Understanding Digital Forensic Characteristics of Smart Speaker Ecosystems

Xuanyu Liu, Ang Li, Xiao Fu*, Bin Luo
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
{xuanyuliu, mf20320076}@smail.nju.edu.cn
{fuxiao, luobin}@nju.edu.cn

Xiaojiang Du
Dept. of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, NJ, USA
dxj@ieee.org

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

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685816

***Abstract*—With a built-in intelligent personal voice assistant providing Q&A services, smart speaker ecosystems combine multiple compatible components, including the internet of things (IoT) technology, mobile devices, and cloud computing. However, as it is closely related to people's daily lives, security and privacy issues have gained worldwide attention. Components in the ecosystem are interconnected and chained together to enable the ecosystem to perform increasingly diverse operations. By collecting meaningful data from smart speaker ecosystems, we can reconstruct user behavior and provide a holistic explanation for finding the root cause of an observable symptom. This highlights the need for digital forensic research to enhance the security and privacy of smart speaker ecosystems. In this paper, we first discuss the digital forensic characteristics of a smart speaker ecosystem. Then, we propose a proof-of-concept digital forensic tool based on *data provenance*, that supports the identification, acquisition, and analysis of client-side artifacts from local devices.**

***Index Terms*—Digital forensics, smart speaker ecosystem, data provenance**

I. Introduction

User-friendly smart speaker ecosystems continue to proliferate with their ubiquitous use, improving user convenience for a better life. In contrast, the expansion of smart speaker ecosystems brings about new challenges in terms of security and privacy [1]–[3]. The observable abnormal symptoms (e.g., an IoT device in an error state), need to be backtraced to identify root causes to explain the nature of attacks, intrusions, or misconfigurations.

The smart speaker ecosystem is an excellent source of forensic evidence for enhancing security due to its always-on operation mode [4]. Components such as smart speakers, IoT devices, mobile devices, and cloud services are chained together in long sequences of trigger-condition-action rules based on voice commands, pre-configured automation rules, operations in applications, and instructions from cloud servers. However, it is nontrivial to perform digital forensics among smart speaker ecosystems [5] because the ecosystem, comprising interconnected compatible components, is complex and heterogeneous. Fortunately, various digital forensic shreds of evidence exist, and they require distinct data acquisition methods and a global forensic analysis view.

Cloud-side evidence acquisition is significant because the cloud server functions as a controller of the entire ecosystem. Numerous pieces of data, valuable for forensic analysis, are stored in the cloud, for example, voice recognition metadata, personal information, and runtime status [6]. However, obtaining cloud-side evidence is a difficult task. The cloud server does not readily provide undisclosed data owing to financial interests and user privacy policy. Therefore, due to the challenges associated with the acquisition of cloud-side evidence, the demand for client-side forensic approaches has dramatically increased, increasing the amount of controlled data on local devices. Thus, in this paper, we first discuss the digital forensic characteristics of the smart speaker ecosystem and then propose a proof-of-concept digital forensic tool focusing on client-side data. We introduce *data provenance* to correlate heterogeneous forensic evidence and provide a comprehensive analysis. We believe that our findings and the proposed tool can help forensic investigators enhance the security of smart speaker ecosystems.

II. Related Work

A. Digital Forensics for Smart Speaker Ecosystem

Recently, several methods have been proposed for digital forensics for smart speaker ecosystems. Chung et al. combine cloud forensics, client forensics, and device forensics for the Amazon Alexa ecosystem [5]. Merrill et al. propose a proof-of-concept tool, CIFT, which supports the identification, acquisition, and analysis of local artifacts, i.e., mobile applications and web browsers in their setting [7]. Lin et al. introduce a non-intrusive forensic tool for anomaly detection by comparing users' voice commands with the smart speaker's network patterns based on traffic fingerprinting [8]. Existing methods usually focus on limited data types [9], [10], whereas our proposed tool can support multiple types of client-side forensic evidence and correlate them for a comprehensive analysis.

