue than once a request is generated, the produced traffic volumes or the appearance of specific network addresses may 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. Then, we discuss some possible remediation to mitigate such unexpected identification by acting at the traffic level. In fact, designing appropriate mitigation techniques is often overlooked (see, e.g., [4] for a notable exception) or addressed at an API-permission level [2], which is definitely out of the scope of the paper.

The remainder of the paper is structured as follows. Section 2 presents the reference architecture used by smart speakers to control IoT devices, the threat model and the machine learning mechanism exploitable by an attacker. Section 3 deals with the testbed used to collect data, while Section 4 showcases the experimental results. Section 5 proposes some countermeasures and Section 6 concludes the paper and hints some possible future directions.

2 Architecture, Threat Model and IA Tools

Smart speakers provide a user interface to issue requests or commands using natural speech. They can be also used as hubs for other IoT nodes and network appliances or to perform tasks like playing music and

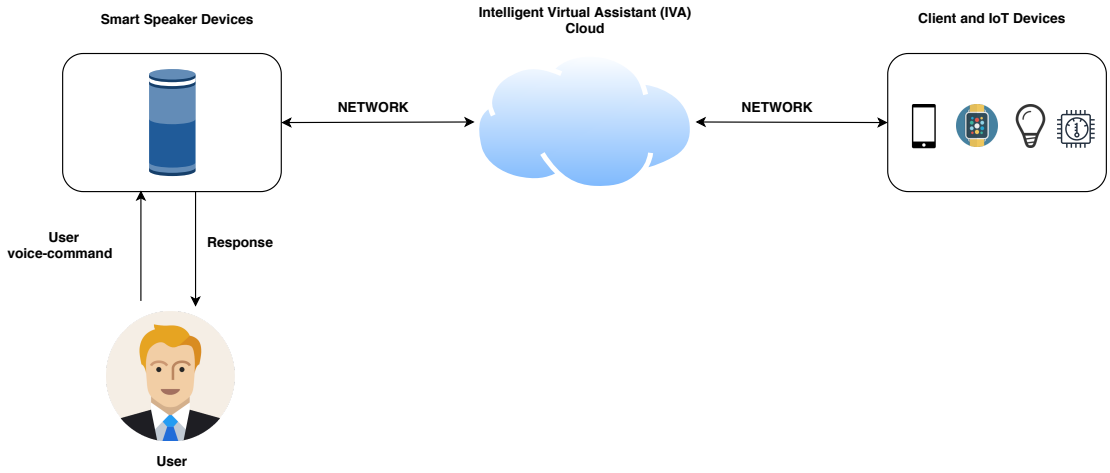


Figure 1: Reference architecture leveraging smart speakers to control nodes in smart home scenarios.

video, buying items, and making recommendations. In addition, smart speakers and virtual assistants can provide a variety of information including news, weather forecasts and traffic information.

For the time being, Google Home[1], HomePod[2] and Amazon Echo[3] are the most popular smart speakers incorporating the above features, while Amazon Alexa[4], Apple Siri[5] and Google Assistant[6] are the main virtual assistants. For this class of devices and services, literature still lacks a unified terminology, referring to smart speakers and virtual assistants with terms like "smart personal assistants", "virtual personal assistants" or"home digital voice assistants", to mention a few. Therefore, in the following, we use the terms smart speakers or Intelligent Virtual Assistant (IVA) interchangeably, except when doubt may arise.

Even though specific design choices and configurations characterize each smart speaker model, the core architectural blueprint is quite standard and it is composed of four major component, as shown in Figure 1:

- **Smart Speaker** or **IVA**: it is responsible for gathering vocal commands, sampling them and transmitting the data to a backend via the Internet. Upon receiving a response, the smart speaker or software 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). 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. In some cases, the Smart Speaker 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) [21].

- **Client** and **IoT Devices**: they are the targets of commands of the ecosystem. Sensors, actuators, Bluetooth/ZigBee bridges or IoT-capable devices are typical examples of IoT nodes installed in a smart home. As previously said, some entities belonging to this class can be co-located within the smart speaker.

[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/`

- **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 composite set 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 [15, 21]. Indeed, in a typical smart home scenario, the vocal stimuli are processed in the vendor datacenter and sent back to the IVA. At the same time, the multimedia content is transmitted through a third-party CDN, plus some of the IoT nodes composing the smart home environment could establish a direct point-to-point connection with the computing infrastructure of the manufacturer.

- **Network**: it provides connectivity. Typical deployments use a single local (wireless) network connected via a router/gateway to the Internet. In most complex scenarios, different links could be present, e.g., local access cabled network for some IoT nodes and hubs and multiple wireless loops to connect smart devices [14]. Concerning protocols used for data exchange between the IVA and the cloud, the TCP is the primary choice, with the multipath variant optimizing performance and reducing delays [15]. A notable exception is the Google ecosystem. In details, Google Home exploits QUIC [9], a protocol originally engineered to improve performance issues of HTTP/2 and based on transport streams multiplexed over UDP. 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 behaviours, which can be highly influenced by the underlying network conditions (see, e.g., [13] for a sensitivity/performance analysis of the SPDY counterpart in different wireless settings).

The overall set of components of the architectural blueprint 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.

2.1 Threat Model

In a typical scenario, smart speakers rely on a microphone to sense commands, which are processed by a local vocal interpreter. Only wake-up commands are executed within the device, while others are transmitted remotely to the cloud for the elaboration.

From a security standpoint, the continuous data exchange between the IVA and the cloud represents a desirable target for attackers. In details, attacks such as Man-in-the-Middle (MitM) [5, 31, 27], attacks on RPL (Routing Protocol for low power Lossy network) and 6LowPAN [35] or spoofing attacks [37, 42] aim to intercept and exfiltrate sensitive data, even if the communication is encrypted using TLS/SSL. For instance, [17] provides an extensive survey on 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). Besides, [30] reports a MitM attack expressly crafted for the Alexa IVA. By using "skills", which are extension added to incorporate third-party devices and services into the Amazon ecosystem, the authors were able to redirect the victim's voice input to a malicious node, thus hijacking the conversation. Still, even if commercial IVAs implement state-of-the-art security mechanisms to protect the network traffic, they are prone to a variety of privacy-breaking attacks targeting a composite set of features observable within the encrypted traffic flows [2, 6, 41].

