D. Pawlaszczyk, J. Friese, C. Hummert, ""Alexa, tell me ..." – A forensic examination of the Amazon Echo Dot 3 rd Generation", International Journal of Computer Sciences and Engineer...

Article *in* INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING · November 2019

DOI: 10.26438/ijcse/v7i11.2029

CITATIONS
5

READS
2,062

2 authors:

Dirk Pawlaszczyk
Hochschule Mittweida
36 PUBLICATIONS **157** CITATIONS
SEE PROFILE

Christian Hummert
Agentur für Innovation in der Cybersicherheit
27 PUBLICATIONS **154** CITATIONS
SEE PROFILE

IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-11, Nov 2019 E-ISSN: 2347-2693

"Alexa, tell me …" - A forensic examination of the Amazon Echo Dot 3 rd Generation

D. Pawlaszczyk[1*], J. Friese[2], C. Hummert[3]

[1,2]Dept. of Computer Science, Hochschule Mittweida – University of Applied Sciences, Mittweida, Germany
[3]Zentrale Stelle für Informationstechnik im Sicherheitsbereich ZITiS, Munich, Germany

[*]*Corresponding Author: pawlaszc@hs-mittweida.de, Tel.: +49-3727-581218*

DOI: https://doi.org/10.26438/ijcse/v7i11.2029 | Available online at: www.ijcseonline.org

Accepted: 23/Nov/2019, Published: 30/Nov/2019

***Abstract*—** In the recent past, smart speakers and personal assistants like *Alexa* have been used as evidence in different criminal cases. With more than 6 million *Echo* devices in 2019 worldwide, there is a good chance for the constantly active devices to record criminal behavior. That is reason enough to take a closer look at the subject. This article discusses different starting points for a forensic investigation of the latest generation of the *Echo Dot* smart speaker. Therefore, the paper examines three possible aspects for the forensic analysis of the *Echo 3* the latest generation of this device. In hardware analysis, the focus is not only on the general examination of the extracted data but also on validating the existing methods for data extraction. A second focus is the cloud-side analysis. Since this type of analysis requires the knowledge of credential data, a client-side analysis will take place too. The last is done because the use of at least one Companion Client is necessary to operate and control *Alexa*-capable devices. The main findings of this study are presented in this article.

***Keywords*—**Amazon Echo 3, digital forensics, hardware analysis, IoT, IPA

I. INTRODUCTION

In the recent past, smart speakers and personal assistants like *Alexa* have been used as evidence in different criminal cases. In the United States of America, these assistants have already contributed to the clarification of cases. In 2015 Arkansas state police wanted to solve a murder case using the records of an *Amazon Echo*. It has been reported several times that *Amazon* refused to give law enforcement agencies access to their servers before the suspect himself decided to provide the data [1]. With more than 6 million *Echo* devices in 2019 worldwide, there is a good chance for the continuously active devices to record criminal behavior. That is reason enough to take a closer look at the subject.

The research subject in this article is the *Intelligent Personal Assistants (IPA) Alexa* used on the *Echo Dot 3nd generation*. The IPA examination is adjusted based on the multi-level forensic approach. The results presented in this article are mainly of interest to forensic examiners who inquire data on IPA for evidence. Before we start with the examination of the smart speaker, we first want to light up the technical background and the experimental design we used.

Smart Speakers are loudspeakers connected to the Internet, which are characterized by the use of IPA with simplified control [2]. They allow the execution of automated processes, control by voice commands, and above all, the operation of devices without any contact [3]. Besides the use of music streaming services, news services, podcasts, and other audio services from the Internet, the use of the Smart Home is also available.

The 3rd generation *Amazon Echo Dot* is such a smart speaker. It is a 2-way loud-speaker from the range of Amazon's Echo devices and has been on sale since late 2018. It integrates the functions of the IPA *Alexa* and thus enables voice control and recognition. The basis of the use is a constant Power supply and WIFI connection for link to *Alexa Voice Service (AVS)*. The *Echo dot* thus corresponds to a kind of gateway for the voice commands to the AVS cloud. It itself is not designed to handle, save, or process a large amount of data locally. The *Echo Dot v3* is characterized by its small and compact rounded shape. The four function keys are located on the upper side and include volume control, muting of the microphone, and the activation key. The latter replaces the need to use a signal word to activate the *Echo Dot*.

Figure 1. The Echo Dot 3. Generation

The device itself contains four integrated microphones and a loudspeaker. The Core is a quad-core MediaTek MT8516BAAA 64-bit ARM Cortex-A35 processor, which resides on the same board as the memory chip. The memory chip is an 8GB *Samsung* KMFN60012M-B214, an MCP multi-chip module combining an eMMC with DRAM, on which the operating system resides. The eMMC comes up with a flash memory as a System-in-Package (SIP) solution. The MCP is de-signed as BGA package.

The device is permanently in standby mode, i.e., the language and sounds of the environment are first processed internally by the device. The actual voice control is initiated by spoken keywords, such as *Alexa*, *Echo*, *Amazon* or *Computer*, or by pressing the action button on the top side of the device. The device then digitally transmits the words spoken in the room to the manufacturer via the internet, where an attempt is made to implement the commands. For this purpose, small sections of the ambient noises are recorded and directly deleted or overwritten, if the signal word is not recognizable. According to *Amazon*, speech recognition is not yet running in this phase, only the acoustic sound of the signal word. Therefore, no transmission of data takes place to the Cloud [4]. Following, the user receives an answer regarding his command. The communication ends, when the user directly terminates the interactions via voice command or additional communication that is not addressed to the IPA [5].