*: Corresponding author

978-1-7281-8104-2/21/$31.00 ©2021 IEEE

B. Data Provenance

By converting individual forensic evidence into causal relationship graphs to describe the history of actions taken on a data object from its creation up to the present state, data provenance enables forensic investigators to reason about the interrelationships between different events and objects [11], [12]. Using data provenance, the forensic investigator can trace back from a given security indicator to its entry point and trace forward to determine the scope of influence, i.e., what other actions have been taken on the system. Recent studies have attempted to use data provenance to generate causal associations between IoT devices within a smart home, which can help understand user behavior, explain the reason for errors, and discover attack traces [13], [14]. It is foreseeable that data provenance is also suitable for smart speaker ecosystems when used as a type-independent data format to represent causal relationships among various actions, events, and data.

III. Smart Speaker Ecosystem Forensics

In this section, we first present the smart speaker ecosystem, and then discuss its digital forensic characteristics.

A. Components of Smart Speaker Ecosystem

The smart speaker ecosystem is a cloud-based intelligent voice assistant system with a centric smart speaker in conjunction with nearby IoT devices, mobile apps, and various internet-based services. The entire ecosystem can be divided into seven distinct parts, as shown in Fig. 1.

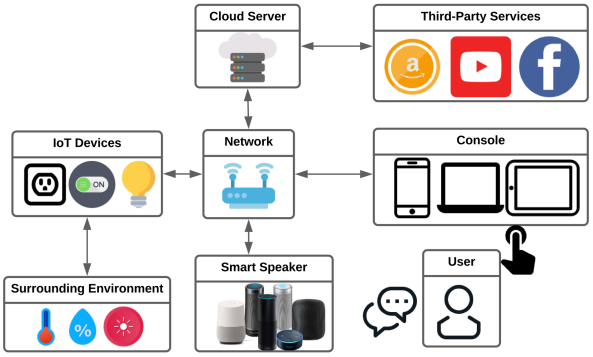


Fig. 1. Components of a smart speaker ecosystem.

1) Consoles: The device vendor usually officially offers a console (e.g., a mobile application or website) to control its product. The console provides graphical interfaces for configuring system settings, paring IoT devices, inspecting running status, and looking over auditing logs. The user can also interact directly with the intelligent voice assistant via a mobile app without a smart speaker.

2) Smart Speakers: A smart speaker is usually a standalone device containing a loudspeaker, microphone, and touch screen. It is one of the vital entrances for users to access intelligent voice assistants. Its primary functions include converting audio data and network packets bidirectionally, establishing communication to the cloud server, and interoperability with nearby devices.

3) IoT Devices: IoT devices are designed for specific purposes, such as detecting temperature changes or turning on the light. Users can include compatible IoT devices in a smart speaker ecosystem using the console via device pairing. These devices are controlled by the users' voice commands or operations on the console.

4) Physical Environment: The smart speaker ecosystem may cause changes to the surrounding environment (i.e., physical properties), such as temperature, illumination, and humidity, by the actions of the IoT devices. For instance, when the user asks the smart speaker to turn on the heater, the temperature of the surrounding environment may increase. In contrast, changes in physical properties may also generate responses from nearby devices [15].

5) Cloud Servers: The cloud server is the control center responsible for user authentication, decision making, and the Q&A service of the intelligent voice assistant. For example, in sounds, commands are parsed into user requests via an AI-powered audio analysis technique in the cloud server. The server then responds to these requests (e.g., the weather today) and generates appropriate answers. In addition, large amounts of valuable data are stored in the server, such as personal information and auditing logs that are beneficial for the forensic analysis of smart speaker ecosystems.

6) Third-Party Services: Some third-party services join the cloud server to enrich the functions of the smart speaker ecosystem, such as posting a status on a social platform or buying food from an online shopping site.

7) Network Communications: Consoles, smart speakers, IoT devices, cloud servers, and third-party services are interconnected via network communications (e.g., WiFi, ZigBee, and BLE). Network data are transmitted between them, such as login verification, device status, commands, and responses, to collaborate on a task.

B. Digital Forensic Characteristics

Based on the smart speaker ecosystem introduced above, we discuss its digital forensic characteristics, indicating that meaningful data from different components can be deemed forensic evidence.

1) Mobile Forensics: Mobile applications have become a highly available and convenient console for users. A considerable amount of data correlated with the smart speaker ecosystem is deposited naturally in mobile applications, such as system configurations and device lists. Data not directly reachable in other devices can also be accessed, such as conversations between users and the smart speaker. It is imperative to obtain these artifacts and consolidate them, along with forensic evidence from other components.

