A Dynamic Method to Protect User Privacy Against Traffic-based Attacks on Smart Home

Bruna V. dos Santos[†], **Andressa Vergütz**[*], **Ricardo T. Macedo**[†], **Michele Nogueira**[‡]
[†]Dept. of Information Technology - Federal University of Santa Maria (Frederico Westphalen), Brazil
[*]NR2/CCSC - Federal University of Paraná, Brazil
[‡]Dept. of Computer Science - Federal University of Minas Gerais, Brazil
Emails: bruna.vitoria@acad.ufsm.br, avergutz@inf.ufpr.br, rmacedo@inf.ufsm.br, michele@dcc.ufmg.br

2022 IEEE Latin-American Conference on Communications (LATINCOM) | 978-1-6654-8225-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/LATINCOM56090.2022.10000503

Abstract—**The Internet of Things (IoT) has revolutionized how people interact with their living spaces. However, attackers can perform traffic-based attacks to reveal the behavior of legitimate users, seriously compromising their privacy. Studies have proposed to obfuscate network traffic to avoid these attacks. However, there is still the challenge of ensuring a trade-off between privacy and network overhead. This work introduces the MITRA method to protect user privacy in smart homes while keeping IoT network overhead low. The method relies on the dummy traffic injection following different levels of obfuscation. The different levels mask network traffic, improving privacy without harming network performance unnecessarily. Results show that the obfuscation of IoT device traffic reduces the traffic identification accuracy by up to 42%.**

Index Terms—**IoT, Traffic Obfuscation, Traffic-based Attacks**

I. Introduction

The rapid advances in the Internet of Things (IoT) have revolutionized how people interact with their living spaces [1], [2]. Smart homes containing heterogeneous internet-connected devices have become a part of people's daily life. Forecast pointed out that, by 2023, the number of IoT devices will be up to three times the world population, where 48% of devices consist of smart home devices [3]. IoT devices usually are single-purpose devices with limited capabilities, comprising only a few states or actions. For instance, a motion sensor allows a user to detect any movement in a physical space, but the sensor has only two states: motion and no-motion. If an attacker can reveal the current state of the sensor, s/he will reveal the user's presence at home [4].

IoT devices contain much sensitive information about users and devices [5]. Some devices record user activities like sleeping patterns, exercise routines, and even medical information. Given the limited-purpose nature of many IoT devices, it is often straightforward for an adversary to map changes in traffic rates to a particular user's actions, i.e., perform traffic-based attacks. Governments have created regulations, such as the General Data Protection Regulation (GDPR), to privacy [6]. However, to protect user privacy, any passive adversary must not be able to identify the device type correctly and device states through traffic analysis [1]. However, the limited resources of IoT make this task even more difficult.

There are defense techniques against traffic-based attacks, such as obfuscation techniques based on packet size padding [2], [7], [8] and dummy traffic infection (false traffic generation) [9]–[11]. Some works employ Generative Adversarial Networks (GANs) to mask network metadata [12]. However, such techniques are not enough to protect in-home user privacy without causing too much network overhead. It is crucial to consider IoT constraints, such as power and bandwidth [5], [13]. Hence, there are opportunities for novel IoT defenses against traffic-based attacks that jointly consider computational cost and privacy protection.

This work introduces MITRA, a **M**ethod for **I**ot Ne**T**work **TR**affic Obfusc**A**tion. It improves in-home user privacy under network traffic-based attacks. The MITRA method analyzes the network traffic and dynamically generates different levels of dummy traffic for privacy protection according to the context to avoid unnecessary network overhead. The levels differ by the number of generated packets (e.g., low and random levels). Also, MITRA innovates by creating false devices similar to the original IoT devices. Hence, it masks IoT network traffic, offering different levels of privacy without harming network behavior.

MITRA evaluation follows a trace-driven approach in two steps: *(i)* the characterization of IoT network traffic and *(ii)* traffic obfuscation. Experiments employ the IoT dataset in [14] seeking a realistic environment since it contains real smart home network traffic. MITRA extracts network features and computes statistical measurements. Through classifiers, it identifies IoT devices. Then, MITRA creates dummy IoT devices to generate false traffic packets. This paper presents the results of a comparison between MITRA and a representative packet padding technique from the literature [7]. MITRA obfuscate the IoT devices identification by up to 42%, achieving a performance higher than 20% when compared to the literature, with less than 1% network overhead.