The *Echo Dot* signalizes the individual phases between readiness, speech recognition, and speech processing with different colors on his light ring. If several devices of the *Echo* series are in the vicinity, speech recognition is always carried out on the device closest to the user (*Echo* spatial perception). The voice commands are recorded using several integrated microphones, whereby audio recordings from relatively big removal and with higher noise levels are generated and effectively monitored on the processing servers. Although the signal word only triggers the IPA function, the microphone function is active permanently [6][7].
Within the *Amazon Alexa echo* system, various components play an essential role [4][8]. The central element is *Alexa* because all components can interact with her via a network (see Figure 2). *Alexa* enables the use of the cloud services offered via *Amazon's Web Service* (AWS). The services provided by *Alexa* include authentication, data management, and logging. The *Alexa*-enabled devices, for example, the Echo Dot, are essential in this system to talk to the *Alexa* Cloud Service [5],[6].

The control and administration of the system take place via the Companion Clients (laptop with a web browser, mobile devices with the Amazon *Alexa* App) through direct access to the cloud server. Companion Clients are personal devices of a user. They primarily serve to execute the apps belonging to Alexa. On the PC, access to the cloud is done via a web browser.

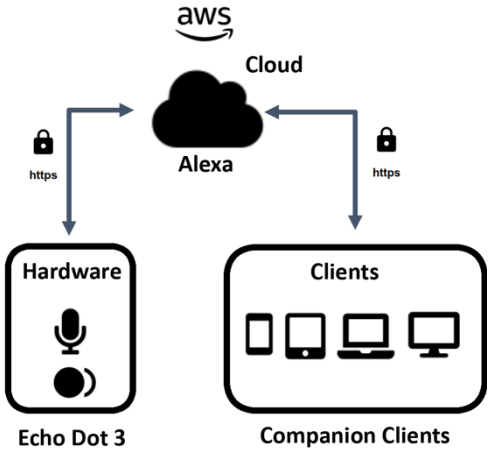

Figure 2. The Amazon Echo Ecosystem

II. RELATED WORK

Based on the above, we have four starting points for a forensic investigation. First, you can analyze the network traffic between the echo and the cloud [6],[7][8]. A permanent network connection is one of the basic pre-requisites for *Echo* devices, because the *Alexa*-enabled devices and the *Companion Clients* communicate with *Alexa* over the internet. Much of this data traffic, which may contain relevant artifacts, is transmitted over an encrypted connection, after starting a session with a valid *userID* and *password*. The SSL encryption must be broken for this. However, this is not part of the present contribution. For a more detailed evaluation of the network data, please refer to an analysis by the *Ford and Palmer* [8]. Another contribution dealing with monitoring function of the *Amazon Echo* can be found in [9]. They perform a hardware analysis and examined in their work an *Amazon Echo* device of the first generation at the hardware level. In [7],[10],[11] different potential methods for hardware-side analysis of the *Amazon Echo* device are presented. In addition to using the Developer and Debug Ports to dump the firmware, knowledge of a JTAG pinout can also be used to access the firmware without destroying the device. The authors describe some possible approaches for data extraction through eMMC root, JTAG or debug ports. However, they do not go into detail regarding the data stored on the memory. In [6] also published a pinout for the ISP approach for the *Echo Dot v2*, so that it can be used to obtain a firmware dump using a non-destructive method. This would allow analysis of the hardware of the Echo device.

With each new generation of devices, *Amazon* apparently changes the structure and hardware design as well. Thus, the results are not simply transferable and require at least a

review. Another potential access option is the hidden USB interface of the *Echo 3*. Although the external USB port has been removed from the newer device, access via the PCB is still possible. This way some researchers could at least read the encrypted bootloader [7],[8].

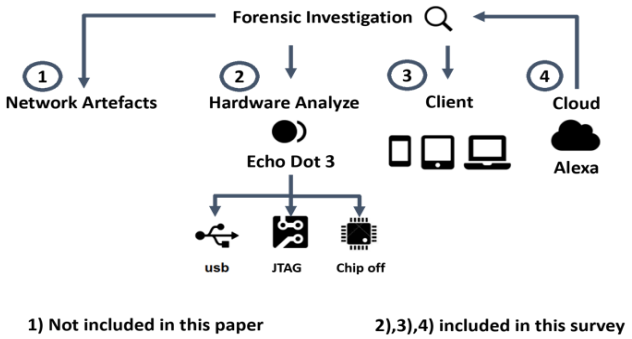

Figure 3. Possible linchpins for a forensic investigation

Last but not least we can take a closer look at the cloud. The *Alexa cloud system*, like other cloud-based services, communicates with a list of predefined APIs. Even though this list is not public, *Chung* et al. [6] tried to find and publish part of them. Most of the publications focus on the 2nd generation *Amazon Echo Dot*. In addition, there are a number of publications that shed general light on the forensic investigation of smart home systems [16],[17],[18]. For the latest generation of Amazon's smart speaker, however, there are few publications yet. This article attempts to close this gap.