2) Smart Speaker Hardware Forensics: As the core component of the smart speaker ecosystem, the smart speaker can be decomposed to perform hardware-level analysis. Clinton et al. introduced their research on reverse-engineering Amazon Echo at the hardware level [16]. They explained some possible methods for enabling access to the internal parts, including soldered memory chips, though with no details about the

device's data. Since smart speakers are proprietary and close-sourced, performing hardware forensics in a non-intrusive manner is quite challenging.

3) IoT Forensics: A user can manipulate IoT devices with the help of a voice assistant or console. The actions and states of IoT devices can be used to decide whether the execution result of a user command is reasonable. Thus, it is vital to perform digital forensics on IoT devices. For instance, we can inspect the network traffic of these devices in the case of any intrusion or abnormal state to identify policy violations. Furthermore, IoT devices may affect (or be influenced by) the surrounding environment, physical properties, which can help further understand the influence of user commands and even latent security risks. For example, the unexpected turning on of a heater increases the surrounding temperature, making the user feel uncomfortable.

4) Cloud Forensics: The cloud server, which functions as the "brain" of the smart speaker ecosystem, is the control and data center. It stores valuable data in the form of auditing logs regarding the working of the entire ecosystem (e.g., how user commands can be parsed and executed, how the cloud can create a response to a user request, and how different components communicate with each other). However, these data are not publicly available, as cloud servers have safety considerations and legal regulations. Furthermore, forensic investigators cannot readily expect cooperation from cloud servers to obtain data. Analogously, the abovementioned situations apply to third-party services. In summary, the data in the cloud is not easily accessible.

5) Network Forensics: Because the smart speaker, IoT devices, and the console communicate with the cloud, we can utilize network communication to analyze the type of data being transmitted and verify whether the conveyed command is correct. However, most network packets associated with forensically meaningful evidence are transferred over an encrypted connection. For example, smart speakers use HTTPS for communication with the cloud. Man-in-the-Middle (MITM) methods can be used to decrypt protected network communications with traffic analysis to identify anomalies. In addition, network fingerprinting is another option via a side-channel analysis, as packets of different devices usually show unique network patterns [8].

6) User Intent Analysis: Precisely understanding what a user needs the smart speaker ecosystem to do (i.e., user intent) can help verify whether the final execution result meets the user's expectations. Talking to the voice assistant and operating on the consoles are two fundamental means for users to interact with the smart speaker ecosystem and convey their intent. The defects of the voice recognition technology or the bugs in the consoles may cause the smart speaker ecosystem to misunderstand user intent. User intent can be analyzed based on what the user utters and his operations on the consoles.

IV. System Overview

In this section, we discuss our motivation, introduce the data provenance model, and present the modular design of our forensic tool.

A. Motivations

Smart speaker ecosystems face two primary security risks:

- **Anomaly**:The ecosystem may have an abnormal status, for example, when IoT devices appear in error states or when the smart speaker or cloud server does not serve. In addition, the maloperation or misconfiguration of a user can also cause an abnormal status.
- **Vulnerability**: An adversary may exploit the vulnerabilities of the ecosystem. For example, a defective console or a malicious cloud server may steal privacy-sensitive information secretly, or a hacker may hijack the smart speaker for eavesdropping.

Components of a smart speaker ecosystem are interconnected to perform an increasingly diverse range of activities. Although challenging, the observable symptom could be back-traced through a chain of actions to identify the root cause and explain the nature of attacks and anomalies. Therefore, we must locate forensic evidence to answer the following questions, which can help verify whether user expectations are met and whether results caused by actions are reasonable.

- Q_1: What actions does the user require the ecosystem to perform? What kind of feedback is expected?
- Q_2: How does the ecosystem handle user requests and perform corresponding actions?
- Q_3: What are the final symptoms caused by the actions? For example, what are the current states of the IoT devices, what are the responses from the smart speakers, and what are the invocations of applications and services?