This paper proceeds as follows. Section II presents related works. Section III details MITRE. Section IV describes the performance evaluation and the results. Finally, Section V concludes the work and presents future directions.

II. Related Works

The ability to infer information via packet and flow-level features has been previously established [5]. Recently, a number of techniques have been proposed for IoT device identification [10], [15], [16]. However, it is also possible to identify IoT devices, exhibiting malicious activity, by analyzing their

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.

network traffic [4], [7], [10], [11], [17]. Traffic analysis attacks (also known as traffic-based attacks) have been used to track in-home users' activities.

Traffic-based attacks are usually modeled as a classification problem in which attackers extract features from network traffic and classify devices to infer user activities [12]. Literature works have been proposed several defense techniques against traffic-based attacks such as obfuscation techniques based on packet size padding [2], [7], [8] and dummy traffic infection (false traffic generation) [9]–[11]. The obfuscation technique used varies according to the context. Some works combine both obfuscation techniques [1], [9]. There are also works employing Generative Adversarial Networks (GANs) to camouflage network traffic from Internet apps [12]. However, the combination of techniques or cost mechanisms (GANs) may cause a huge network overhead. IoT limited-nature makes it difficult to obfuscate network traffic and improve privacy.

For instance, the authors in [11] obfuscated user activities by injecting dummy traffic into the IoT network. The authors in [10] employed both dummy traffic injection and packet padding to mask network traffic. However, the authors pointed out that, prior obfuscation, it is essential to analyze the environment available resources to understand the particularities and not cause overhead in a limited network. Furthermore, the authors proposed a mechanism to inject dummy traffic with fixed-size. Nonetheless, it was not considered a previous analysis to define the fixed packet size. IoT network traffic is dynamic and contains variable sizes. It is essential to inject dummy packets based on the original network packet sizes.

The authors in [7] proposed a packet padding mechanism to obfuscate IoT network traffic. They created obfuscation levels to find an equilibrium between privacy and network overhead. The authors in [8] proposed a packet padding technique to obfuscate the packets size through random sizes. The technique selects the target application to change the packets size of the source application based on the target. However, packet padding tends to considerably increase the number of bytes of network packets, increasing the overhead. Therefore, similar to [7], the proposed method employs different levels of network traffic obfuscation to avoid network overhead. However, unlike the literature, this work employs obfuscation levels with dummy traffic injection based on the original network capture, which allows to create dummy devices as similar as possible to the original one, improving obfuscation.

III. The MITRA Method

This section detaisl MITRA, a dynamic method for IoT network traffic obfuscation. It aims to improve user privacy and avoid traffic-based attacks. Different from the literature, MITRA is based on the original network traffic to dynamically create false devices and generate false traffic following different obfuscation levels. Fig. 1 shows the MITRA steps: *i)* network traffic collection, *ii)* network traffic pre-processing, *iii)* device identification, *iv)* dummy devices creation and, *v)* false traffic generation. The following subsections describe each MITRA step.

A. Network Traffic Collection

By sniffers, MITRA passively captures network traffic in a promiscuous mode. Considering a smart home, it is suggested to positioned the MITRA in the home gateway or access point since it is necessary to collect all network traffic generated by IoT devices. The device running MITRA must have access or be accessed by the IoT devices connected to the smart home network. The collected network traffic serves as input for the following MITRA steps, such as network traffic pre-processing and device identification.

B. Network Traffic Pre-Processing

MITRA pre-processes, cleans, and extracts the network features. It ignores network packets belonging to broadcast communication, packets with zero-size, and packets directed to Domain Name System (DNS) because they are not relevant to IoT device identification. The method extracts relevant network traffic features such as the source and destination IP addresses, the transport layer protocol used in the communication, the transmission time instant (timestamp), the source and destination MAC addresses, packet size, and port numbers.

