The Journal of Supercomputing
https://doi.org/10.1007/s11227-021-03992-1

# AI-enabled device digital forensics for smart cities

**Sungbum Kim[1] · Wooyeon Jo[2] · Jino Lee[1] · Taeshik Shon[1,3]**

Accepted: 9 July 2021


**Abstract**
Recently, smart cities provide various services to citizens through the convergence of Information and Communications Technology and industries such as transportation, health care, and automobiles. Accordingly, the number of smart devices that use artificial intelligence technology to store the personal information of users to provide services efficiently is increasing. Smart devices can be used to acquire key evidence through digital forensics, which can also serve, as evidence in a court. In this study, we acquire and analyze user data stored in wearable devices by applying a data acquisition framework for smart devices. This study contributes to the acquisition of key evidence for investigations.

**Keywords** Digital forensics · Smart city · IoT · Wearable device

## 1 Introduction

Smart cities provide convenient services to citizens by combining Information and Communications Technology, such as big data and IoT (Internet of Things), with industries such as health care, automobiles, construction, and transportation. As such, various smart devices have been developed and released [1, 2]. In particular, smart devices such as home IoT devices (smart speaker, smart TV, etc.) and wearable IoT devices (smartwatches, smart glasses, etc.) have been in use. Smart devices are generally connected to smartphones to receive personal information from users and provide customized services. These devices store personal information, such as health care and net-browsing, that is used to provide convenient services on a device or the cloud [3]. For example, an AI speaker provides functions such as weather, news, and Internet search through voice, as well as stores the user

✉ Taeshik Shon
tsshon@ajou.ac.kr

[1] Department of AI Convergence Network, Ajou University, Suwon, Korea

[2] Department of Computer Engineering, Ajou University, Suwon, Korea

[3] Department of Cyber Security, Ajou University, Suwon, Korea

Published online: 15 July 2021

location, search history, and third-party application account information. In wearable devices, health-related functions such as heart rate and movement record are provided through a smartphone connection, whereas user location information, heart rate information, and connected smartphone information are stored. In addition, there is personal information used for device setting. The data stored to provide these services can be used as evidence in an investigation from a digital forensic perspective. For example, in 2016, a suspect was arrested in Arkansas by acquiring voice data recorded on an Amazon AI speaker [4]. In addition, data stored on a Fitbit, a smart band, were used as court evidence in a 2015 murder case in Connecticut, USA [5]. Thus, many types of personal information exist in smart devices, and data acquired through a data extraction and analysis of the device can be used as conclusive evidence in an investigation. Smart devices are very small and limited. In addition, since many manufacturers use different types of operating systems, it is difficult to apply traditional forensic techniques. Therefore, digital forensic studies focusing on smart devices are required. In this study, a data acquisition framework for smart devices was applied to wearable devices such that they can be used in the investigation process through data acquisition and analysis. This can enable the investigation of wearable devices at crime scenes. Section 2 introduces related studies on smart devices. In Sect. 3, a data acquisition framework that can be used for smart devices is described. In Sect. 4, data acquisition and analysis enabled by applying the framework to a wearable device are presented. The experiment is described in Sect. 5, and the conclusions are provided in Sect. 6.

## 2 Related work

As mentioned before, smart cities provide services by being connected to external networks through the convergence of industry and Information and Communications Technology. Connecting to external networks provides convenience while also increases cyber threats. In order to respond to smart city cyber incidents, many security-related studies have been conducted on infrastructure such as industry and health care [6–13]. Digital forensic studies on smart devices are also being conducted extensively, such as smart home, wearable devices, and ecosystem that can be easily accessed. [14–17]. In studies pertaining to the smart home environment, data acquisition and analysis have been performed by configuring various home IoT devices to analyze user behavior [18–26]. Kim et al. proposed a digital forensic framework, as well as data acquisition and analysis methods, for Nest Hub, Samsung SmartThings, and Kasa cam, which are smart home IoT devices [25]. Hutchinson et al. explained security threat scenarios for various home IoT devices and analyzed home IoT devices such as smart hubs and smart cameras [26]. In addition, digital forensics for AI speakers that control the smart home environment have been investigated [27–32]. Shin et al. collected and analyzed encrypted traffic between devices and the cloud that stores user data for AI speakers; they developed a tool to acquire artifacts stored in the cloud [32].