To this aim, our work focuses on 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. In particular, we are interested in threat scenarios where the adversary can exploit the encrypted traffic to infer "behavioural" information, e.g., when the victim is not at home. Figure 2 depicts the reference threat model.

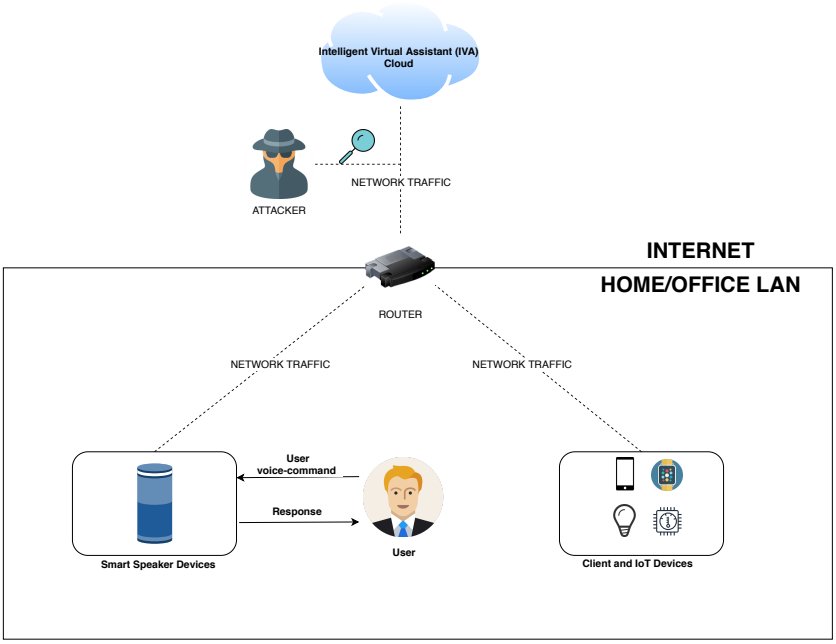


Figure 2: Reference threat model for privacy-breaching attacks targeting the encrypted traffic.

In more detail, we assume an adversary (denoted as the *malicious user* in the figure) that can only 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., [16] 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 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, we can assume that the attacker can isolate and recognize traffic that comes from different IoT devices, following the same approach of [8, 26, 28, 38].

2.2 Machine Learning Techniques for Attacking the IoT Ecosystem

Collecting and analyzing traffic is nowadays a core technique used during the reconnaissance phase of an attack [39, 29], for instance, to enumerate devices or fingerprint hosts to search for known vulnerabilities. In this work, we consider the attacker wanting to infer high-level information, e.g., to launch social engineering campaigns or plan physical attacks. The problem of data classification has been extensively discussed in the literature and machine learning is considered the best technique [33] for this kind of tasks, thus becoming a *de-facto* standard in the field. Those techniques are also used to resolve the classification problem of encrypted network traffic, as discussed in [3, 36, 32, 43].

In our work, the goal of the attacker is to gather information on the state of the smart speaker or the IVA by only monitoring encrypted network traffic, which reduces to a classification problem applied to network flows. In the following, we shortlisted the most promising and used algorithms in the literature.

k-Nearest Neighbors (kNN) is a supervised learning algorithm suitable for classification and regression problems [22]. The algorithm assumes that similar things exist in close proximity. Accordingly, kNN finds the distances between a query and all the data with a known label and select the entries (i.e., k) closest to the query. Then it votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Decision Tree (DT) is a family of non-parametric supervised learning methods for classification and regression problems [22]. The DT builds classification or regression models in the form of a tree structure by breaking 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 inaccurate rules [22]. 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.

Random Forest is a model composed of many decision trees [11, 22] and relies upon the following key steps: random sampling of training data points when building trees, and definition of random subsets of feature generated when splitting nodes. When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with a replacement technique, known as bootstrapping [22], which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, the entire forest will have lower variance without increasing the bias. We point out that, when using the random forest, only a subset of all the features are considered for splitting each node in the decision tree.

Support Vector Machine (SVM) is a group of supervised learning models for solving both regression or classification problems [22, 25, 40]. SVM builds non-probabilistic binary classifiers whose purpose is the search for the optimal hyper-plan of separation between the two possible classes within the feature space. The SVM is used more when the input data is not directly separable as it allows to map the initial data into a higher dimension space where it is possible to find a separation hyperplane.

Neural Networks are a family of algorithms, loosely modelled after the human brain, that are designed to recognize patterns [10, 22]. Neural networks are organized in a series of layers, where the input vector enters at the left side of the network (the input layer), which is then projected to one or more hidden layers. Each input for each hidden layer, is a weighted sum of the values produced by the preceding layer. This vector (or tensor) is called summed activation of the node. Each summed activation is transformed via an activation function and then becomes the input for the next layer. Nonlinear activations, which allow the network to model even very complex systems, are typically used after the input layer and after all the hidden layers except the last one. In fact, before the output layer, a linear activation function is used in case of regression or a softmax function for classification.

K-fold cross-validation avoids the risk of missing important patterns or trends in the dataset. To this aim, the data is randomly partitioned into k equal-sized sub-samples. Each single sub-sample is retained as the validation data for testing the model, and the remaining $k-1$ sub-samples are used as training data. The process is then repeated k times, with each of the k sub-samples used exactly once as the validation data. The k results from the folds are averaged to produce a single estimation. An interesting property of

this method allows to limit overfitting phenomena.