MITRA computes statistical measurements to increase the granularity and amount of samples based on the extracted network traffic features. Statistical measures consist in mean, minimum, maximum, variance, and standard deviation. MITRA considers a sample size equal to five since some IoT devices generate a tiny amount of network packets, e.g., blood pressure devices send an average of three to ten packets per day. Thus, MITRA calculates statistical measurements for every five network packets. Moreover, it filters devices by MAC address to label them. The MAC address serves as ground-truth information to assist the correct device identification. Therefore, MITRA creates a file containing the network traffic features, statistical measurements, and MAC address that serve as input for device identification.

C. Smart Home Device Identification

This step receive as input the network traffic features and simulates the traffic-based attack by identifying the IoT devices. The device identification employs machine learning (ML) algorithms suitable for multi-classification problems, such as Random Forest and Decision Tree [5]. Device identification uses supervised algorithms based on the MAC addresses labels. The ML algorithms follow the holdout method, i.e., it splits the network traffic features into training and testing datasets. Hence, MITRA splits the dataset containing the network traffic features into 60-40, 60% for training and 40% for testing. The trained algorithms predict the devices classes with the testing sample. After segmenting and processing the dataset, MITRA executes the ML algorithms. Note that the success on identifying the IoT devices traffic simulates the traffic-based attacks. Higher the success on the device identification, greater the attack success.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.

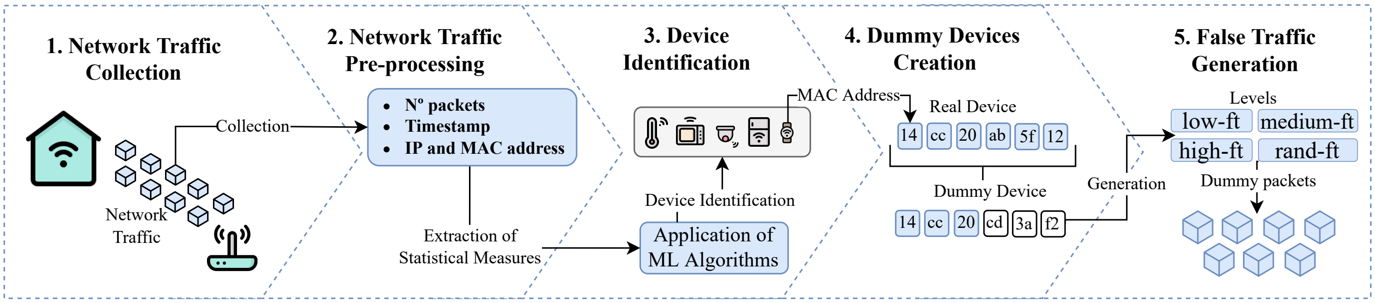


Fig. 1. The MITRA Method Steps

D. IoT Dummy Devices Creation

MITRA creates dummy devices to confuse the attacker. It takes as input the devices identified in the earlier step to create false device as similar as possible to the original ones. Particularly, MITRA uses a part of the real IoT devices MAC address to create dummy devices. This part of the MAC address corresponds to the manufacturer's identifier code (OUI – Organizational Unique Identifier), as shown in Fig. 2. The OUI code of MAC address keeps the same for both real and dummy devices to have similar type of devices. Meanwhile, the other values of the dummy MAC address (i.e., dummy device), represented by Network Interface Controller (NIC), are randomly generated. Thus, the method creates dummy MAC address by using the same OUI code from the real devices and randomly NIC values. In addition to the MAC address, dummy devices receive other dummy data, such as: the device name and the type of network used for communication, e.g., wireless.

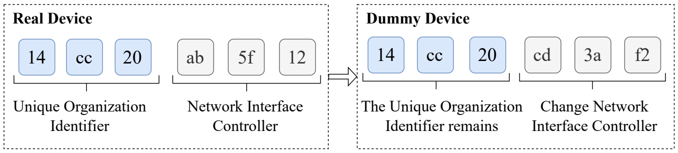

Real Device
14 cc 20 ab 5f 12
Unique Organization Identifier
Network Interface Controller
Dummy Device
14 cc 20 cd 3a f2
The Unique Organization Identifier remains
Change Network Interface Controller

