A Non-Intrusive Method for Smart Speaker Forensics

Li Lin†, Xuanyu Liu†, Xiao Fu⋆, Bin Luo
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
{mf1932111, xuanyuliu}@smail.nju.edu.cn
{fuxiao, luobin}@nju.edu.cn

Xiaojiang Du
Dept. of Computer and Information Sciences
Temple University
Philadelphia, PA, USA
dxj@ieee.org

Mohsen Guizani
Dept. of Computer Science and Engineering
Qatar University
Qatar
mguizani@ieee.org

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500679

***Abstract*—With the rapid development of the Internet of Things technology, smart speakers have become increasingly popular. However, smart speaker security is an ensuing threat. At present, smart speakers are activated by voice, and they monitor users' voices 24 hours per day. Consequently, there may be problems with user privacy leakage. In this paper, we propose a non-intrusive digital forensic method for smart speakers. The main contribution of the paper is an effective method of combining network traffic analysis with the extraction of user intent and alarms about abnormal network traffic to support the investigation of security. We use Xiaomi smart speakers as an example in an experiment to verify our forensic method. The evaluation results show that our method works well for detecting security risks.**

***Index Terms*—digital forensics, smart speaker, non-intrusive, network traffic, users intent**

I. Introduction

In the past few years, Internet-connected consumer devices, which comprise the Internet of Things (IoT), have rapidly increased in penetration and availability. In this paper, the IoT device of interest is the smart speaker. Many smart speakers are available on the market, including the Xiao du Smart Speaker 1S, Xiaomi smart speaker, Tmall Elf Sugar R, Amazon Echo, Google Home, and Huawei AI Speaker. Smart speakers can provide users with genuine convenience, but they also pose security issues. Smart speakers usually work with a connection to a cloud server. It is thought that even if a user does not activate their smart speaker, the speaker may eavesdrop on their daily background conversation and surreptitiously upload it to the cloud server, thereby invading user privacy [1]. Sensitive information, such as passwords, credit card numbers, and home addresses, is vulnerable to being revealed.

Many methods have been proposed to protect user privacy and enhance the security of smart speakers; among these, digital forensics seems to be effective [2]. Nevertheless, existing forensic methods for smart speakers are usually intrusive, either changing smart speakers or requiring support from the cloud server. Smart speakers are not open source, and the cloud server itself may be suspect. A noninvasive research method that can work independently is therefore a better choice. We use network traffic analysis to monitor smart speaker events because the network traffic of IoT devices usually has a fixed pattern [3]. To verify whether the current events of a given smart speaker are expected, we also need to extract the user intent in order to understand their commands.

In this paper, we propose a non-intrusive forensic method for smart speakers, which combines network traffic analysis and the extraction of user intent. For the network traffic analysis, we inspect the network traffic between the smart speaker and the cloud server using a middleman method to capture network traffic patterns. To extract the user intent, we use a device with a microphone to monitor and record the user's voice. The voice is then translated into text in real-time, and matched keywords are found through Natural Language Processing (NLP) technology which is a subfield of artificial intelligence. The user's intent is extracted based on the keywords and compared with the current network traffic pattern of the smart speaker to determine whether the smart speaker event is abnormal. The forensic module records this information for further forensic analysis and warns the user about discovered risks or anomalies.

We use the Xiaomi smart speaker [4] to explore the security issues of smart speakers and to conduct the forensic analysis because it is one of the most popular smart speakers on the market. Xiaomi smart speakers can interact with various IoT devices and third-party applications by converting voice requests into native communication protocols. Xiaomi smart speakers are closed source, making them suitable for our non-intrusive forensic method.

Contributions:

- We verify the one-to-one mapping between traffic patterns and smart speaker events. We propose a non-intrusive method to inspect the network traffic of smart speakers and extract user intent.
- We propose a forensic method to protect user privacy and enhance the security of smart speakers. It can warn users about risks or anomalies.
- We use the Xiaomi smart speaker to verify our non-intrusive forensic method. The results show that our

†: These authors contributed to the work equally; ⋆: Corresponding author

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

method works well and that it can detect security risks.