Considering the three questions mentioned above, we design our forensic tool with the following requirements:

- **Integrity**: It can logically explain all causal relationships among the actions, events, and states within the smart speaker ecosystem.
- **Validity**: Collected data should be accurate with no ambiguity so that they can be used for forensic explanations.
- **Minimality**: The tool must be invasive for straightforward deployment on existing smart speaker ecosystems. In addition, redundant data should be discarded to decrease storage overhead, with only the evidence retained.

B. Defining the Data Provenance Model

We select *data provenance* as the unified data format of our tool, based on the Open Provenance Model [17], following the requirement of causal analysis. With this model, we can utilize forensic data in a type-independent manner, enabling the combination of causal relationships across multiple objects. Therefore, the following definitions are introduced to represent the data provenance of smart speaker ecosystems:

- **Agent**: An agent indicates the creator or target of an action, e.g., user, mobile application, smart speaker, IoT device, cloud server, or third-party service.
- **Entity**: An entity indicates the intermediate state or carrier of data, e.g., command, network message, question, response, or device event.

- **Action**: An action indicates the behavioral correlation between an agent and entity, e.g., talking, touching the screen, and transmitting network packets.

We define a *data provenance graph* as a directed acyclic graph, whose vertices represent agents, entities, and actions with edges indicating the relationships among the elements. A sample of the data provenance graph is shown in Fig. 2. We can then perform backward and forward tracking from a symptom node in these data provenance graphs to find the root cause and understand the influence.

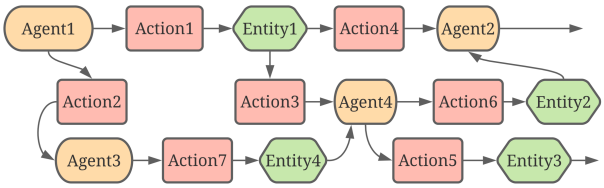


Fig. 2. A sample of a data provenance graph.

C. Modular Design

A modular design is used to decouple the acquisition, management, and analysis of forensic data from smart speaker ecosystems, including four modules, as shown in Fig. 3:

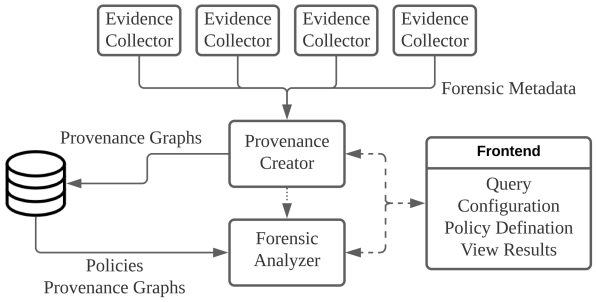


Fig. 3. Modules and workflow of the forensic tool.

1) Evidence Collector: This module collects forensic metadata from the different components of a smart speaker ecosystem. An individual collector is inadequate owing to the various types of metadata and interactions between these components. Therefore, multiple collectors are distributed across diverse parts to gain a global view of the entire ecosystem for integrity.

2) Provenance Creator: Forensic metadata, gathered by *evidence collectors*, are encapsulated into the data provenance format and then merged to build data provenance graphs, indicating causal relationships. These data are stored in a database for forensic analysis. In addition, it provides interfaces for external access to the provenance graphs.

3) Forensic Analyzer: This module is responsible for performing security enforcement by analyzing provenance graphs based on predefined policies. It describes sequences of causal relationships among the components of the smart speaker ecosystem, explains the root cause and impact of a symptom, and performs specified actions (e.g., an alarm). The policies contain rules about how the ecosystem should appropriately function, e.g., trigger-condition-action correlations between IoT devices and keyword blacklists of sensitive data. It can check data provenance graphs against the set of policies at runtime or afterward.

4) Frontend: The frontend works as a console, for example, a mobile app or a website, that provides a graphical interface for users to interact with the modules of our forensic tool. It allows users to set configurations, define policies, query data provenance graphs, and view forensic results.

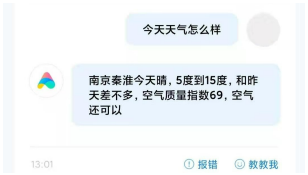
V. Implementation

Data stored in the cloud are not always attainable because of the cloud server's low eagerness to cooperate. Thus, we focus on extracting and analyzing client-side forensic evidence from accessible local devices, leaving cloud-side forensic analysis as future work. We select the *Xiaomi smart speaker* [1], one of the most widespread smart speakers in China, to investigate its forensic characteristics and verify our tool.