3 Experimental Evaluation

To prove the effectiveness of privacy threats of smart speakers, we developed an experimental testbed leveraging the machine learning techniques presented in Section 2.2. Due to the lack of public datasets containing network traffic of smart speakers, we have also developed an automated framework for generating and collecting the relevant network traffic.

3.1 Testbed

For our analysis, we tested a Google Home Mini[7] smart speaker, equipped with an IEEE 802.11 L2 interface, an internal microphone for detecting commands and the surrounding environment, and a loudspeaker for audio playback and LEDs for visual feedback.

To connect to the network, a companion mobile application[8] must be used. To this aim, we provided the SSID and the password of our ad-hoc 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 perform other tweaks as to reproduce an average installation usually accounting for the device deployed by the user in an out-of-the-box flavour.

To capture data, we prepared a standard computer to act as the IEEE 802.11 access point of the network 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. Scikit-learn is an open-source library developed in Python that contains the implementation of the most popular machine learning algorithms. However, after some preliminary trials, we shortlisted the considered techniques. Specifically, in the perspective of investigating the feasibility of performing an attack via off-the-shelf methods to unhinge the privacy of encrypted traffic generated by IVAs, we discarded SVMs. Finally, we implemented a neural network with only two hidden layers using the popular Fast.ai library. Fast.ai is a Python library built on top of Pytorch that allows to create and train the most popular neural network models in a simpler way compared to Keras/Tensorflow.

3.2 Experimental Definition

Since we are interested in discovering "behavioural" knowledge from the encrypted traffic, we aimed to detect if we can infer the presence of a human inside a building by the traffic generated by the smart speaker. In more detail, we focused our attention on the behaviour of the microphone, thereby trying to distinguish whether it is disabled or if it is sensing various situations, e.g., the execution of a query or a quiet condition.

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. 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

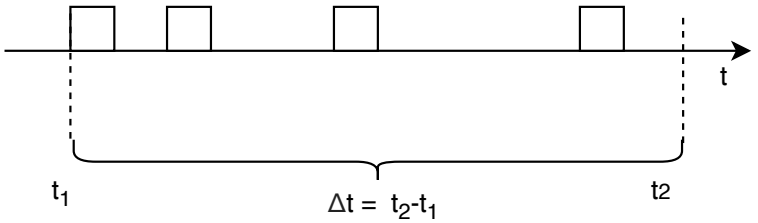
[7] `https://store.google.com/it/product/google_home_mini`
[8] `https://play.google.com/store/apps/details?id=com.google.android.apps.chromecast.app`
[9] `https://www.wireshark.org/docs/man-pages/tshark.html`
[10] `http://scikit-learn.org/`

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 (i.e., female and male speakers of different ages).

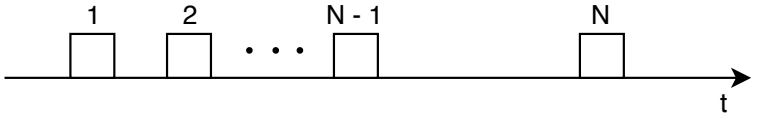
3.3 Data Handling

As said in the previous sections, we only collected traffic in a passive way thus we do not carry out any operation intended to break 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). In this case, the considered threat model 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 compute indicators by using two different "grouping" schemes, as depicted in Figure 3. Specifically, 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 3a);
- bursts of a fixed length of N (see Figure 3b).



(a) Packets grouped in a window of Δt seconds



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

Figure 3: Policies used for grouping packets to compute statistical information.

We point out that the size of the windows affects the amount of information to be processed by the machine learning algorithm. In fact, even if the dataset sill remain 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.** They allow to quantify the composition of the traffic in terms of observed protocols. UDP datagrams, for example, indicate the presence of voice information by the QUIC protocol, whereas TCP segments may represent the exchange of additional data such as multimedia.

- **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-party entities or providers, IoT nodes, private datacenters or CDN facilities.
- ***Per*-window Inter packet time (IPT) or packet count**. It determines how traffic is distributed within the two windows used to group packets described in Figure 3. The 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**. Those values describe the behaviour of the flow in term 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. 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.** They hint 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.** It 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 [6]. 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.

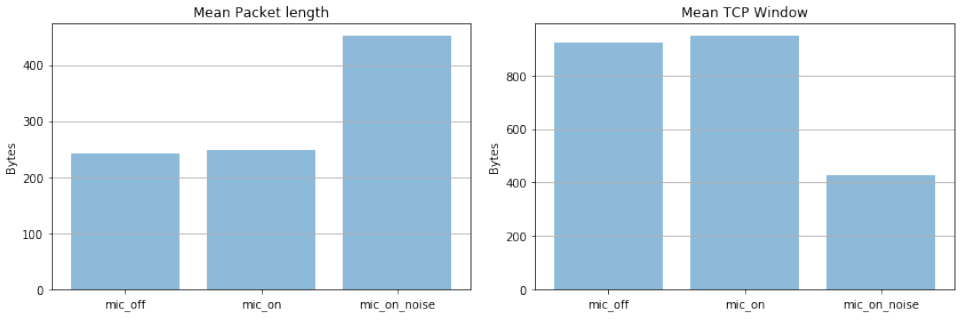
4 Experimental Results

In this section, we showcase the numerical results obtained in our trials. First, we show an overview of the collected data set, then we present the performance of machine learning algorithms used to leak user privacy with particular attention to the time required for the training phase.