Fig. 2. Creation of Dummy Devices based on the Real IoT Devices

E. False Traffic Generation

MITRA uses the dummy IoT devices to generate false network traffic. The method transmits all false traffic to the in-home gateway identified earlier to follow the usual behavior from the real devices. Thus, the destination of false traffic receives the gateway MAC address and a random IP address. Moreover, MITRA follows packet size and destination port number according to the original network traffic capture. Particularly, the packet size of the false traffic follows the most recurrent value found in the original capture to approximate the dummy behavior to the original. Hence, MITRA uses a fixed destination, and dynamically fills the data referring to the dummy device. Such data refers to the MAC and IP addresses, packet size, port number, among others. For the transmission of false traffic, the method considers the same network interface used in the original traffic. MITRA provides four levels of false traffic generation, low (*low-tf*), medium (*medium-tf*), high (*high-tf*) and random (*rand-tf*), to avoid unnecessary network overhead, as shown in Fig. 3. The *low-tf* level generates the lower amount of false packets when compared to the other levels, such as *medium-tf* and *high-tf* levels. The *rand-tf* level randomly switches between the minimum (*low-tf*) and maximum (*high-tf*) values of false packets, changing the amount of packet for each dummy device. The lowest false traffic generation level aims to reduce the network overhead in situations where the traffic insertion can generate issues in the network behavior. Thus, MITRA allows to dynamically select the obfuscation levels according to the current network state, decreasing the network overhead. For instance, in an idle network, more false traffic can be generated. According to the chosen level, the method generates n network packets for each dummy device. The false traffic generated is injected into the IoT network to obfuscate the original traffic, which make difficult to identify the real IoT devices, and improves the users privacy.

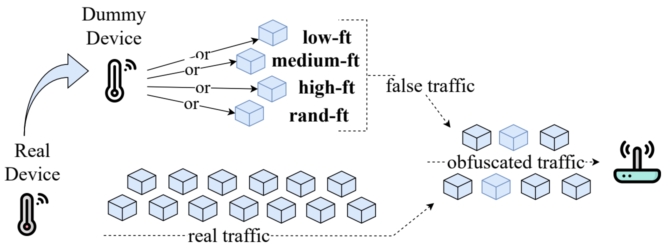


Fig. 3. False Traffic Generation following MITRA Obfuscation Levels

IV. Performance Evaluation

The MITRA performance evaluation follows a trace-based approach under two steps: *i)* the characterization of IoT network traffic, and *ii)* traffic obfuscation. The IoT Traffic Traces dataset was used seeking for a realistic environment since it contains real smart home network traffic [15]. First, MITRA extracts network features and computes statistical measurements. Through classifiers, MITRA identifies the IoT devices. Second, MITRA creates dummy IoT devices to generate false traffic packets. Finally, the method was compared with the packet padding technique from [7]. Next subsections detail the dataset, evaluation methodology, and results.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.

Dataset Details

The IoT dataset contains network traffic from a smart home composed of 31 IoT and non-IoT devices [14]. The capture was from September 22th, 2016 to October 12th, 2016, totaling 20 days of capture with 32GB of network traffic, i.e., 20 packet capture (PCAP) files. The dataset contains devices such as gateway, laptops, cameras, tablet, smart thing devices, BlipcareBP, smoke alert, smart plug, and others. Some devices generated more traffic than others, and such network traffic behavior is usual for the IoT environment. Fig. 4 shows the number of generated packets and the total bytes of the main IoT devices. The blood pressure monitor, followed by the smoke detector, are the IoT devices generated few amount of network traffic. For instance, the smoke alert only generates traffic when someone is smoking near the device, while monitoring cameras generate continuous network traffic. Thus, it is harder to identify the behavior of small packet count.