II. Related Work

A. Traffic Analysis of IoT Devices

For the network traffic analysis of IoT devices, several researchers have proposed methods and models [5]–[7]. nPrint, which is a standard representation of network traffic based on data packets, can be used as input to train various machine learning models without extensive functional design [8]. It has been demonstrated that nPrint can provide a suitable traffic representation for machine learning algorithms, which can solve three common network traffic classification problems: device fingerprinting, operating system fingerprinting, and application recognition. Many smart home devices use encryption. However, Internet Service Providers (ISPs) those can provide services such as dial-up Internet service, Internet browsing, downloading files, receiving and sending emails or other network observers can infer privacy-sensitive family activities by analyzing the network traffic of smart homes that contain commercially available IoT devices. Several strategies for mitigating the privacy risks associated with smart home device traffic have been evaluated, including blocking, tunneling, and rate shaping. Experiments have shown that traffic shaping can effectively reduce the privacy risks associated with smart home IoT devices [3]. Unlike the above, a new defensive measure called ”random traffic filling” (STP) has been developed, making it difficult for passive network opponents to reliably correlate real user activity with its associated traffic patterns from behavior that looks like user interactions.

B. Digital Forensics for Smart Speakers

Recently, many methods have been proposed for the digital forensics of smart speakers. The digital forensic method of the Amazon Alexa ecosystem uses an effective new method that combines cloud forensics, client forensics, and device forensics to support actual digital investigations [9]. Based on an in-depth understanding of the target ecosystem, a proof-of-concept tool, CIFT, which supports the identification, acquisition, and analysis of local artifacts in the cloud and local devices (e.g., mobile applications and web browsers) has been proposed [10]. Another automatic voice traffic collection tool collected two large-scale datasets on two smart speakers, the Amazon Echo and Google Home [11]. A proof-of-concept attack was then implemented using deep learning. The experimental results from these two datasets show that privacy issues are an issue for smart speakers. Several papers (e.g., [12]–[14]) have studied related issues.

C. Voice Analysis

There has been considerable research on voice analysis, such as on optimized language models (LMs). A novel technology, namely, the optimized N-gram (Op-Ngram), uses an end-to-end N-gram pipeline to effectively utilize mobile resources to achieve faster word completion (WC) and next-word prediction (NWP). Compared with the BerkeleyLM, the LM-ROM size of Op-Ngram is improved by 37%; the LM-RAM size is improved by 76%; the load time is improved by 88%, and the average suggestion time is improved by 89% [15]. In NLP, part-of-speech tagging (POS tagging) and blocking may be the right choice for keyword preprocessing; the resulting keywords can then provide the required, essential text documents [16].

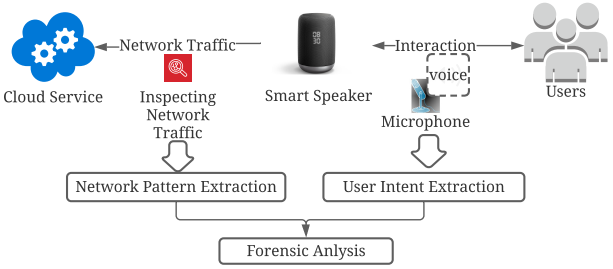


Fig. 1. The three modules of the forensic method for smart speakers and how they work together.

III. System Overview

In this section, we represent the problems we aim to solve, the modules of our method, and how our method works.

A. Threat Model

Human voices contain a wealth of personal information. Through voice content, analysts can infer sensitive personal information such as the user’s age, gender, birthplace, or health. Voice collection by smart speakers thus involves user privacy issues. Even if a user has not awakened their smart speaker, they may eavesdrop and upload their conversations to a third-party server. We assume that the network traffic of smart speakers has not been tampered with and that the network condition of smart speakers is average. We also assume that the smart speakers function naturally, i.e., that they are not hiding their network traffic patterns.

B. Modules

Our goal is to develop a non-intrusive forensic method for smart speakers to monitor abnormal events; the method includes three main modules, as shown in Fig. 1.

1) Network Traffic Analysis Module: This module inspects the network traffic between a smart speaker and the cloud server and analyzes the network patterns. The network traffic is inspected through a man-in-the-middle method. We use a device with ARP spoofing software to act as the middleman, which captures all the network packets between the smart speaker and the cloud server. Based on the captured network packets, network patterns can be extracted, and the current event of the smart speaker can be inferred. The inferred event is then used to determine if the network traffic is consistent with user instructions or if the smart speaker is secretly recording the user in the background.