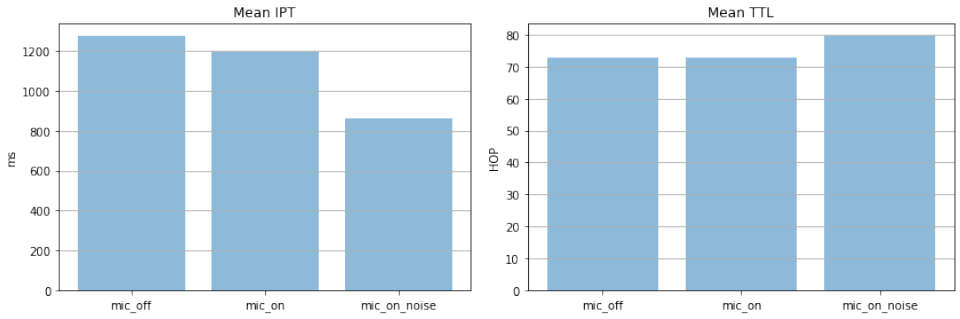
4.1 Dataset Overview

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 shows the average values in each scenario that characterize the dataset. It is worth noticing that the average packet length and the average size of the TCP window for the `mic_off` and `mic_on`

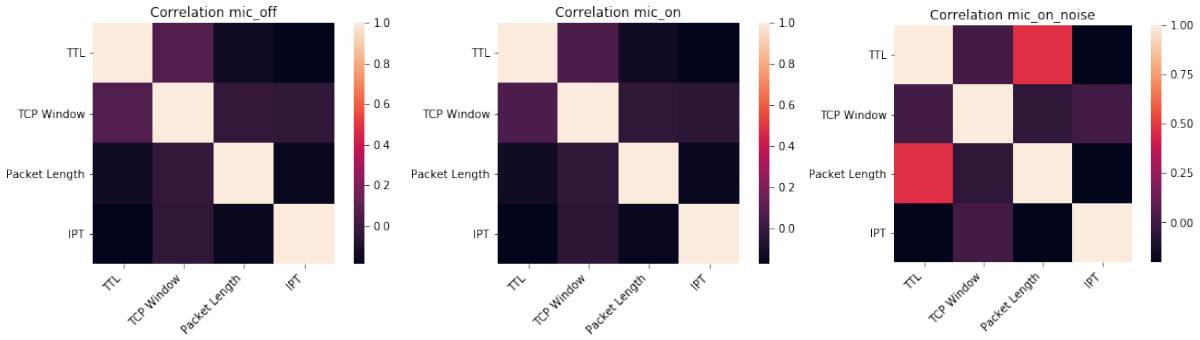


(a) Mean Packet Length (b) Mean TCP window



(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) Correlation of `mic_off` (b) Correlation of `mic_on` (c) Correlation of `mic_on_noise`

Figure 5: Correlation analysis on all the performed measurements for the TTL, IPT, TCP Window, and packet lenght.

cases are very similar. Instead, for the `mic_on_noise` case, the average packet length doubles, whereas the average TCP window size halves.

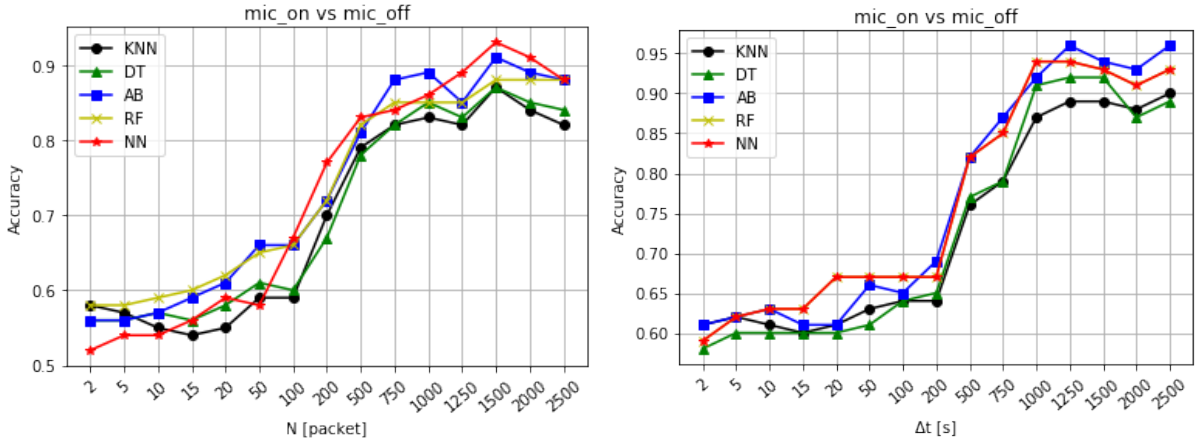
Figure 5 depicts the correlation matrices computed for each test composing the dataset. In general, there are not strong relationships among the data. The only exception is when considering the packet length and the TTL when the microphone is sensing noise. In fact, as shown in Figure 5c, a positive correlation is present.

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 also be 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 6 shows the accuracy of the various classifiers adopted to infer from the traffic whether the microphone is ON or OFF, i.e., discriminate between `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.



(a) Grouping in a window of N packets

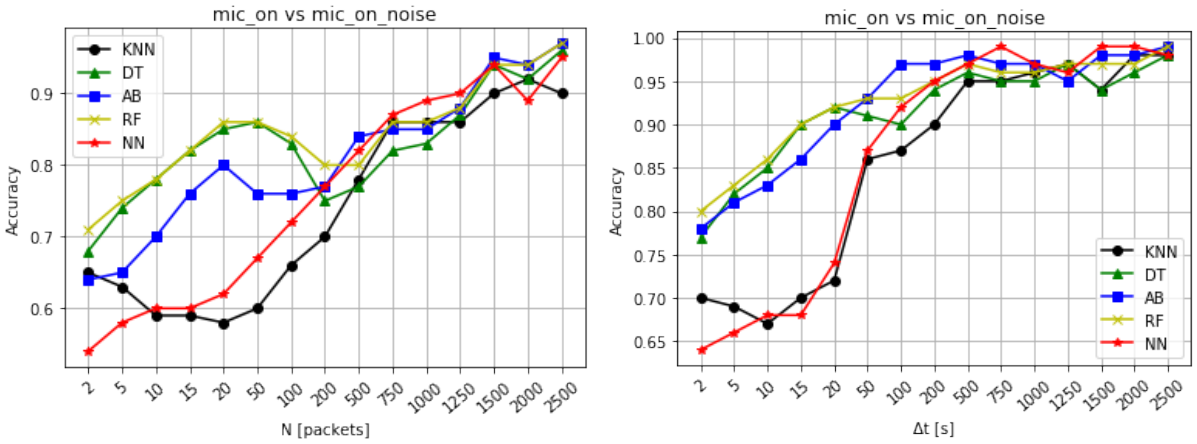
(b) Grouping in a window of Δt seconds

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

The best results are achieved by using the AdaBoost algorithm (denoted as AB in the figure). 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.