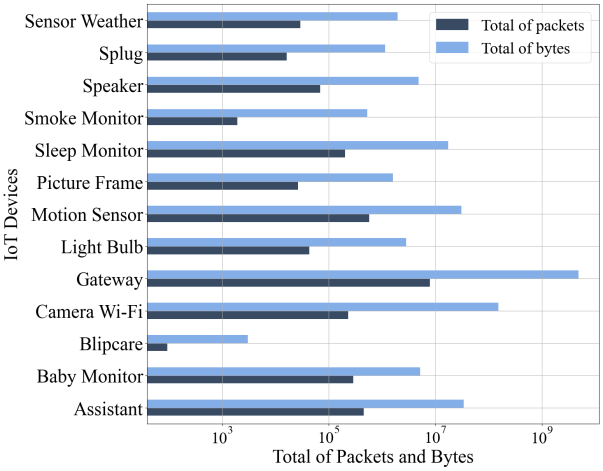


Fig. 4. Total of Packets and Bytes generated by the Devices (Log)

Details on Original IoT Network Traffic Analysis

To analyze the original IoT network traffic, it was extracted the network traffic features through Tshark tool from the dataset. Network features involve MAC address, IP address, transport layer protocol, packet size, number of packets, timestamp, and inter-packet-time (IAT). In this work, IAT consists in the time between two sequential network packets. Furthermore, in data cleaning, the analysis ignores packets destined to DNS servers, broadcast, and packets with size equal to 0 since they are packets without relevant content. The feature extraction and cleaning output consists in 20 CSVs files for each capture day (from 1 to 20 days) containing the packets network features and the device label. After extracted the network traffic features, from the CSVs, we compute statistical features to increase the dataset details, such as: *mean, standard deviation, min, max, sum*, and *median*. To calculate the statistical features, we group the packets belonging to the same device and create small samples of the network traffic features (samples size = 5) because some devices generated only five packets per day (e.g., BlipcareBP). Hence, we create small samples to not loss information from devices that generated small amount of traffic. Then, we merge all devices features, including the device label, into one unique final file. This final file serve as input for device identification.

For device identification, we consider the following supervised algorithms: Extreme Gradient Boosting (XGBoost), Classification and Regression Trees (CART), Random Forest, and Bootstrap Aggregating (Bagging), since they are commonly used in the literature for traffic identification [2], [4], [7]. The performance analysis of ML algorithms followed the metrics of accuracy, F1-Score, recall, and precision. These metrics are based on rate of true positives (VP), true negatives (VN), false positives (FP), and false negatives (FN). We analyzed device identification performance through these evaluation metrics. The device identification simulates the traffic-based attack on IoT network traffic; hence, higher the performance identification results, higher the attack success.

Details on Network Traffic Obfuscation

For network traffic obfuscation, MITRA creates dummy IoT devices based on the MAC address of the original IoT devices. MITRA generates false network traffic data by Ostinato[1] tool. It creates one false MAC address for each real MAC address through the Python Generate MAC library[2]. Hence, as the domestic IoT traffic dataset contains 31 devices, it was created 31 false devices, totaling 62 devices. Periodically, when the dataset contains more than 60 false MAC addresses, MITRA remove them from having a sizeable false dataset. The false devices receive random labels to use in the ML algorithms. Moreover, the false traffic generation followed a constant rate (i.e., sending packets at a fixed frequency), static value for packet size of 120 bytes, and destination port number 443, since the IoT original devices commonly used these values.

MITRA follows different levels of false traffic injection to compare the difference between the small and large amount of false network traffic on obfuscation efficiency. The levels are *low-ft*, *medium-ft*, *high-ft*, and *rand-ft*, as shown in Tab. I. It was generated 100, 600, and 1000 dummy packets per device for the *low-ft*, *medium-ft*, and *high-ft* levels, respectively. While, the *rand-ft* level adds a dynamic amount of packets ranging from 1000 to 10000 false packets per device. Hence, in the *low-ft* level, MITRA generates 100 packets per dummy device, totalling 3000 dummy packets from the 30 false devices. Finally, false and real traffic are shuffled into the same dataset for the network feature and statistics extraction, and submission to ML algorithms. Thus, we compare the IoT device identification results with the original traffic (without obfuscation) *versus* traffic with MITRA obfuscation.

MITRA was compared to the packet padding technique from [7]. The authors proposed a packet padding technique to obfuscate IoT network traffic. They created obfuscation levels for packet size padding: 100, 500, 700, and 900, where the

[1] Ostinato: http://ostinato.org/. Accessed on Apr/2022.