2) User Intent Extraction Module: This module aims to determine the user intent, that is, what the user wants the smart speaker to do. A recording device with a microphone is deployed to record conversations between the smart speaker and the user. The user's voice is converted into text in real time, and their intent is extracted from the text by keyword matching. After obtaining the user intent, we can verify whether the smart speaker event matches the user expectations.

3) Forensic Analysis Module: This module combines network traffic analysis with the user intent to monitor the smart speaker events and collect relevant forensic evidence. The current events of the smart speaker can be obtained from the results of the network traffic analysis, and the current user instructions to the smart speaker can be obtained from the extracted user intent. If the event and the instruction are inconsistent, the event is considered abnormal. When an abnormal situation occurs, the user is alerted, and the forensic details can be viewed via a user interface.

C. Workflow

In this section, we discuss the details of how our forensic method works. A device with a microphone is deployed near the smart speaker to continuously monitor the conversation between the smart speaker and the user. When the user says something, voice recognition technology translates what they said into text. The text is then analyzed by NLP for keyword matching to determine what the user wants to do (e.g., the user wants the smart speaker to play a song). This information is summarized as the user intent. Simultaneously, the middleman device with ARP spoofing software continues to monitor the network traffic during the smart speaker and the cloud service and infer smart speaker events by analyzing the network patterns under different events (e.g., the network pattern indicates that the smart speaker is playing music).

The forensic analysis module collects and preserves the inferred smart speaker events and the user intent. The events and user intent that occur at the same time are then compared to discover whether the smart speaker is following the user's instructions. For example, if the network pattern indicates that the smart speaker is playing music and the user truly wants the smart speaker to play a song, the current smart speaker event is expected. However, if we find that the smart speaker is transmitting data to the cloud server when the user does not want to interact with it, the event is abnormal. When there is an abnormal event, we alert the user, who can view the forensic details to decide what to do.

IV. Forensic Analysis of Smart Speakers

A. Network Traffic Analysis

In this section, we introduce a method for inspecting the network traffic between a smart speaker and a cloud server and propose how to analyze network traffic patterns.

1) Inspecting Network Traffic: We inspect network traffic to obtain a more intuitive one-to-one mapping between network patterns and smart speaker events. Because smart speakers are a closed source, it is more difficult to obtain their traffic information directly; hence, it is necessary to use an intermediary to implement it indirectly. The preparation for this step is the installation of the ARP spoof software [17] on the Ubuntu system. We then use the ARP spoof software to deceive the smart speaker. The principle of ARP spoofing is that two hosts can communicate through MAC-Address addressing in the same local area network. If the two hosts are not in the same subnet, they can communicate with each other. The data are transmitted to the respective router gateway, and the IP address of the gateway is used to achieve communication [18]. However, when the gateway communicates with hosts in its LAN, it still relies on MAC-Address addressing; therefore, if the attacker fakes the gateway to achieve deception, the MAC address cache of the gateway on the target host must be changed to the MAC address of the attacker.

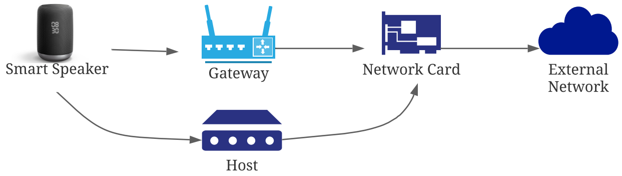
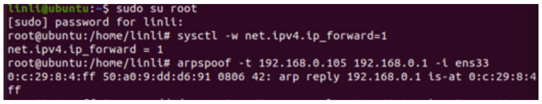


Fig. 2. The process of inspecting network traffic using ARP spoofing.



```
linli@ubuntu:~$ sudo su root
[sudo] password for linli:
root@ubuntu:/home/linli# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@ubuntu:/home/linli# arpspoof -t 192.168.0.105 192.168.0.1 -i ens33
0:c:29:8:4:ff 50:a0:9:dd:d6:91 0806 42: arp reply 192.168.0.1 is-at 0:c:29:8:4
ff
```

Fig. 3. Command line for traffic forwarding.