The second experiment aimed at discriminating between the two different behaviours of the surrounding environment, i.e., the `mic_on` and `mic_on_noise` states. The attacker can use such states to determine if the smart speaker works in a silent environment or in the presence of noise, for example, when people talk to each other or if the TV 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 entity cannot exploit "macro" features of the traffic, such as the number of TCP connections, the IP range or the traffic volume [6, 34].

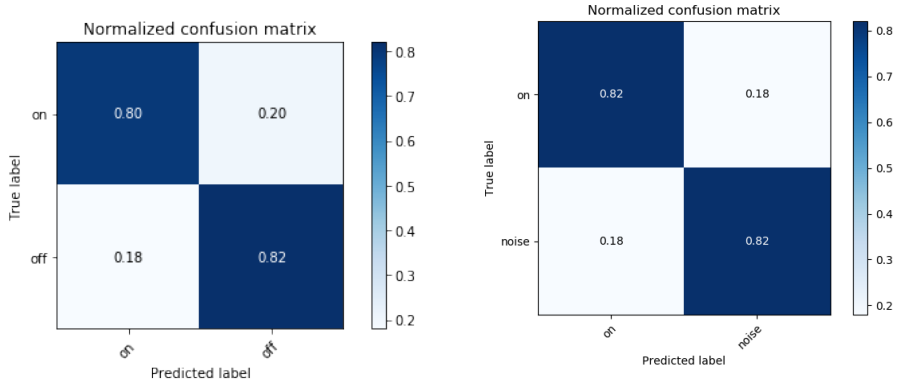
Figure 7 shows the obtained results. Compared to the previous experiment, to reach a good level of accuracy, it is sufficient to use a reduced amount of packets. For example, in the case of the DT, good degrees of accuracy are achieved by using time-windows with $\Delta t = 15$ seconds or a burst of $N = 20$ packets to determine if the smart speaker is in the `mic_on` or `mic_on_noise` states. From the perspective of understanding the security and privacy of voice-based appliances, this result reveals potential exploitable hazards. Indeed, 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



(a) Grouping in a window of N packets

(b) Grouping in window of Δt seconds

Figure 7: 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 8: Confusion matrix showing the best results obtained in different use-cases.

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 behaviours 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 entirely handled locally by the smart speaker or the IVA.

To assess the performances of the different classifiers in a comprehensive manner, Figure 8 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.

4.3 Analysis of the Training Time

The ability to perform a real-time classification is a critical factor for successfully launch an attack targeting the physical space, and the time needed to train from scratch a statistical tool is of paramount importance. Table 1 and Table 2 report the training times for each classifier when the different "grouping"

Table 1: Training times for the used classifiers with different windows of length Δt.

	Training Time [s]									
	`mic_on - mic_off`					`mic_on - mic_on_noise`				
Δt [s]	kNN	DT	AB	RF	NN	kNN	DT	AB	RF	NN
2	1.52	0.47	11.04	44.01	246.0	1.55	0.41	13.28	41.16	212.0
5	0.71	0.29	8.35	29.01	197.0	0.58	0.28	9.90	26.20	195.0
15	0.07	0.15	4.47	15.21	151.0	0.05	0.13	5.26	11.95	181.0
50	0.01	0.05	1.68	5.29	165.0	0.01	0.04	1.96	4.56	165.0
200	0.01	0.01	0.51	1.45	151.0	0.01	0.01	0.61	1.09	151.0
500	0.01	0.01	0.27	0.55	150.0	0.01	0.01	0.31	0.46	150.0
1000	0.01	0.01	0.19	0.29	150.0	0.01	0.01	0.21	0.30	150.0
1500	0.01	0.01	0.16	0.21	150.0	0.01	0.01	0.19	0.24	150.0
2000	0.01	0.01	0.15	0.19	150.0	0.01	0.01	0.17	0.21	150.0
2500	0.01	0.01	0.14	0.18	150.0	0.01	0.01	0.16	0.19	140.0

strategies described in Section 3.3 are used.

As shown in the tables, by increasing the window (i.e., the value for Δt and N) the training time decreases. This behaviour is justified by the fact that the classifiers are trained by using statistical information about the traffic provided by every single window. In other words, the larger the window, the more packets are contained, meaning that less information is provided to the machine learning algorithm. Such behaviour can also be spotted by comparing the tables. In fact, despite the used window scheme, when the resulting amount of information is comparable, the algorithms behave in the same manner. We point out that, high values for Δt have been reported only for the sake of comparison, as they can be hardly used in real-world attacks. Nevertheless, this allows to better understand that the traffic produced by smart speakers when in idle or with the microphone in an inactive state is limited. Specifically, the amount of time needed to collect a suitable amount of data to have statistical relevance could approach one hour.

In general, classifiers such as DT and kNN turn out to be faster in training already for small values of Δt and N. Instead, techniques using NN are the most time-consuming. Specifically, the time needed to train a NN remains constant above a certain threshold, mainly since the overheads for loading data, allocate memory, etc., are greater than the time needed for performing computations. Therefore, in the perspective of launching an attack, less sophisticated algorithms such as DT are the preferred choice. Indeed, the higher speed accounts for reducing times to launch the attack, thus minimizing the chance of being detected.

In this sense, the ability to inject additional information into the traffic to inflate the time needed to train the machine learning mechanism could be considered a prime and valuable countermeasure.