A. Local Forensic Data Collection

The console (i.e., an android application in our setting) and network communication are two critical components of a smart speaker ecosystem with easily accessible data. In this section, we present how our *evidence collectors* extract forensic data from these components.

1) Analyzing Android Applications: An official android application is publicly available for users to interact with the Xiaomi smart speaker, where two critical sets of information exist. One is the conversation between the user and the voice assistant, and the other is the auditing logs of the application. **Conversation.** The conversation indicates what the user said to the voice assistant (e.g., questions and commands) and includes the response of the smart speaker ecosystem (e.g., answers and actions); this helps answer Q_1 and Q_3. Example pieces of the conversation are shown in Fig. 4(a).



(a)

```
32434  11615  - 04-04 13:00:44.802 -
I/ConversationService - notifyTopAppChanged:
com.sina.weibo
32434  11615  - 04-04 13:00:59.893 -
I/ConversationService - notifyTopAppChanged:
com.miui.home
32434  11615  - 04-04 13:01:38.274 -
I/ConversationService - notifyTopAppChanged:
com.miui.home
```

(b)

Fig. 4. Fragments of conversations and logs.

Based on our observations, the android application of the Xiaomi smart speaker does not offer a specific file that contains the text content of the conversations. Instead, as the conversations can be seen via graphical user interfaces (GUI), extracting texts from the GUI is an alternative, with two scenarios. Typically, the GUI of an application, such as the document object model (DOM) tree of a website, consists of various components with attributes. Thus, the text content

[1]https://www.mi.com/aispeaker

of the conversations can be accessed from the attributes of the corresponding GUI elements. The GUI analytical tool, Layout Inspector[2], is used. However, some applications may utilize scalable vector graphics (SVG) to render their GUI, meaning that the GUI mentioned above analytical tool does not work. Therefore, we continuously obtain screenshots of the application and use the optical character recognition (OCR) technology [18] to extract texts from the screenshots.

After obtaining the text content (i.e., sentences) of the conversations, we must understand their meanings and distill critical information, as the volume of raw data is enormous. Therefore, sentences are processed based on natural language processing (NLP) technology, including three sequential steps: data preprocessing, corpus training, and feature extraction. For data preprocessing, Jieba[3], a Python Chinese word segmentation module, is used for word segmentation and filtering out stop words. Then, the Word2Vec[4] model is selected for the corpus training of the preprocessed sentences. Finally, the word vector technology of Word2Vec is adopted for feature extraction based on the trained corpus. Subsequently, we can obtain significant keywords from the text content and understand the communication process between the user and the voice assistant.

Auditing Log. An unencrypted auditing log file of the application is available, which records detailed information about the performed actions and runtime status. Thus, we can access the plain text content organized in a well-defined data format. Each entry in the log file can be summarized as a tuple of $(timestamp, service, action, object)$. For example, $Service$ points out the invoked function to perform the $Action$, targeted at the $Object$; $Timestamp$ indicates when this $Action$ occurred. These tuples can assist in answering Q_2 and Q_3. An example of an auditing log file is presented in Fig. 4(b). A shell script is developed to automatically handle the auditing log file and parse its entries into tuples, helping us understand what happens in the underlying system.

2) Inspecting Network Communications: The smart speaker and android application communicate with the cloud server via the HTTP or HTTPS protocol. We can obtain valuable information that is not readily available elsewhere by collecting and analyzing the network data.

To analyze HTTPS-encrypted communication, the Fiddler web proxy[5], a secure HTTP protocol decoder, is used to decrypt network traffic utilizing the MITM technique. Since HTTPS requires the use of a web proxy, Fiddler's certificate is installed on the phone. Then, we can view the contents of the packets after intercepting the certificate used by the HTTPS protocol through Fiddler. Furthermore, side-channel analysis, that is, network fingerprinting, can be a supplementary approach to overcome the shortcomings of parsing encrypted network traffic [8]. We configure a testbed to capture and analyze packets from the smart speaker and android applications. As an access point (AP), we use a laptop to mediate communication between the smart speaker, android application, and cloud server, as shown in Fig. 5. All network data are bidirectionally communicated via the AP. Subsequently, the AP inspects network packets in real-time.