[2] Python-generate_mac v.1.3: https://pypi.org/project/python-generate-mac/. Accessed on Apr/2022.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.

TABLE I
False Traffic Generation Levels

Levels	Number of Packets p/ Device	Packets Total
low-ft	100 false packets	3.000
medium-ft	600 false packets	18.000
high-ft	1000 false packets	30.000
rand-ft	Between 50 - 1000 false packets	Random

numbers denote the minimum length of the packets that will belong to the respective level after padding. The lower the level, the fewer the number of extra bytes inserted into the packets. For instance, network packets with sizes smaller than or equal to 100, are now 100 bytes long. As the authors, we also compared MITRA method to others padding technique well-known in the literature, such as: linear, exponential, mouse elephant, MTU, random, and random 255. The authors in [7] used the same dataset, and presented similar obfuscation levels, allowing a fair obfuscation comparison.

A. Results

Fig. 5 presents the performance results of ML algorithms on the IoT device identification considering only the original network traffic. In general, all classifiers presented high rate of success in identifying the traffic of IoT devices (≈ 65% and ≈ 70% of precision, and F1-score). The success in the IoT device identification allows attackers to learn the users behavior. Once the device is identified, the attacker monitors its traffic to learn about the user's routine.

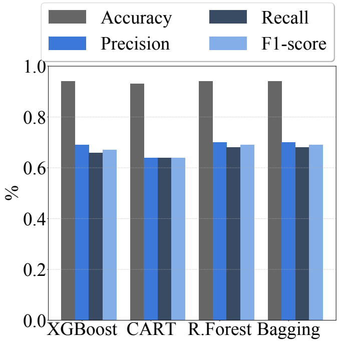
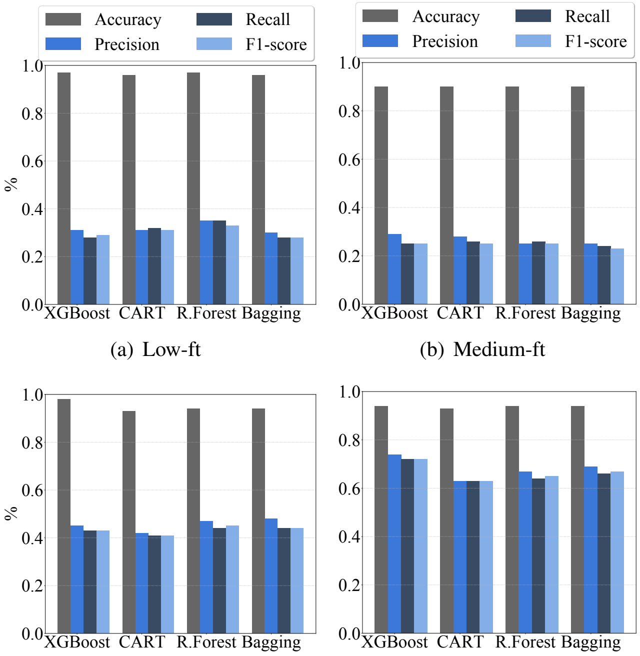


Fig. 5. Packet Padding Literature Results

Fig. 6 shows the MITRA performance on obfuscating IoT device traffic following the four levels of false traffic (ft) generation (*low-ft*, *medium-ft*, *high-ft*, and *rand-ft*). Fig. 6(a) presents the *low-ft* obfuscation level results, where MITRA achieved ≈ 30% of recall and F1-score, respectively. These results show the feasibility of MITRA on masking IoT device traffic, reducing by up to 30% when compared to the identification results without defenses against traffic-based attacks (Fig. 5). Fig. 6(b) corroborates the MITRA efficiency by applying the *medium-ft* level, reaching ≈ 28%, ≈ 25% of precision, and F1-score, respectively. Note that accuracy achieved high results (≈ 90%) because it represents the overall success on device identification, without considering the different amount of traffic from each device class, not impacting on obfuscation.