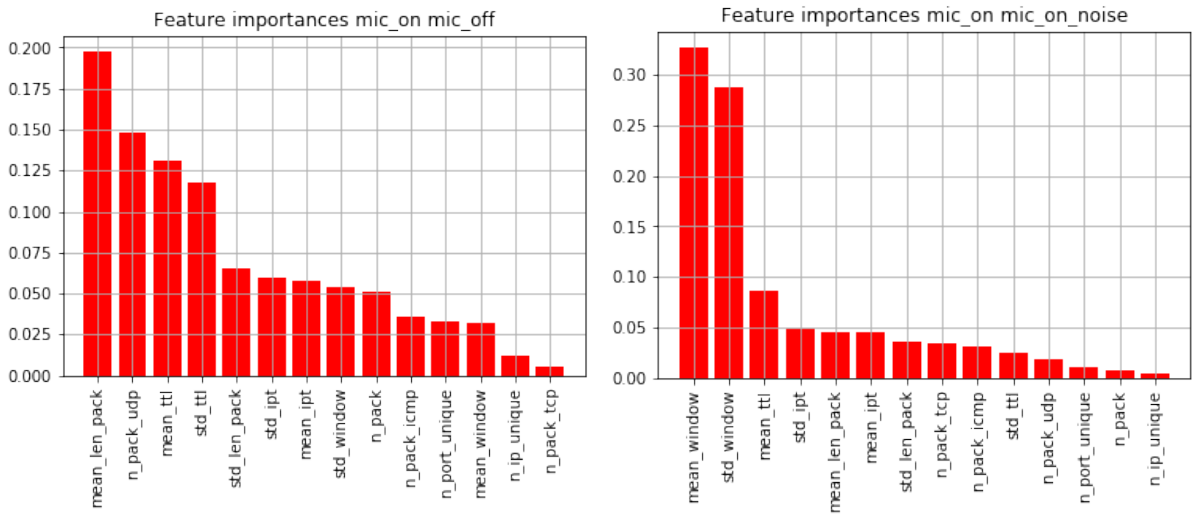
5 Development of Countermeasures

In order to develop appropriate countermeasures and mitigation techniques, we carried out a set of tests to identify the most important traffic features that could be used to feed machine-learning-capable threats. To this aim, we considered all the high-level traffic features introduced in Section 3.3. The results are summarized in Figure 9: for the sake of brevity, we only report results when using the two best classifiers (i.e., AdaBoost with $\Delta t = 500$ and Decision Tree with $N = 15$) as they represent the worst case scenario

Table 2: Training times for the used classifiers with different windows of length N.

N [packet]	Training Time [s] `mic_on - mic_off` kNN	DT	AB	RF	NN	`mic_on - mic_on_noise` kNN	DT	AB	RF	NN
2	21.07	1.46	38.14	158.81	231.0	13.27	1.01	25.84	101.26	257.0
5	1.47	0.62	13.97	63.17	227.0	1.82	0.78	21.08	78.31	238.0
15	0.04	0.18	4.76	18.16	171.0	0.05	0.22	6.91	21.55	181.0
50	0.03	0.05	1.44	4.78	157.0	0.01	0.05	2.02	5.62	161.0
200	0.01	0.01	0.42	1.04	151.0	0.01	0.02	0.58	1.33	151.0
500	0.01	0.01	0.23	0.45	150.0	0.01	0.01	0.31	0.58	150.0
1000	0.01	0.01	0.17	0.25	150.0	0.01	0.01	0.22	0.33	150.0
1500	0.01	0.01	0.15	0.19	150.0	0.01	0.01	0.18	0.24	150.0
2000	0.01	0.01	0.14	0.18	150.0	0.01	0.01	0.17	0.22	150.0
2500	0.01	0.01	0.13	0.17	150.0	0.01	0.01	0.16	0.20	150.0

for the victim (i.e., the attacker uses his/her best tools). To rank the importance of the features, we used the Gini Impurity and Gini Gain metrics [22].



(a) AdaBoost: $\Delta t = 500$ - `mic_off` vs `mic_on`

(b) Decision Tree: $N = 15$ - `mic_on` vs `mic_on_noise`

Figure 9: Importance of traffic features that can be used to leak information about smart speakers.

In the tests aimed at discriminating between the `mic_on` and `mic_off` states, we can notice that according to Figure 9a the characteristics having a greater impact on the performance of the prediction are: *i)* the average length of the packet, and *ii)* the number of UDP datagrams present within the considered window. Therefore, a possible countermeasure could be the insertion of suitable padding inside the packets to normalize the average length as well as the standard deviation. As regards the number of UDP packets, using a unique protocol for the transport could add another layer of privacy.

Besides, Fig 9b reports results when a statistical tool has to recognize the `mic_on` and `mic_on_noise` states. The best features, in this case, that better discriminate the state of a smart speaker are the aver-

age value and standard deviation of the TCP window, the average TTL and the packet length standard deviation. Consequently, a possible mitigation technique could be the alteration of the traffic with the insertion of appropriate "noise", for instance by exploiting some form of traffic camouflage or morphing [20]. Furthermore, we advocate that a security-by-design revision of the architecture used to exchange data between the IVA and the backend infrastructure could act as a suitable countermeasure for those kinds of attacks. Specifically, as shown in Table 1, the reduced amount of traffic produced by the protocol architecture used for the communications between the device and the remote datacenter could make difficult to collect the needed amount of data in a limited time frame. Therefore, further reducing the traffic could not only represent an improvement in term of bandwidth optimization but can also partially void the effectiveness of machine-learning-capable attacks. At the same time, avoiding constant traffic patterns, for instance by deploying port hopping or the aforementioned morphing mechanisms, can be an additional countermeasure built in the software layers responsible of implementing the network services for exchanging data with the IVA.

Lastly, we underline the importance of this kind of analysis, not only in the perspective of understanding the risks of machine-learning-capable privacy threats. In fact, as shown in Figure 4 and Figure 9, the investigation of this class of threats can be used to anticipate the attacker. For instance, it turns out that the TCP window can be considered a sort of privacy-leaking side channel, which transmits a bit describing the state of the surrounding environment. Consequently, such behaviour should be taken into account to prevent attacks or to deploy proactive defence mechanisms to poison the database of the attacker (e.g., in an adversarial machine learning flavour) or to limit the effectiveness of reconnaissance campaigns.