Under normal circumstances, when communicating between two different subnets, the traffic flow is "*SmartSpeaker→ Gateway→ NetworkCard→ ExternalNetwork.*" The two subnets can be connected through a gateway and the network card. We changed this traffic flow to "*SmartSpeaker→ Host→ NetworkCard→ ExternalNetwork.*" We fake the host into a gateway. All the data from the smart speaker to the external network passes through the host and is analyzed. Fig. 2 shows how we inspect network traffic using ARP spoofing.

After the gateway has been forged, the data must be forwarded. Otherwise, the smart speaker will not gain access to the Internet and may find that the gateway is forged. The command line for traffic forwarding is shown in Fig. 3. Next, we open Wireshark [19], which can help analyze the traffic network that we have already captured in real-time. The information observed from the traffic packet mainly includes the packet length, packet direction, and timestamps. We can also obtain plaintext information from the packet header, which involves the source address, the destination address, used protocol, and port number. These metadata are used to extract network patterns for smart speaker events.

2) Extracting Network Patterns: In this section, we mainly analyze the one-to-one mapping relationship between network traffic patterns and the events of smart speakers. We analyze

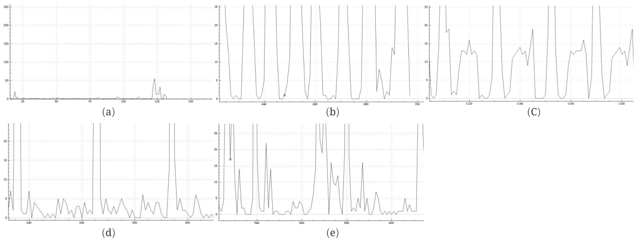


Fig. 4. Network traffic patterns in different scenarios: (a) the transition from a resting state to a working state, (b) chatting with the smart speaker, (c) playing an audiobook, (d) playing music, and (e) an interactive state for riddles or idioms.The x-axis shows the timestamps in seconds. The y-axis shows the number of packets transmitted per second.

the events of smart speakers by observing the number of packets passing through the traffic packet capture software in each period. The number of traffic packets that pass in a certain period or the size of the packets can, to a certain extent, be used as a measure of smart speaker instructions. Fig. 4 shows I/O graphs of the events of a Xiaomi smart speaker.

Fig. 4(a) shows the Xiaomi smart speaker transitioning from a resting state during which no one had been talking, to a working state during which someone is talking, but not by using ”Xiao Ai” to wake up the Xiaomi smart speaker. In the first two situations, almost no traffic packets are generated; occasionally, a small number of TCP traffic packets that maintain the communication between the Xiaomi smart speaker and the server pass. When waking up the smart speaker, there is a higher peak during which the number of traffic packets increases. This interaction leads to an increase in the number of traffic packets.

Fig. 4(b) shows the traffic status when chatting with the speaker. When the user asks a question or sends a corresponding instruction to the Xiaomi smart speaker, it responds to the user request. As a result, more traffic information passes through. In this image segment, there are dense peak traffic packets, indicating that the interactive dialogue between the user and the Xiaomi speaker is relatively intensive.

Fig. 4(c) shows the traffic status when the user instructs the speaker to play an audiobook. When switching between different audiobooks, more packets are generated. After the playback status is stable, the traffic is significantly reduced.

Fig. 4(d) is similar to Figure 3(c), but it depicts music switching. When switching to a new piece of music, the traffic increases sharply, whereas the traffic in the normal playback state is low, much less than for audiobooks.

Fig. 4(e) shows the interactive state of guessing riddles and playing idiom solitaire. Riddles are cultural products created by the collective wisdom of the working people in ancient China. They mainly refer to hidden words that imply things or words for people to guess. Furthermore, idiom Solitaire is a traditional word game of the Chinese nation. There are various rules for idiom solitaire. It is generally known to use the method of connecting the prefix and ending of the idiom to continuously extend the solitaire. The Xiaomi smart speaker

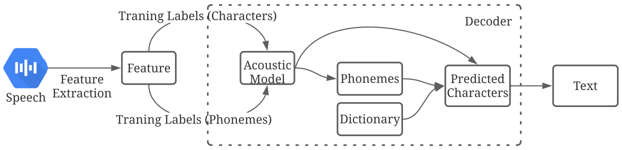


Fig. 5. Flow chart of the principle of speech recognition.

and the user both have a higher dialogue peak, and then there tends to be less traffic.