III. METHODOLOGY

Our IPA examination is adjusted based on the multi-level forensic approach. Therefore, this contribution examines three possible aspects for the forensic analysis of smart speakers. In hardware analysis, the focus is not only on the general examination of the extracted data, but also on validating the existing method for data extraction or finding an alternative. A second focus is the cloud-side analysis. This is based on the use of predefined APIs to send and receive data. However, the list of available APIs is not public. Based on the list provided by previous analyses uncovered unofficial APIs used by *Alexa* for transporting data cloud-native artifacts should be preserved for further investigation. Since this type of analysis requires the knowledge of certain data, a client-side analysis will take place. This is because the use of at least one Companion Client is basic to operate and control *Alexa*-capable devices.

Table 1. Components of the test system

Object	Description
Alexa enabled devices	Amazon Echo 3. Generation (DSN:G090VC0991650XXX) Amazon Echo 3. Generation (DSN:G090U50991854XXX)
Companion clients and applications	ZTE BLADE L110 (Android 5.1) +Amazon, Alexa App (2.2.280247.0) Google Pixel 3A (Android 9) + Amazon Alexa App (2.2.291588.0) HP Laptop 15-da0xxx (Windows 10 Pro) + Firefox (Quantum 68.0.2)

The configuration of the settings, the tracing of past conversations with Alexa, as well as activating and deactivating skills can be done using the app or the Web Browsers. The combination of the acquired data from the client and cloud is therefore fundamental for a good and comprehensive analysis. For the forensic investigation a test environment with two *Amazon Echo 3* and three Companion Clients was built. The latter consisted of two smartphones, both installed with Alexa app and a laptop (see Table 1). The method we have chosen to study the device is divided into the following three steps (see Figure 3):

1. **Hardware Analyse**: We first try to get access to the memory chip. For this, we apply methods already used in earlier Echo Dot versions like eMMC root, JTAG, and Debug ports and check their applicability. We try to find information about configuration and user data artefacts, which are saved directly on the device. This step involves identifying and examine the partition and file systems of the *Echo Dot*.
2. Analysis at the **Client level.** We try to recover device ids, user information as well as token information of the user that was signed onto Android and on the desktop system. We try to find the position of artefacts like *to-do* lists and *shopping lists* on all Companion clients.
3. In the third step, our analysis involves creating a web session with *Alexa* using credentials like *userID* and *password* to acquire data from the **Cloud**. This part involves gathering information about registered user accounts, saved Wi-Fi settings, *Google* calendars linked to *Alexa*, and voice recordings from last commands spoken.

The procedure chosen by us ensures that most essential forensic aspects are taken into account. In this way, we get a comprehensive forensic picture of the Echo Dot 3rd Generation. A recording and analysis of the data traffic is omitted here since the forensic value of this investigation is somewhat limited.

IV. RESULTS AND DISCUSSION

In the following, the three different ways for data extraction, their implementation and the forensic results are presented.

A. Hardware Analysis

In order to perform a hardware analysis, direct access to the memory chip is re-quired. To access the memory chip, we

first have to open the case of the *Echo Dot*. Therefore, the glued cover on the underside of the device has to be removed using a scalpel (see Figure 4). Then the screws must be loosened. Next to the loud-speaker is the board on which the CPU and the memory chip reside (see Figure 5). The board can also be unscrewed (see Figure 6).

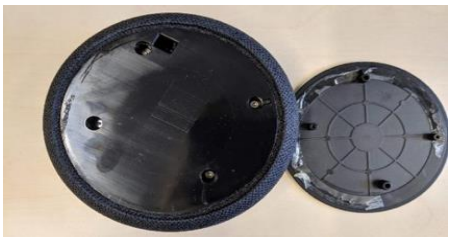

Figure 4. Removing the glued cover

Figure 5. Loudspeaker and circuit board after removal of the covers

Figure 6. The main board

The configuration of the settings, the tracing of past conversations with *Alexa*, as well as activating and deactivating skills can be done using the app or the Web Browsers. The combination of the acquired data from the client and cloud is therefore fundamental for a good and comprehensive analysis. For the forensic investigation a test environment with two *Amazon Echo 3* and three Companion Clients was built. The latter consisted of two smartphones, both installed with *Alexa* app and a laptop (see Table 1). When looking at the circuit board, it became evident that for the construction of the *Echo dot v3*, changes have been made compared to the *dot v2*. Instead of a single, now there are two circuit boards installed. To access data over ISP, pin layout would have to be known. However, by now, this is not documented to the public. Even the USB interface determined did not provide much information. The only finding was the presence of an encrypted boot loader. Since no JTAG interface is available, instead of the planned ISP method, the destructive method of chip-off is considered alternatively.

Chip-Off is a method of data extraction in which relevant ICs are physically extracted from the device board, in order to be able to connect them directly via a standardized interface [8],[9]. The eMMC memory chip is desoldered from the printed circuit board and then bridged, to get access to the data. There are two different types of chip-off [14][17]:

- *Cold chip-off*: A particular grinding device is used to grind the circuit board below the memory chip. The milling cutter used must be tiny. This method requires high accuracy to prevent causing damage to the chip.
- *Warm chip-off*: For this, the component is desoldered. The predominant danger represents a possible excessive heat supply, which could result into the destruction of the memory chip. Furthermore, the solder pads may be separated from the IC by levering out the chip.

For preliminary analysis, a collection of information was carried out on the built-in chip *Samsung* KMFN60012M-B214 MCP, including a test of the contact points. The circuit board, including the chip, was examined to analyze whether the chip is soldered or additionally glued. The latter makes it more difficult to remove the chip and it was initially designed to ensure that neither water nor dust gets in between the chip and the circuit board. The desoldering should have done with the aid of hot air. The target temperature for the process is 250°C to hit the memory chip. The temperature set at the device was 335°C is significantly higher, simply because the measure–ment is made inside the tip of the device, and the air cools down immediately after leaving.