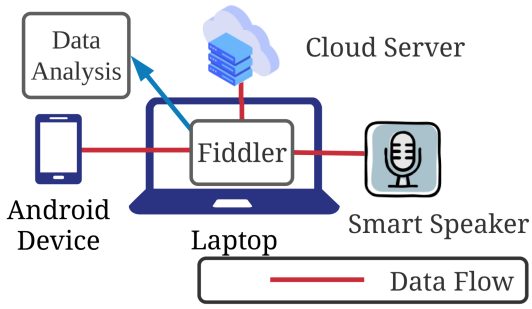


Fig. 5. Using Fiddler to inspect encrypted network traffic.

The experimental results show that valuable data from the network traffic of the android application can be obtained. However, we find that, in our testbed, Fiddler cannot extract clear-text data from the network traffic of the smart speaker. Fortunately, because the android application is in charge of configuring the smart speaker and setting the communication between the smart speaker and the cloud server, an observation is that the smart speaker's status is continuously synchronized with the android application to keep the user informed at all times. Therefore, inspecting the network communication of the android application is sufficient for forensic purposes. We plan to parse the network traffic of the smart speaker in our future work.

In addition to common artifacts extracted from network analysis, such as login information, access tokens, and commands, an interesting finding is that extra data not found in conversations or logs can be obtained. For example, a user asks for music with a response from the voice assistant. This action is recorded in the conversations as sentences (e.g., "play music XX") and in the log file as an entry. Although these records are adequate to represent the action, the concrete contents of the execution are not included; that is, is the correct song played in the previous example. Instead, as shown in Fig. 6, the metadata of the music being played, such as lyrics, are parsed out.

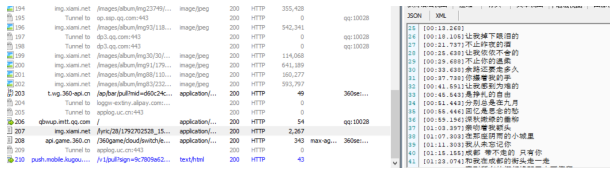


Fig. 6. An example of parsing out the lyrics.

B. Forensic Analysis

The obtained forensic data are then converted into data provenance graphs using the *Provenance Creator*. It is worth

[2]https://developer.android.com/studio/debug/layout-inspector
[3]https://pypi.org/project/jieba
[4]https://www.tensorflow.org/tutorials/text/word2vec
[5]https://www.telerik.com/fiddler

noting that forensic data from auditing logs are at the action level; they do not contain the content of processed data, whereas those from conversations are at both action and data levels. Additional data level information, not shown in the conversations and logs, is extracted from the network communications to complete the description of the actions, as data are transmitted between the components via the network connections. In addition, network analysis results are used to re-verify the accuracy of the data extracted from the conversations and auditing logs in the application.

Type-independent nodes (i.e., agents, entities, and actions) are created based on keywords from the conversations, tuples in the auditing log file, and network analysis results. Edges, which indicate the relationships among the nodes, are determined according to contextual and temporal information. An example of a data provenance graph is presented in Fig. 7.

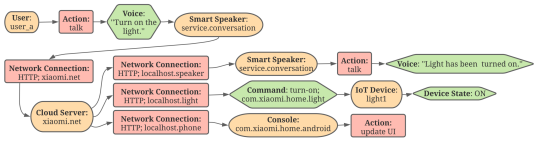


Fig. 7. A user asks the smart speaker to turn on the light.

Based on the generated data provenance graphs, the *Forensic Analyzer* understands what the user wants the system to perform, the underlying operations, and the results of the system's actions. Following this, the *Forensic Analyzer* discovers risks based on security policies (predefined by the tool or configured by the user through *Frontend*).

Backward and Forward Tracking. Based on the causal relationships included in data provenance graphs, the *Forensic Analyzer* conducts backward and forward tracking starting from any node to explain a symptom or ascertain its influence. For example, it is found that the root cause of an unexpected turned-on light is the voice assistant misunderstanding the user's voice command. As mentioned in the previous example, the turned-on light causes a change in the state of the illumination sensor, which in turn causes the cloud server to deem the user's existence. Through backward and forward tracking, the *Forensic Analyzer* provides a global view of the entire ecosystem to understand how it works, and investigators can view this from the *Frontend*.