Through repeated verification of a large number of controlled experiments, the I/O graphs that correspond to given user instructions match the graphs above. That is, each set of user instructions forms specific network flow patterns. In addition, based on the information for each traffic packet, when asking the Xiaomi smart speaker to play a song, the byte stream of the network traffic from the source address to the destination address is about 248 bytes, and the size of the network traffic from the destination address back to the source address is approximately 54 bytes. The byte stream size of the network traffic transmission of other events also fluctuates within a specific range, making it possible to judge smart speaker events based on the network traffic patterns.

From the one-to-one mapping relationship between network traffic patterns and smart speaker events, we can say that the network traffic characteristics of different events have approximately a fixed pattern and that similar instructions may show similar flow characteristics. That is, network traffic patterns can be used to infer the events of smart speakers.

B. *Extraction of User Intent*

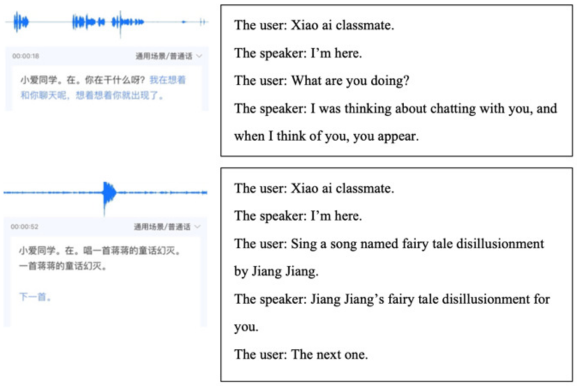
By extracting the user intent, we can determine what the user wants the smart speaker to do and check whether the smart speaker follows their instructions. There are several alternative methods for extracting the user intent.

- We can obtain the cached data from the smart speaker app that records the conversations between the smart speaker and the user. However, the cached data may be encrypted, and some apps may not even preserve these data. Therefore, this method has specific limitations and is not universal.
- We can obtain the UI components of the message dialog box from the smart speaker app to analyze the dialogue between the smart speaker and the user in order to extract the user intent. The typical way to obtain UI component data is to use UI automation [20]. However, the disadvantage of this method is that the data under the UI interface are not always obtainable, and it requires the app to continue running.
- We can also extract the user intent by analyzing the user’s voice. By obtaining the content of the conversation, recording it and converting it into text, and matching the text with keywords, the user intent can be extracted. This method is relatively mature and applicable to a wide range of scenarios without technical challenges.

In this paper, we chose to analyze the user's voice to extract their intent. We used a recording device with a microphone to monitor and record the conversation between the user and the smart speaker, and we used an end-to-end automatic speech recognition system implemented in TensorFlow [21] to translate the monitored speech into a text format. Fig. 5 shows the structure of a typical speech recognition system.

First, as input, the voice enters the ASR system in the form of audio. Then after feature extraction, it is transformed into a voice feature vector for subsequent processing and recognition. Subsequently, the speech feature vector and the label are given to the acoustic model (the core module of the entire ASR system) for training. Note that there are usually two different forms of label data. One uses convention orthography, and the second uses phonemes, which are the basic units of sound in human language. When the trained acoustic model performs prediction in the test phase, it outputs the result according to the label used during its training. If the convention orthography is used as the label, the model will return the convention orthography. If the phoneme is used as the label, the model will spit out the phoneme. If the final goal is to translate the text, the phoneme output is not sufficient. Therefore, a phoneme-to- convention orthography dictionary is usually used for translation to ensure that the final result is in the form of convention orthography. With the output as described above, the final result is equivalent to words and sentences.

To some extent, the conversion from speech to text is complete at this point. KenLM [15] is an excellent open-source language model. Usually, the acoustic model, dictionary, and language model in the dashed box are included, and the decoder is called the input. The input of the decoder is the voice, and the output is the final text. We then use NLP to match keywords to understand the user's instructions. Keywords are words that can express the main content of a document [14]. They are often used in computer systems to index the content of papers, for information retrieval, and for system collection for readers to review. Keyword extraction is a branch of text mining, and it is among the primary work of text mining research, which includes text retrieval, document comparison, abstract generation, document classification, and clustering. From an algorithmic perspective, there are two main types of keyword extraction algorithms: unsupervised keyword extraction methods and supervised keyword extraction methods. In this study, we use a topic model-based method to match keywords. Topmine [22] is a typical keyphrase extraction method based on topic models. It aggregates adjacent words in the text into phrases based on the results of topic analysis and then selects high-frequency phrases as key phrases. The key phrases obtained by this method are more readable than keywords. We first collect a set of user instructions that are typically used to interact with Xiaomi smart speakers based on the functions provided by the speakers. Then, the instructions text is segmented by phrase mining, and a topic model is executed on it. Thereafter, we associate the same topic with each word in the text, and the topic that successfully matches the phrase is determined to be the keyword. For example, we match the subject of a word to the parsed text verbatim. If it is found that the played song matches the existing theme of the lexicon, it means that the match is successful.