In the next step, we had to loosen the surrounding metal frames of the memory chip. Otherwise, it might absorb too much heat. The building block was heated evenly to prevent deformation. A scalpel was also used to loosen the brick, as this could be done without much force. We have to be careful, since components pre-sent in the memory chip could be destroyed, and the data would be lost. Also, the contact points of the chip could be damaged. The retention of the solder pads of the MCP device is essential in this method in order to be able to read out the data.

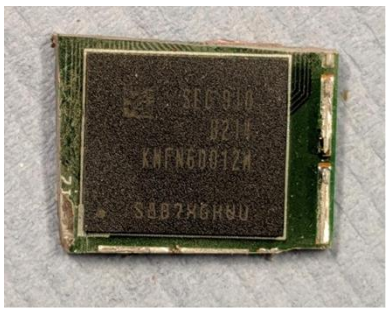

Figure 7. Memory chip after PCB milled

During the implementation, it became clear that the memory chip is additionally glued. Furthermore, the relatively thick printed circuit board posed a problem, as too much heat was generated by it. It was necessary to replace it. Thus, the majority of the circuit board around the memory chip gets replaced with a *Dremel*.

Afterward, the chip was soldered out with hot air (365°C). It turned out that, for the industrial solder used, significantly higher temperatures are required. Even more, the chip was glued in the middle. After successful loosening the circuit, it was cleaned using a cleaning blade. Under the microscope, the chip could be identified as an MCU with an integrated eMMC as 221 BGA. This model provides a description of the connectors on the back panel of the chip, which are small solder beads standing next to each other in a grid. Following the analysis of the chip, the BGA soldering points were re-tin-plated.

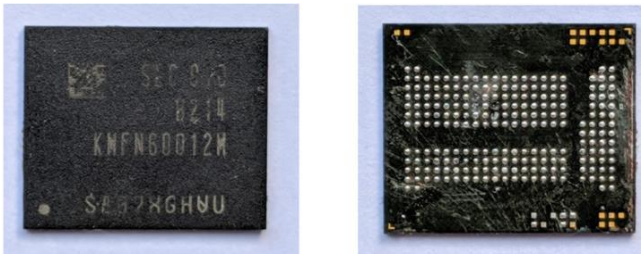

Figure 8. Front (left) and back (right) sides of memory chip

To be able to create an image in the last step and read out the memory chip, a suitable adapter is required. Therefore, the chip size was measured. An adapter of the company *Rusolut* was used. An adapter of the right size is crucial so that the con-tact points are correctly positioned and the memory chip can be recognized.

The *Access Data FTK* imager was used to create a raw image of the memory chip. Since it was not known, which operating system or file system is available, the first Investigation of the image was done using the *Visual Nand Reconstructor (VNR)* software from *Rusolut*. In doing so, it was possible to process the raw data of the memory chip, even if the memory chip has been potentially damaged during chip-off to store specific data to save the world.

Property	Value
Phone (4)	
Phone Vendor	Amazon
Phone Model	AEODN
User Device Name	donut
Serial Number	G090U50991854P3J
OS (1)	
Phone OS	Android
Location (1)	
Time Zone	Europe/Berlin
Wi-Fi (2)	
Hotspot AP Name	AndroidAP4P3J
Hotspot Password	[redacted]

Figure 9. Device Information from Android template (VNR)

Since not necessary in this case, the first overview has been made regarding the image with *XWays Forensics*. Due to hints to an Android system, later on, the better preparation of certain data types and integrated viewer, the analysis could be performed with the *UFED Physical Analyzer*. The memory chip itself provided 7GB of the specified 8GB as storage space, which also corresponds to the size of the created image. The first time the image was viewed using VNR, 16 partitions were detected, and the partitions 01 to 08 were not readable. However, there can be less than 16 partitions if some of these partitions belong together. An operating system was not directly identifiable, and only a few partitions (09, 13-16) had a recognizable file system, in that case, *ext4*.

Partitionierungstyp: GPT

Name	Beschreibung	Typ	Größe
Aus Sektoren ausgegliedert	Verzeichnis, virtuell (für Un...		0 B
Anfang der Platte	Datei, virtuell (für Untersuc...		1,0 MB
Partition 01 [kb]	Partition, existierend	?	1,0 MB
Partition 02 [dkb]	Partition, existierend	?	1,0 MB
Partition 03 [lk_a]	Partition, existierend	?	1,0 MB
Partition 04 [tee1]	Partition, existierend	?	5,0 MB
Partition 05 [lk_b]	Partition, existierend	?	1,0 MB
Partition 06 [tee2]	Partition, existierend	?	5,0 MB
Partition 07 [expdb]	Partition, existierend	?	10,0 MB
Partition 08 [misc]	Partition, existierend	?	512 KB
Partition 09 [persist]	Partition, existierend	Ext4	16,0 MB
Partition 10 [boot_a]	Partition, existierend	?	16,0 MB
Partition 11 [boot_b]	Partition, existierend	?	16,0 MB
Partition 12 [swdl]	Partition, existierend	?	32,0 MB
Partition 13 [system_a]	Partition, existierend	Ext4	1,0 GB
Partition 14 [system_b]	Partition, existierend	Ext4	1,0 GB
Partition 15 [cache]	Partition, existierend	Ext4	32,0 MB
Partition 16 [userdata]	Partition, existierend	Ext4	5,1 GB
Partitionslücke	Datei, virtuell (für Untersuc...		16,5 KB