Security Enhancement. Data provenance graphs contain the runtime status of the smart speaker ecosystem, and the *Forensic Analyzer* continuously checks the status according to the security policies. Security policies determine the valid workflow and the data flow of the ecosystem. For example, the cloud server should not directly control the light without the user's permission, and no sensitive data can be transmitted secretly. The policies also specify the thresholds of the component states. For instance, the house's temperature should be within an appropriate range. Thus, the *Forensic Analyzer* can verify the accuracy of the commands, detect device states, discover data leakage, and investigate the signs of intrusions. In case of any suspicious security indicators, a notification is sent to the *Frontend* with an alarm.

VI. Conclusion

We first discussed the digital forensic characteristics of the smart speaker ecosystem and then proposed a modular digital forensic tool by focusing on the client-side. The tool extracts forensic data from the console and the network traffic. It then encapsulates these data into data provenance graphs, which are used for causal analysis to explain the nature of an observable symptom. Finally, we verified the tool on an off-the-shelf smart speaker product and obtained some interesting findings.

References

[1] N. Malkin, J. Deatrick, A. Tong, P. Wijesekera, S. Egelman, and D. Wagner, "Privacy attitudes of smart speaker users," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 250–271, 2019.

[2] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening? privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–31, 2018.

[3] M. Shen, Y. Deng, L. Zhu, X. Du, and N. Guizani, "Privacy-preserving image retrieval for medical iot systems: A blockchain-based approach," *IEEE Network*, vol. 33, no. 5, pp. 27–33, 2019.

[4] W. Jo, Y. Shin, H. Kim, D. Yoo, D. Kim, C. Kang, J. Jin, J. Oh, B. Na, and T. Shon, "Digital forensic practices and methodologies for ai speaker ecosystems," *Digital Investigation*, vol. 29, pp. S80–S93, 2019.

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

[6] Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, "Prometheus: Privacy-aware data retrieval on hybrid cloud," in *IEEE INFOCOM*, 2013.

[7] M. Merrill, "An uneasy love triangle between alexa, your personal life, and data security: Exploring privacy in the digital new age," *Mercer Law Review*, vol. 71, no. 2, p. 7, 2020.

[8] L. Lin, X. Liu, X. Fu, B. Luo, X. Du, and M. Guizani, "A non-intrusive method for smart speaker forensics," in *IEEE International Conference on Communications*, 2021.

[9] Y. Xiao, H.-H. Chen, X. Du, and M. Guizani, "Stream-based cipher feedback mode in wireless error channel," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 622–626, 2009.

[10] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, and X. Du, "Achieving efficient detection against false data injection attacks in smart grid," *IEEE Access*, vol. 5, pp. 13 787–13 798, 2017.

[11] T. Pasquier, X. Han, T. Moyer, A. Bates, O. Hermant, D. Eyers, J. Bacon, and M. Seltzer, "Runtime analysis of whole-system provenance," in *ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[12] Y. Duan, X. Fu, B. Luo, Z. Wang, J. Shi, and X. Du, "Detective: Automatically identify and analyze malware processes in forensic scenarios via dlls," in *IEEE International Conference on Communications*, 2015.

[13] Q. Wang, W. U. Hassan, A. Bates, and C. Gunter, "Fear and logging in the internet of things," in *Network and Distributed Systems Symposium*, 2018.

[14] M. S. Aktas and M. Astekin, "Provenance aware run-time verification of things for self-healing internet of things applications," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 3, p. e4263, 2019.

[15] W. Ding and H. Hu, "On the safety of iot device physical interaction control," in *ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[16] I. Clinton, L. Cook, and S. Banik, "A survey of various methods for analyzing the amazon echo," *The Citadel, The Military College of South Carolina*, 2016.

[17] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers *et al.*, "The open provenance model core specification (v1. 1)," *Future generation computer systems*, vol. 27, no. 6, pp. 743–756, 2011.

[18] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," *arXiv preprint arXiv:1710.05703*, 2017.