6 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 an 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. 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: *i)* refining our framework by considering smart speakers from other vendors and formally testing the security properties of their protocols [7], and *ii)* investigating other privacy-related attacks, such as identifying the type of queries submitted to profile the users. Besides, we are working towards the implementation of a sort of "warden" able to normalize traffic generated towards the IVA cloud.

References

[1] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and A. S. Uluagac, "Peek-a-Boo: I See Your Smart Home Activities, Even Encrypted!" http://arxiv.org/abs/1808.02741 [Online; Accessed on March 25, 2020], 2018.

[2] E. Alepis and C. Patsakis, "Monkey Says, Monkey Does: Security and Privacy on Voice Assistants," *IEEE Access*, vol. 5, pp. 17 841–17 851, August 2017.

[3] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying ssh and skype," in *Proc. of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA'09), Ottawa, Ontario, Canada*. IEEE, July 2009, pp. 1–8.

[4] Y. Amar, H. Haddadi, R. Mortier, A. Brown, J. Colley, and A. Crabtree, "An Analysis of Home IoT Network Traffic and Behaviour," http://arxiv.org/abs/1803.05368 [Online; Accessed on March 25, 2020], 2018.

[5] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in *Proc. of the 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI'17), Yogyakarta, Indonesia*. IEEE, September 2017, pp. 1–6.

[6] N. Apthorpe, D. Reisman, and N. Feamster, "A Smart Home is No Castle: Privacy Vulnerabilities of Encrypted IoT Traffic," http://arxiv.org/abs/1705.06805 [Online; Accessed on March 25, 2020], 2017.

[7] A. Armando, G. Pellegrino, R. Carbone, A. Merlo, and D. Balzarotti, "From model-checking to automated testing of security protocols: Bridging the gap," in *Proc. of the 6th International Conference on Tests and Proofs (TAP'12), Prague, Czech Republic*, ser. Lecture Notes in Computer Science, vol. 7305. Springer Berlin Heidelberg, May-June 2012, pp. 3–18.

[8] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Z. Yang, "Automatic Device Classification From Network Traffic Streams of Internet of Things," in *Proc. of the 43rd IEEE Conference on Local Computer Networks (LCN'18), Chicago, Illinois, USA*. IEEE, October 2018, pp. 1–9.

[9] P. Biswal and O. Gnawali, "Does QUIC Make the Web Faster?" in *Proc. of the 2016 IEEE Global Communications Conference (GLOBECOM'16), Washington, DC, USA*. IEEE, December 2016, pp. 1–6.

[10] A. Bivens, C. Palagiri, R. Smith, B. Szymanski, M. Embrechts *et al.*, *Network-based intrusion detection using neural networks*. ASME, 2002, vol. 12.

[11] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, October 2001.

[12] Canalysis, "Smart Speaker Prevision," https://www.canalys.com/newsroom/canalys-global-smart-speaker-installed-base-to-top-200-million-by-end-of-2019 [Online; Accessed on March 25, 2020], April 2019.

[13] A. Cardaci, L. Caviglione, A. Gotta, and N. Tonellotto, "Performance Evaluation of SPDY Over High Latency Satellite Channels," in *Proc. of the 5th International ICST Conference on Personal Satellite Services (PSATS'13), Toulouse, France*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 123. Springer, Cham, June 2013, pp. 123–134.

[14] L. Caviglione and F. Davoli, "Peer-to-peer middleware for bandwidth allocation in sensor networks," *IEEE Communications Letters*, vol. 9, no. 3, pp. 285–287, March 2005.

[15] L. Caviglione, "A first look at traffic patterns of Siri," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 4, pp. 664–669, August 2013.

[16] L. Caviglione, M. Coccoli, and A. Merlo, "A Taxonomy-based Model of Security and Privacy in Online Social Networks," *International Journal of Computational Science and Engineering*, vol. 9, no. 4, pp. 325–338, April 2014.

[17] M. Conti, N. Dragoni, and V. Lesyk, "A Survey of Man in the Middle Attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, March 2016.

[18] B. Copos, K. Levitt, M. Bishop, and J. Rowe, "Is Anybody Home? Inferring Activity from Smart Home Network Traffic," in *Proc. of the 2016 IEEE Security and Privacy Workshops (SPW'16), San Jose, California, USA*. IEEE, May 2016, pp. 245–251.

[19] W. Diao, X. Liu, Z. Zhou, and K. Zhang, "Your Voice Assistant is Mine: How to Abuse Speakers to Steal Information and Control Your Phone," in *Proc. of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM'14), Scottsdale, Arizona, USA*. ACM, November 2014, pp. 63–74.

[20] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail," in *Proc. of the 2012 IEEE Symposium on Security and Privacy (S&P'12), San Francisco, California, USA*. IEEE, May 2012, pp. 332–346.

[21] M. Ford and W. Palmer, "Alexa, Are You Listening to Me? An Analysis of Alexa Voice Service Network Traffic," *Personal and Ubiquitous Computing*, vol. 23, pp. 67–79, June 2018.

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

[23] 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 [Online; Accessed on March 25, 2020], April 2019.

[24] S. Li, S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "IoT Forensics: Amazon Echo as a Use Case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, August 2019.

[25] Z. Li, R. Yuan, and X. Guan, "Accurate classification of the internet traffic based on the svm method," in *Proc. of the 2007 IEEE International Conference on Communications (ICC'07), Glasgow, UK*. IEEE, June 2007, pp. 1373–1378.

[26] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, and Y. Elovici, "ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis," in *Proc. of the 2017 Symposium on Applied Computing (SAC'17), Marrakech, Morocco*. ACM, April 2017, pp. 506–509.