Figure 10. View of the Echo Dot 3 Image in XWays Forensics

For a more detailed analysis, only partition 16 was considered, as it is called *user_data/data* and thus represents the data partition. Furthermore, it is the largest partition, holding 5.1 GB. Because the partition structure was unknown, an analysis of all files and directories was done. Looking at the directories, many were empty or contained only a few files. For the first analysis, every single file of partition 16 was viewed and interpreted using a hex-editor. It turned out that mainly files from the *shared_prefs* subfolders contained specific information. Among other things, there was the XML-based file *network_deregistration.handler.xml*. This file contains the SSID of the router and the encryption type of the network. Furthermore, the *userID* and various *Unix* timestamps could be found.

The hardware analysis did provide not only the user-specific *userID*, but also general information about the last registered user and other registered users, accounts, and devices, as well as the wireless network in which the device was last connect-ed. Whether the echo dot was registered for a specific time in that network, cannot be reconstructed, since no geo-coordinates or time stamps are available. Using the extracted Unix timestamps, it is possible to set up the dots, but also the

time of the last registration of the known user to comprehend. The device thus primarily provides general information that forms the basis for further aspects of analysis. Furthermore, it enables a unique physical assignment of the device using the data of the *Amazon* account. In future work, a non-destructive method should be used for successful hardware analysis. The creation of an ISP pinout based on the Chip-Off Methodology is a first step. Furthermore, it should be considered to try the buffer memory on the device, which contains the ambient noises while waiting for the activation word to be heard and written. The problem is the volatility of the data. It is whether known how large this buffer nor how quickly the data is overwritten.

B. Client-side analysis

The client-side analysis describes the extraction of the *Amazon Alexa* app and its data from the *ZTE Blade L110* mobile device. There are two methods to extract the data from a mobile device:

- The *logical backup*: Only the described partitions are backed up. Deleted or hidden files are not backed up and can, therefore, not be considered later in the analysis.
- The *physical backup*: All partitions are backed up, including the blank ones and the root area. Thus, also already deleted and hidden files, which can contain potentially relevant information, are considered.

The backup and extraction of the app data was done with the licensed software *UFED* from the company *Cellebrite*. Since it is not known where the app stores data and what kind of data is stored at all, a physical backup was performed. For saving the device, several methods are available. The choice of methods differs, depending on the chipset or firmware installed on the smartphone. However, the image should be the same in all cases.

The *ZTE Blade L110* is equipped with a *Spreadtrum* chip. The access to this device was done via the Physical Boot Loader. In fact, a weakness in the boot loader was used. If the smartphone was connected to the laptop according to the given methodology - the device switched off by pressing the magic keys, in this case, the volume keys - the dumping process starts. The backup included the individual partitions (system, cache, user data). After the backup was performed and the image was analyzed, it became apparent that the backed-up user data partition did not contain any data.

The image is viewed and evaluated in the *UFED Physical Analyzer*, also provided by *Cellebrite*. In addition to a complete overview of the extraction, the program also prepares files of any type, and it is possible to view databases or text files directly. In this case, the preceding filtering option to *amazon* is especially helpful in order to reduce the number of files. In addition to identified user accounts, device users, passwords, and wireless networks could be found. The client-side analysis focuses on the *Amazon Alexa* app. The app data can be found under *\data\com.amazon.dee.app* in the image of the mobile device on the Userdata partition. The folder on the system tested is about 33MB in size.

The first thing that was noticeable when looking at the files in the different folders was the number of files marked with *.DELETED* that no longer exists. Furthermore, the files to be found are mostly XML files or databases in *SQLite3* format. In addition to the databases of the app, the *shared_prefs* folder contains relevant files. This folder contains files whose data have to be entered or defined once by the user. This data will be used again and again in the future so that reentering is not necessary when restarting the application. The data is stored in the files as a key-value pair.

File *account_change_observer.xml* can also be found on the image of the Echo Dots and also contains the *userID* of the *last_seen_account.*

The file *com.amazon.dee.app_preferences.xml* provides information about the time of the last successful synchronization, i.e., the last use of the app. The timestamp is displayed in *Unix* format. This information can be especially important. Only if the app has been successfully synchronized to the current state it will contain all information. Information about the user, in addition to the *userID*, was available for the first time in the file *service.identity.xml*. The user name and e-mail address used to register for the *Amazon* account are stored here (see Figure 12). The latter, in particular, plays a fundamental role in the successful performance of the cloud analysis.