(a) Low-ft
(b) Medium-ft
(c) High-ft
(d) Rand-ft

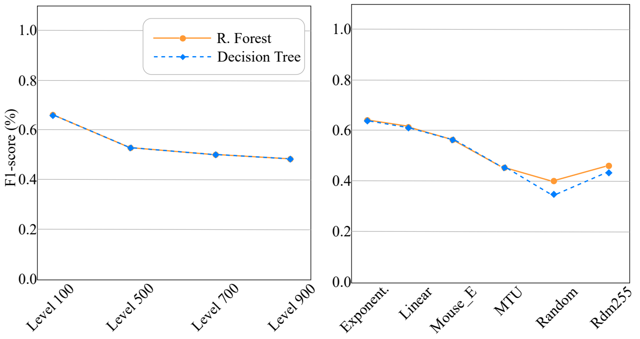
Fig. 6. MITRA Results on Obfuscation Levels

In contrast to the lowest obfuscation levels, the *high-ft* obfuscation level increased the performance results, as shown in Fig. 6(c). As the MITRA sends additional traffic to hide real traffic, we expected the heavier the false traffic, the higher the network traffic obfuscation. However, the obfuscation results showed the contrary. As many traffic samples are given to the supervised classifier, better it is the performance. A rich training dataset, with large network traffic samples, increases the probability of a classifier identifying the devices with success. Fig. 6(d) confirms that by increasing the amount of false traffic, reduce the obfuscation efficiency, reaching around 66% of accuracy and F1-Score in *rand-ft* obfuscation level. In *rand-ft*, MITRA generated more than 30,000 false packets, increasing the number of samples for the classifiers. Thus, the *medium-ft* level obtained the best performance result in network traffic obfuscation, decreasing by up 42% of classifiers performance. Moreover, when applying the *low-ft* and *medium-ft* levels, MITRA uses less computational resources since it generates less amount of traffic, which is an excellent result for the IoT environment. Lower the false traffic added to the network, the lower the overhead.

Fig. 7 presents the obfuscation results by applying the packet padding technique from [7]. Fig. 7(a) shows the obfuscation results following the 100, 500, 700, and 900 obfuscation levels. Each level filled the packet size according to the number level. The highest level of obfuscation (900) presented the best obfuscation result when compared to the other levels, achieving ≈ 46% of F1-Score. However, the 900 level increments all network packet sizes to 900 bytes, increasing the

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.

overhead. Fig. 7 presents the obfuscation results by applying well-known packet padding techniques from the literature such as Exponential (Exp), Linnear (Lin), Mouse_Elephant (M_E), Maximum Transmission Unit (MTU), Random (R), and Random255 (R255). The Random and MTU showed the best obfuscation performance, reaching around 30% and 40% of F1-Score, respectively. When compared to the MITRA obfuscation results, the *high-ft* level obtained similar results to the 900 level from [7] and MTU. The lowest level 100 from [7] did not achieve a satisfactory result (≈ 63% of F1-Score). Therefore, MITRA *low-ft* level achieved the best obfuscation result, making it difficult to identify IoT devices.



(a) Packet Padding from [7] (b) Well-known Techniques

Fig. 7. Packet Padding Literature Results [7]

Tab. II shows the network overhead caused by obfuscation. MITRA *medium-ft* level increased the network overhead in 0.82% by adding 600 false packets per dummy device. In contrast, the 500 level from [7] resulted in 602% of network overhead, which is critical for IoT. Therefore, we conclude that MITRA presented the best performance in network traffic obfuscation. These results can be observed in terms of privacy since the device identification was reduced to 20% by MITRA. Finally, MITRA network overhead was less than 1%.

TABLE II
NETWORK OVERLOAD COMPARISON OF OBFUSCATION LEVELS

Traffic Type	Traffic Size	% of Network Overhead
Original Traffic	269,41 MB	0%
MITRA tf-low	269,79 MB	**0,13%**
MITRA tf-medium	271,65 MB	**0,82%**
MITRA tf-high	273,13 MB	1,3%
[7] 500	1622,9 MB	602,2%
[7] 700	2271,49 MB	843,1%
[7] 900	2920,49 MB	1084%

V. CONCLUSIONS

This paper introduced MITRA, a dynamic method to protect in-home user privacy against traffic-based attacks. MITRA uses the original IoT network traffic and follows different levels of false traffic generation to avoid unnecessary network overhead. MITRA evaluation followed a trace-oriented approach, taking as input a smart home dataset. Results indicated the feasibility of MITRA to protect user privacy. MITRA reduced the device identification by up to 42%, improving in 20% the obfuscation performance when compared to the literature [7]. Moreover, MITRA achieved less than 1% of network overhead. As future work, it is expected to evaluate MITRA under experimental IoT environments.