[27] T. Melamed, "An active man-in-the-middle attack on bluetooth smart devices," *International Journal of Safety and Security Studies*, vol. 8, no. 2, pp. 200–211, 2018.

[28] A. Merlo, M. Migliardi, and P. Fontanelli, "Measuring and estimating power consumption in Android to support energy-based intrusion detection," *Journal of Computer Security*, vol. 23, no. 5, pp. 611–637, November 2015.

[29] M. Migliardi and A. Merlo, "Modeling the energy consumption of distributed IDS: A step towards Green security," in *Proc. of the 34th International Convention (MIPRO'11), Opatija, Croatia*. IEEE, May 2011, pp. 1452–1457.

[30] R. Mitev, M. Miettinen, and A.-R. Sadeghi, "Alexa Lied to Me: Skill-based Man-in-the-Middle Attacks on Virtual Assistants," in *Proc. of the 2019 ACM Asia Conference on Computer and Communications Security (AsiaCCS'19), Auckland, New Zealand*. ACM, July 2019, pp. 465–478.

[31] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *Proc. of the 2016 3rd International Conference on Electronic Design (ICED'16), Phuket, Thailand*. IEEE, August 2016, pp. 321–326.

[32] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, Fourth Quarter 2008.

[33] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proc. of the 2002 Conference on Empirical methods in natural language processing (EMNLP'02), Philadelphia, Pennsylvania, USA*. Association for Computational Linguistics, July 2002, pp. 79–86.

[34] A. J. Pinheiro, J. d. M. Bezerra, C. A. Burgardt, and D. R. Campelo, "Identifying IoT Devices and Events Based on Packet Length from Encrypted Traffic," *Computer Communications*, vol. 144, pp. 8–17, August 2019.

[35] P. Pongle and G. Chavan, "A survey: Attacks on RPL and 6LoWPAN in IoT," in *Proc. of the 2015 International Conference on Pervasive Computing (ICPC'15), Pune, India*. IEEE, January 2015, pp. 1–6.

[36] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, May 2019.

[37] H. K. D. Sarma and A. Kar, "Security threats in wireless sensor networks," in *Proc. of the 40th Annual 2006 International Carnahan Conference on Security Technology (CCST'06), Lexington, Kentucky, USA*. IEEE, October 2006, pp. 243–251.

[38] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "IoT Devices Recognition Through Network Traffic Analysis," in *Proc. of the 2018 IEEE International Conference on Big Data (BigData'18), Seattle, Washington, USA*. IEEE, January 2018, pp. 5187–5192.

[39] S. A. Shaikh, H. Chivers, P. Nobles, J. A. Clark, and H. Chen, "Network Reconnaissance," *Network Security*, vol. 2008, no. 11, pp. 12–16, November 2008.

[40] A.-m. Yang, S.-y. Jiang, and H. Deng, "A p2p network traffic classification method using svm," in *Proc. of the 9th International Conference for Young Computer Scientists (ICYCS'08), Hunan, China*. IEEE, November 2008, pp. 398–403.

[41] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, April 2017.

[42] M. H. Yılmaz and H. Arslan, "A survey: Spoofing attacks in physical layer security," in *Proc. of the 40th IEEE Local Computer Networks Conference Workshops (LCNW'15), Clearwater Beach, Florida, USA*. IEEE, October 2015, pp. 812–817.

[43] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *Proc. of the 30th IEEE Conference on Local Computer Networks (LCN'05), Sydney, New South Wales, Australia*. IEEE, November 2005, pp. 250–257.

Author Biography

Davide Caputo is a second-year Ph.D. student in Computer Science at the University of Genova, Genova, Italy. He obtained both his BSc and MSc in Computer Engineering at the University of Genova, and now he is working under the supervision of Prof. Alessio Merlo. His research topic focuses on Mobile Security and IoT Security.

Luca Verderame obtained his Ph.D. in Electronic, Information, Robotics, and Telecommunication Engineering at the University of Genoa (Italy) in 2016, where he worked on mobile security. He is currently working as a post-doc research fellow at the Computer Security Laboratory (CSEC Lab), and he is also the CEO and Co-founder of Talos, a cybersecurity startup and university spin-off. His research interests mainly cover information security applied, in particular, to mobile and IoT environments.

Andrea Ranieri is a researcher at CNR-IMATI where he works with Deep Learning mainly in the field of Computer Vision and Time Series Forecasting. Andrea has a degree in Computer Engineering, a Ph.D. in Space Science and Engineering and an extremely varied technological background ranging from networking and distributed systems to robotics and perception. He has participated in many Italian and European research projects (FP6 INTERMEDIA-NoE, PRIN Italian project MARIS, FP7 MORPH, FP7 CADDY and others) and has co-authored many scientific publications in conference proceedings and international journals. His research interests range from the simple classification of images and time series, to unsupervised pre-learning tasks for transfer learning to supervised tasks. `https://orcid.org/0000-0001-8317-3158`

Alessio Merlo got a Ph.D. in Computer Science in 2010 and he is currently serving as a Senior (Tenured) Assistant Professor at the University of Genoa. His main research interests focus on Mobile and IoT Security and he leads the Mobile Security research group at the University of Genoa. He published more than 100 scientific papers in international conferences and journals.

Luca Caviglione received the Ph.D. degree in electronics and computer engineering from the University of Genoa, Genoa, Italy. He is a Research Scientist with the Institute for Applied Mathematics and Information Technologies of the National Research Council of Italy, Genoa. His research interests include optimization of large-scale computing frameworks, wireless and heterogeneous communication architectures, and network security. He is an author or co-author of more than 140 academic publications and several patents in the field of p2p and energy-aware computing. He has been involved in many research projects funded by the European Space Agency, the European Union, and the Italian Ministry of Research. He is a Work Group Leader of the Italian IPv6 Task Force, a contract professor in the field of networking/security and a professional engineer. He is a member of the technical program committee of many international conferences. From 2016 he is an Associate Editor of the International Journal of Computers and Applications, Taylor&Francis