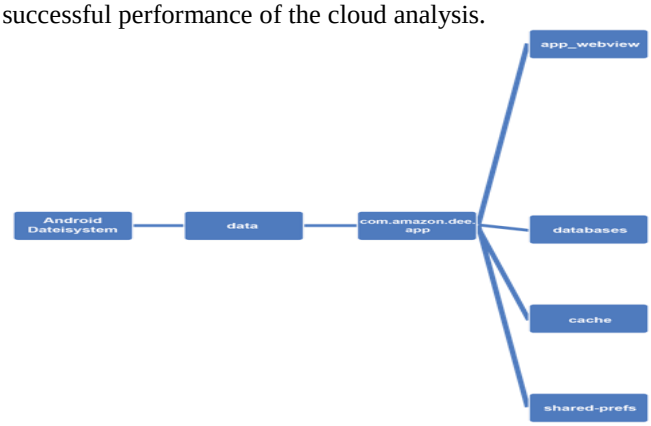

Figure 11. Schematic structure of the Alexa app-folder on Android

<string name="user.name">[redacted]</string>

<string name="user.[version]">5</string>

<string name="user.tokens">{"comms-has-shown-oobe":"true"}</string>

<string name="user.email">[redacted]</string>

<string name="user.profile.lastName">[redacted]</string>

<string name="user.originalMarketplace">[redacted]</string>

<string name="user.profile.firstName">[redacted]</string>

Figure 12. Extracted Data from file service.identity.xml

Also interesting is the database *RKStorage* with the table *catalystLocalStorage* The key *@ElementsHomefedd:carJson* within this table contains several *CARDS* which contain a large variety of forensic artifacts, such as the transcript of what the *Echo* understood in the given voice commands and its respective response to them (see Figure 13). But also, the URL to the location of the audio recordings on the *Amazon* server. Basically, these cards are in JSON format.

Table 2. Sample data from a card in the *RKStorage* database (JSON)

Audio action transcript from *RKStorage*
"cardMetricAttributes":null, "cardType":"TextCard", "**creationTimestamp**":1566677399416, "playbackAudioAction": "actionType":"PlayAudioAction", "**mainText**":" Alexa heard the following: „ Alexa, what new soccer games are there?“", "subText":null, "subTextRoute":null, "**url**":"/api/utterance/audio/data?id=A3MUN6DTXA:1.0/2019/08/24/20/ "primaryActions":[], "prompt": "ssml":"<speak><prosody volume=\"x-loud\"> *That could answer your question: I know three last football matches by starting date, around 24 August 2019 at* <say-as interpret-as='time'>22:00 o'clock </sayas>. </prosody><metadata><promptMetadata> <promptId>AnswerSsml</promptId> <namespace>SmartDJ.MusicQA</namespace><locale>de_DE</locale><overrideId>default</overrideId> <variant>809dfcd2-2807-4eaf-93e9-1130c2db01fa</variant> <stageVersion>Adm-20160403_211532-21</stageVersion> <promptData><replyType/>…<promptID>AnswerSsml</promptID> <namespace>SmartDJ. MusicQA</namespace> <overrideId/><prosodyPreRenderHook/><expectations/> </promptData></promptMetadata></metadata></speak>","ttsUrl":null}, …"registeredCustomerId":"A38IYSD2TKXXXX", "secondaryActions":null, "**sourceDevice**": "serialNumber":"G090VC09916555555"…

In the following, individual parts of the cards with relevant information are highlighted:

- *creationTimeStamp*: The time zone of the timestamp corresponds to the time zone of the wireless network to which the echo dot was connected when the voice command was given. The timestamp itself is in *Unix* format (1566677399416 = 24.08.2019, 22:09:59). This is generated independently of when the *Card* itself is created on the mobile device. If the device is not active or not connected to the internet at the time the *Card* is created, it will be created once there is an active connection.
- *maintext*: This field represents the result of the text-to-speech conversion, i.e., it is the transcript of what the *Echo* device understood at the given voice command.
- *sourceDevice*: This is the DSN identifying the *Echo* device that accepted the voice command. Thus, the device can be physically assigned to the user.
- *url*: Specifies the web address that can be used to track the location of the corresponding audio recording on the *Amazon* server. Navigation to this URL is possible if the login data for the associated *Amazon* account is known.

C. Cloud-side analysis

With known credentials (user id and password), access can be gained via *alexa.amazon.de*. We could find artifacts of different lists like to-do's and shopping lists, which the user has created in the past with *Alexa*. The lists are generated independently by the individual devices. Alarms and timers are also displayed. When viewing the configured timers and alarms, a distinction is made between the individual registered *Echo* devices. With access to *amazon.de/alexaprivacy*, we can get an overview of all audio recordings (see Figure 14). That includes voice commands stored in the user account, except if they have deleted them manually.

In the process, we can find not only every voice command from the user as an audio recording, but we can also listen to it. For every recording, the exact time and date, as well as which device is using the voice command, can be accessed (see Figure 15). The voice command given by the user will also be transcribed for a better understanding of what *Alexa* understood. The answer or reaction of *Alexa* is listed in text form under the transcript of the audio recording and should give the user the possibility to manually evaluate afterward whether the reaction of *Alexa* to the command given was helpful. These usually did not contain any conversation parts, i.e., only silence or parts of private ones, not to *Alexa* directed conversations. Why *Alexa* was activated in these cases is unclear. Many phonetically sounding words are possible, among other things. Even though these commands were not addressed to *Alexa*, they are stored in the user's account and marked as incorrect. The quality of the voice

recordings will vary, probably depending on the volume and distance of the speaker from the Echo device. Often there is also background noise/ -recorded during the analysis of the audio recordings. The last point can play a potentially significant role in criminal investigations.