VI. ACKNOWLEDGMENT

We are grateful for the financial support of the Brazilian agency: Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for processes '313844/2020-8 and '432204/2018-0 that supported the registration of this paper.

REFERENCES

[1] J. Brahma and D. Sadhya, "Preserving contextual-privacy for smart home IoT devices with dynamic traffic shaping," *IEEE Internet of Things J.*, pp. 1–1, 2021.

[2] N. Prates, A. Vergütz, R. T. Macedo, A. Santos, and M. Nogueira, "A defense mechanism for timing-based side-channel attacks on IoT traffic," in *Proc. of the GLOBECOM*. IEEE, 2020, pp. 1–6.

[3] Cisco, "Cisco Annual Internet Report (2018–2023) White Paper." Available in: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html, 2018-2023, accessed April, 2022.

[4] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, and S. Uluagac, "Peek-a-boo: I see your smart home activities, even encrypted!" in *Proc. of the WiSec*. ACM, 2020, pp. 207–218.

[5] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[6] EU, "General Data Protection Regulation (GDPR)," Available in: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG, 2018, accessed April, 2022.

[7] A. J. Pinheiro, P. F. de Araujo-Filho, J. d. M. Bezerra, and D. R. Campelo, "Adaptive packet padding approach for smart home networks: A tradeoff between privacy and performance," *IEEE Internet of Things J.*, vol. 8, no. 5, pp. 3930–3938, 2021.

[8] L. Chaddad, A. Chehab, I. H. Elhajj, and A. Kayssi, "Optimal packet camouflage against traffic analysis," *ACM Trans. on Privacy and Security (TOPS)*, vol. 24, no. 3, pp. 1–23, 2021.

[9] T. Datta, N. Apthorpe, and N. Feamster, "A developer-friendly library for smart home IoT privacy-preserving traffic obfuscation," in *ACM Workshop IoT S&P*. ACM, 2018, pp. 43–48.

[10] 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 *Symposium on Security and Privacy*. IEEE, 2012, pp. 332–346.

[11] K. Yu, Q. Li, D. Chen, M. Rahman, and S. Wang, "Privacyguard: Enhancing smart home user privacy," in *IEEE/ACM IPNS*. ACM, 2021, pp. 62–76.

[12] J. Li, L. Zhou, H. Li, L. Yan, and H. Zhu, "Dynamic traffic feature camouflaging via generative adversarial networks," in *Proc. of the Commun. and Netw. Security (CNS)*. IEEE, 2019, pp. 268–276.

[13] M. Alyami, I. Alharbi, C. Zou, Y. Solihin, and K. Ackerman, "WiFi-based IoT devices profiling attack based on eavesdropping of encrypted wifi traffic," in *IEEE CCNC*. IEEE, 2022, pp. 385–392.

[14] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, "IoT traffic traces," Available: https://iotanalytics.unsw.edu.au/iottraces, 2021, accessed April, 2022.

[15] ——, "Classifying IoT devices in smart environments using network traffic characteristics," *IEEE Trans. on Mobile Comput.*, vol. 18, no. 8, pp. 1745–1759, 2018.

[16] A. Vergütz, I. Medeiros, D. Rosário, E. Cerqueira, A. Santos, and M. Nogueira, "A method for identifying ehealth applications using side-channel information," in *Proc. of the GLOBECOM*. IEEE, 2019, pp. 1–6.

[17] N. Apthorpe, D. Y. Huang, D. Reisman, A. Narayanan, and N. Feamster, "Keeping the smart home private with smart (er) IoT traffic shaping," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 3, pp. 128–148, 2019.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO PARANA. Downloaded on May 15,2023 at 19:35:54 UTC from IEEE Xplore. Restrictions apply.