00:00:18 通用场景/普通话

小爱同学。在。你在干什么呀？我在想着和你聊天呢，想着想着你就出现了。

The user: Xiao ai classmate.

The speaker: I'm here.

The user: What are you doing?

The speaker: I was thinking about chatting with you, and when I think of you, you appear.

00:00:52 通用场景/普通话

小爱同学。在。唱一首蒋蒋的童话幻灭。一首蒋蒋的童话幻灭。

下一首。

The user: Xiao ai classmate.

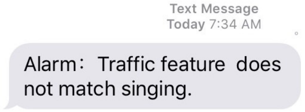
The speaker: I'm here.

The user: Sing a song named fairy tale disillusionment by Jiang Jiang.

The speaker: Jiang Jiang's fairy tale disillusionment for you.

The user: The next one.

Fig. 6. Diagram of the process of transcribing voice into text in real time.



Text Message
Today 7:34 AM

Alarm：Traffic feature does not match singing.

Fig. 7. A warning text message.

C. Smart Speaker Forensics

For smart speaker forensics, we integrate the network traffic analysis and extraction of user intent. For example, suppose a user interacts with their Xiaomi smart speaker, asking it to play a song. The microphone records the user's voice, and the user intent extraction module recognizes the intent. Simultaneously, the network traffic analysis module infers the current of the smart speaker from the network patterns. If the two are consistent, the communication between the smart speaker and the cloud server is regular. Otherwise, if the communication is abnormal, an alarm is sent to the user. Users can view detailed forensic information through the provided UI. The abnormal network traffic flow is marked in red in the I/O graph.

By comparing the user intent with the smart speaker events, we can determine whether the smart speaker is working as expected. In addition, if our microphone detects sensitive keywords from the user, the events of the smart speaker will be monitored to determine whether there is abnormal outbound network traffic flow. In this way, our method can help protect user privacy, avoid information leakage, and enhance the security of smart speakers.

When encountering abnormal traffic conditions, we could also block the abnormal traffic. However, because blocking traffic might hinder the regular operation of the smart speaker, this method is not trivial, and we leave it as future work.

V. EXPERIMENT AND ANALYSIS

In this section, we introduce the verification of the feasibility of our non-intrusive forensic method and analyze the influence of performance factors on the experimental results.

Our experiment was conducted in a large living room with three users. We used a Xiaomi smart speaker, a Windows desktop computer to inspect the network traffic between the Xiaomi smart speaker and the cloud server, and a mobile phone to record the users' voices. Users interacted with the Xiaomi smart speaker as they would in their own home.

Fig. 6 shows that we can transcribe the voice to text in real-time to analyze a user's instructions to the Xiaomi smart speaker. On the left is the real-time transcribed Chinese content, and on the right, the form is translated into English. In the first example of a dialogue, the content of the chat with the Xiaomi smart speaker is displayed. Through keyword matching, it is determined that this is a part of chatting with the user. In the second example of dialogue, the main display indicates that the user is giving instructions to the Xiaomi smart speaker to play a song, and it lets different types of music to be played. Thus, for this dialogue, the matching instruction is an instruction to play music.

To verify the feasibility of non-intrusive methods for smart speaker forensics, we conducted 108 experiments, of which 6 included abnormal traffic. One reason for abnormal traffic is that the Xiaomi smart speaker might have accidentally been offline. When our method observes that the traffic flow pattern does not match the user instruction, a message is sent to the user to alert them. Fig. 7 shows the text message that is sent when the traffic flow pattern does not match singing. If a user wants to view more details about the abnormal network traffic information, they can examine the I/O graphs in the user interface. If the network flow pattern does not match the user instructions, the abnormal flow is marked in red in the graph. If the user mentions sensitive information, and there is an abnormal increase in traffic, the abnormal traffic is marked in yellow. In the experiment, we found that there was a 0.2-millisecond error in the real-time transcription of speech. This corresponds to a time difference between the inferred smart speaker events and the extraction of the user intent. However, this time error does not affect our conclusion.