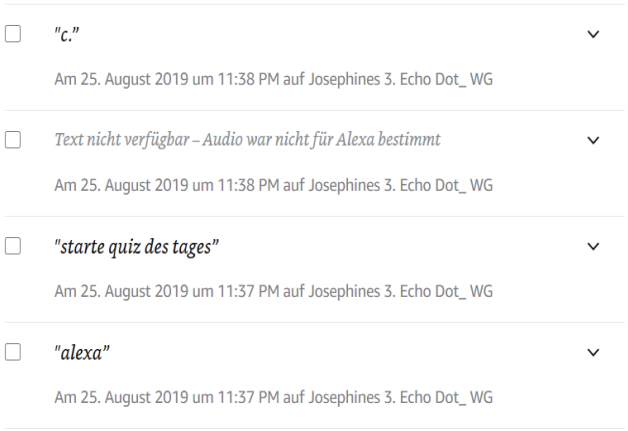

Figure 13. Overview of the voice recordings within the cloud

If we follow the link *contacts* in the *Amazon* online account, we will get various information about the owner. The data will be presented in JSON format as with the other APIs. Despite the full name, the protocol does not contain any personal information. The last finding is an interesting fact, as not even the email address and address, which are in any case stored in the *Amazon* account, are displayed. Furthermore, when using the *Amazon Alexa app* with *Google Pixel 3A*, the mobile phone number is saved, and the contacts are synchronized, too. However, the mobile phone number was never stored directly in the *Amazon* account and is, therefore, missing here.

When navigating to the device URL in the online account, using the previous login, we get information about all devices that are linked with the account. In addition to the user-configured device name, information about the state of the *Echo Dots* is provided. Information that is always available, regardless of the status of the device, is the ID of the device types, and the DSN. (see Figure 16). Furthermore, the capabilities available for the particular device or skills are generally activated via the *Amazon* account and general setting options, such as *Amazon Music* or *Spotify*. These represent options that enable the functionality and control of the device.

The cloud analysis allows us to collect user-specific data as well as device-specific data and to obtain network-specific information. Especially the device information is helpful in order to be able to assign a device uniquely to the user, because the DSN and user information are stored together. However, the most important are the audio recordings stored in the user's account. These can provide information about an exact period and about actions performed at that time. It is most likely possible to receive pieces of information about people present, based on an analysis of background noise.

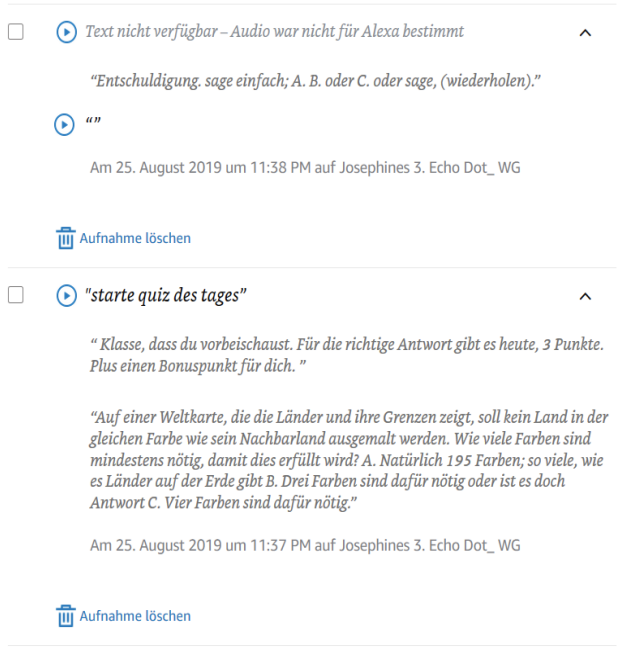

Figure 13. detailed view on the voice recordings (example)

Figure 14. Artefact: The device URL

V. CONCLUSION AND FUTURE SCOPE

Our research focused on identifying and investigating as many different possible forensic artifacts as we could about the *Echo Dot 3*. In general, the way the device and its components are used is decisive for the traces we can find. Similar to the multitude of possible applications of the devices, there are also different aspects of the analysis. As in the sequence of the analyses, the first point of view should be the hardware-side analysis.

Since there is still no official interface available to make a Dump, the only way to access the data is to use destructive methods. In the context of this analysis it was determined that none of the previously known non-destructive methods used to create a firmware dump from earlier Echo devices have been applied to the *Echo Dot* of 3rd generation were transferable. Therefore, the only possible way was a chip-off that delivered first-time information about the operating system and file system running on the Echo Dot. The hardware analysis provided not only the user-specific *userID* but also general information about the last logged user and other registered accounts and devices, as well as the wireless network in which the device was last installed. Whether the echo dot was registered for a specific time in this network cannot be reconstructed, because no geo-coordinates or timestamps are generated. Using the extracted Unix timestamps, it is possible to trace the setup of the dots in particular, but also the time of the last login of the known user.

A client-side analysis of the data contained in the app is only possible regarding the forensic aspect, if a physical backup of the mobile device can be performed. Based on the knowledge of the app analysis from other work [5], a file named *sound.wav* was identified in the cache folder of the application during the client-side analysis. This file can be found there, if one of the previous voice commands was selected and listened to via the activity tab. This activity downloads the file off the Amazon server to the device and plays it. This is the only moment so far in which an audio file can be played back. It is verifiably physically stored on a device. The analysis of the app data of the image showed no existence of such a file. The client-side analysis further showed that, even without direct access to the Amazon account, information about the last given voice commands could be deter-mined. The basis for this is the *RKStorage* database, which contains forensic artifacts in the form of so-called cards.

One aspect, that still needs to be investigated, is whether the removal of the Audio recordings, the corresponding *Cards* in the app, will also be deleted. Those, as mentioned earlier, could be a possibility to display information about manually deleted commands and their potentially relevant content. To sum it up, by far the most data can be obtained through a cloud side analysis. For these, however, knowledge of the credentials for the AVE cloud account is necessary. An analysis of the app data can provide valuable information, too.

REFERENCES

[1] L. French, *"Virtual Case Notes: Not Only Can Alexa Eavesdrop - She Can Also Testify Against You"*, Forensic Magazine, May 10, pp.**1-3**, **2019**.

[2] S. Yadav, S. Srivastava, M. Singhal, "*Smart Home Automation Using Voice Recognition*", International Journal of Computer Sciences and Engineering (IJCSE), Vol.**7**, Issue.**2**, pp.**560-563**, **2019**.

[3] P. Naik, N. Telkar, A. Patil, *"Smart Home Automation Technique using IoT"* , International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), Vol.**2**, Issue.**3**, pp.**2456-3307**, **2017.**

[4] M. Chung, H. Iorga, J. Voas, *"Alexa, can i trust you?"* Computer, Vol.**50**, Issue.**9**, pp.**100-104**, **2017**.

[5] G. López, L. Quesada, L.A. Guerrero, *"Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces"*. Springer International, Advances in Human Factors and System Interactions, Advances in Intelligent Systems and Computing Vol. **592**, pp.**241-250**, **2018**.

[6] H. Chung, J. Park, S. Lee, *"Digital forensic approaches for amazon alexa ecosystem"*, In Proceeding of the 17th Annual DFRWS, **USA**, pp.**15-25**, **2017**.

[7] S. Li, K-K. Choo, W. Qindong, J. William, *"IoT Forensics: Amazon Echo as a Use Case"*, **2019.** doi:10.1109/JIOT.2019.2906946

[8] M. Ford and W. Palmer, *"Alexa, are you listening to me? an analysis of alexa voice service network trace"*. Vol.**23** Issue.**1**, pp. **67-79**, **2019**.

[9] B. Copos, K. Levitt, M. Bishop, and J. Rowe, *"Is anybody home? inferring activity from smart home network traffic,"* in Security and Privacy Workshops (SPW), 2016 IEEE, pp.**245–251**, **2016**.

[10] S. Vasile, D. Oswald, T. Chothia, "*Breaking all the things - a systematic survey of firmware extraction techniques for IoT devices*". Smart Card Research and Advanced Applications, Cham, pp.**171-185**, **2019**.

[11] M. Kirmani, M.T. Banday, *"Digital Forensics in the Context of the Internet of Things"*, **2019.** doi: 10.4018/978-1-5225-5742-5.ch011

[12] E. Oriwoh, D. Jazani, G. Epiphaniou, P. Sant, *"Internet of things forensics: Challenges and approaches"*. In Proceedings of the 9th IEEE Int. Conf. Collaborative Computing: Networking, USA, pp.**608-615**, **2013**.

[13] J.R. Shackleton, *"Alexa, amazon assistant or government informant."*, U. Miami Bus. L. Rev., Vol.**27**, Issue.**2**, pp.**301-227**, **2019**.

[14] S. Li, K.R. Choo, Q. Sun, W. Buchanan, J. Cao., *"IoT forensics: Amazon Echo as a use case."* IEEE Internet of Things, Vol.**6**, Issue.**4**, pp.**6487-6497**, **2019**.

[15] A. Nieto, R. Roman, and J. Lopez, *"Digital witness: Safeguarding digital evidence by using secure architectures in personal devices,"* IEEE Network, **Vol.30**, Issue.**6**, pp.**34–41**, **2016**.

[16] Q. Do, B. Martini, K.-K. R. Choo, *"Cyber-physical systems information gathering: A smart home case study,"* Computer Networks, Vol.**138**, pp.**1–12, 2018**.

[17] N.-A. Goudbeek, R.C. Kim-Kwang, *"A forensic investigation framework for smart home environment,"* in In Proceedings of 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2018). IEEE, , p. **1**, **2018**.

[18] D. Mishra, *"Internet of Everything Advancement Study in Data Science and Knowledge Analytic Streams":* International Journal of Scientific Computer Science and Engineering Vol.**6**, Issue.**1**, pp.**30-36**, **2018**.

Authors Profile

Mr. D. Pawlaszczyk pursued a Diploma in Computer Science from the Technical University of Ilmenau in 2000 and a Ph.D. of Science in the year 2009. He is currently working as a full professor in the Department of Computer Sciences, Hochschule Mittweida – University of Applied Sciences. He has published more than 20 research papers in reputed inter-national journals, including Springer and IEEE, and it is also available online. His main research work focuses on Digital Forensics, Network Security, Cloud Security and Privacy, IoT, Distributed Simulation, and Artificial Intelligence. He has eight years of teaching experience and 12 years of Research Experience.

Mrs. J. Friese pursed Bachelor of Science from University of Applied Sciences in Mittweida (Germany) in Digital Forensics, in 2019. She is currently working as an cybercop for the Federal Police of Thuringia, Germany. Her main research work focuses on common and digital forensics, IoT and cyber crime.

Mr C. Hummert is the director of the Digital Forensics section at ZITiS. Before, he was a full professor for IT-Security/Digital Foren-sics at the faculty of Applied Computer and Biosciences of Mittweida University of Applied Sciences. He has graduated in computer sciences at Albert Einstein University in Ulm. After his doctorate at the Friedrich Schiller University in Jena, he worked six years as a forensic expert for the Federal Police of Thuringia. Dr. Hummert was appointed to the Interpol Digital Forensics Expert Group. Several research projects in his group are focusing on digital forensics. He is especially interested in malware analysis and car forensics. Dr. Hummert is involved in the expert training for digital forensics at the German BKA.