The evaluation results show that our non-intrusive forensic method can work well and meets our requirements of protecting user privacy and enhancing security.

VI. CONCLUSION

We proposed a non-intrusive forensics method for smart speakers. In our method, the one-to-one mapping between the traffic patterns and events of smart speakers is verified, and the network traffic of smart speakers is inspected via a middleman approach. Our method uses a device with a microphone to record a user's voice and then uses an end-to-end automatic speech recognition system implemented in TensorFlow and NLP to understand the user's instructions. We combined these steps to verify whether the smart speaker is doing what the user wants it to do. If the network traffic flow is abnormal, an alarm is issued.

REFERENCES

[1] E. Alepis and C. Patsakis, "Monkey says, monkey does: security and privacy on voice assistants," *IEEE Access*, vol. 5, pp. 17 841–17 851, 2017.

[2] M. Zakariah, M. K. Khan, and H. Malik, "Digital multimedia audio forensics: past, present and future," *Multimedia tools and applications*, vol. 77, no. 1, pp. 1009–1040, 2018.

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

[4] "Xiaomi smart speaker," https://www.mi.com/aispeaker.

[5] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving ddos attack detection using cross-domain traffic in software defined networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 628–643, 2018.

[6] J. Brown and X. Du, "Detection of selective forwarding attacks in heterogeneous sensor networks," in *2008 IEEE International Conference on Communications*. IEEE, 2008, pp. 1583–1587.

[7] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3901–3909, 2019.

[8] J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "nprint: A standard data representation for network traffic analysis," *arXiv preprint arXiv:2008.02695*, 2020.

[9] H. Chung, J. Park, and S. Lee, "Digital forensic approaches for amazon alexa ecosystem," *Digital Investigation*, vol. 22, pp. S15–S25, 2017.

[10] M. Merrill, "An uneasy love triangle between alexa, your personal life, and data security: Exploring privacy in the digital new age," *Mercer L. Rev.*, vol. 71, p. 637, 2019.

[11] C. Wang, S. Kennedy, H. Li, K. Hudson, G. Atluri, X. Wei, W. Sun, and B. Wang, "Fingerprinting encrypted voice traffic on smart speakers with deep learning," *arXiv preprint arXiv:2005.09800*, 2020.

[12] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," *Information Sciences*, vol. 479, pp. 640–650, 2019.

[13] N. Wang, X. Zhou, X. Lu, Z. Guan, L. Wu, X. Du, and M. Guizani, "When energy trading meets blockchain in electrical power system: The state of the art," *Applied Sciences*, vol. 9, no. 8, p. 1561, 2019.

[14] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "Bpds: A blockchain based privacy-preserving data sharing for electronic medical records," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[15] S. Mani, S. V. Gothe, S. Ghosh, A. K. Mishra, P. Kulshreshtha, M. Bhargavi, and M. Kumaran, "Real-time optimized n-gram for mobile devices," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE, 2019, pp. 87–92.

[16] R. S. Dudhabaware and M. S. Madankar, "Review on natural language processing tasks for text documents," in *2014 IEEE International Conference on Computational Intelligence and Computing Research*. IEEE, 2014, pp. 1–5.

[17] S. Whalen, "An introduction to arp spoofing," *Node99 [Online Document], April*, 2001.

[18] A. M. Amin and M. S. Mahamud, "An alternative approach of mitigating arp based man-in-the-middle attack using client site bash script," in *2019 6th International Conference on Electrical and Electronics Engineering (ICEEE)*. IEEE, 2019, pp. 112–115.

[19] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[20] X. Liu, X. Fu, B. Luo, X. Du, and M. Guizani, "Monitoring user-intent of cloud-based networked applications in cognitive networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[21] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition," *Speech Communication*, vol. 108, pp. 15–32, 2019.

[22] A. El-Kishky, Y. Song, C. Wang, C. Voss, and J. Han, "Scalable topical phrase mining from text corpora," *arXiv preprint arXiv:1406.6312*, 2014.