In the case of wearable devices, because users always wear them to obtain information pertaining to health and location, several studies on the data acquisition

and analysis of wearable devices have been conducted [33–37]. MacDermott et al. acquired and analyzed important user data for Garmin, Fitbit, and HETP devices. They performed the imaging and analysis of Garmin instruments using several forensic tools to identify information including user data such as activity duration, GPS coordinates, user-defined settings, or activity logs stored in the devices. For Fitbit, they analyzed the Windows 10 application connected to the device and acquired user personal information from a database in a specific location on the PC. For HETP, they were unable to acquire user personal information from the device itself. Although they analyzed three wearable devices to acquire data that can be used as evidence, the target devices had been released a long time ago; therefore, their methods are restricted in terms of their application to the latest wearable devices [36]. Becirovic et al. proposed a data acquisition and analysis method for Gear S3 Frontier, which is a smartwatch manufactured by Samsung, and derived artifacts stored in the device. They connected the device to a PC through a Wi-Fi wireless connection and accessed the internals of the device using TizenOS via a smart development bridge. Subsequently, through device analysis, it was discovered that personal information such as linked smartphone information and phone number, as well as user artifacts such as GPS information, was stored in the smart device [37]. In addition, digital forensic investigations pertaining to smart devices have been conducted. However, owing to the characteristics of IoT devices, their internal structures differ by model and are miniaturized. Therefore, additional research must be conducted as the latest devices are being released.

## 3 Data acquisition framework for smart device forensics

Figure 1 shows the data acquisition framework used for performing experiments on smart devices in this study. The proposed framework uses software-based and hardware-based data acquisition methods for data extraction. Hardware-based data acquisition, such as JTAG and Chip-off, which should be preceded by PCB analysis, can cause permanent damage to the device. In the case of chip-off, physically separating NAND flash memory limits software-based

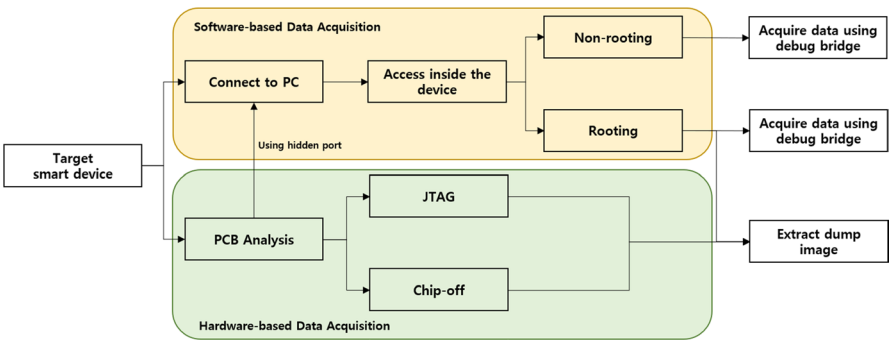


**Fig. 1** Data acquisition framework for smart devices

data acquisition. Therefore, software-based data acquisition must be conducted before hardware-based data acquisition. For software-based data acquisition, the device must be connected to PC first. Several methods can be used to connect a device to a PC, e.g., via USB, Wi-Fi, and Bluetooth connections. Although no other physical ports are officially provided by manufacturers to users, hidden ports for developers may exist. It may be possible to connect a device to a PC through a pin map on PCB. When a connection to the PC is established, then the inside of the device can be accessed. Android-based devices can be accessed through an Android debug bridge (ADB). After accessing the inside of the device, the internal data can be acquired using the debug bridge, etc., in a non-rooting state. However, because of permission issues, only some data can be acquired, and in some cases, important data cannot be obtained.

Alternatively, data can be obtained after administrator privileges are acquired through rooting. This rooting method is different for each operating system installed on a smart device, and rooting may be difficult for some operating systems. In addition, integrity may be violated during the rooting process. This rooting method presents several limitations; however, if administrator privileges are acquired, meaningful user data can be acquired.

In addition, data inside a device can be acquired using physical methods, including the joint test action group (JTAG) method. Flash memory can be imaged by connecting a pin corresponding to the JTAG port through PCB analysis. However, this is difficult to perform when the manufacturer removes the JTAG port or when the pin map is not known. Another method is the chip-off method, which physically separates the flash memory. In general, IoT devices use NAND flash memory. After separating the NAND flash memory in the device, it is mounted on a PC through a NAND flash reader to acquire data. However, expensive equipment is required to perform this chip-off process, which cannot be performed if the corresponding device does not use NAND flash memory. Therefore, in this study, we investigated whether the chip-off method is feasible. In addition, a few methods can be used to verify internal data through device manipulation, e.g., through manipulating displays and buttons, as well as to acquire data for connected smartphones. However, they are omitted in the framework proposed herein.

## 4 Data acquisition and analysis

In this study, data were acquired by applying the data acquisition framework to some wearable devices, and analysis was performed on the acquired data. Five wearable devices were used in the experiment, and the device information used in the experiment is listed in Table 1. These wearable devices were selected for study because they are the world's best-selling products and have been released by major wearable device manufacturers. In some cases, the FTK Imager and ADB were used for data acquisition and analysis.

**Table 1** Wearable device used for test

| Manufacturer | Model | Release date | OS |
|---|---|---|---|
| Xiaomi | Amazfit Stratos 3 | 2019 | Amazfit OS |
| Huawei | Huawei Watch GT 2 | 2019 | Harmony OS |
| LG | LG W7 | 2018 | Wear OS |
| Fitbit | Charge 4 | 2020 | Fitbit OS |
| Xiaomi | Mi Band 4 | 2019 | - |

### 4.1 A. Amazfit Stratos 3

The Amazfit Stratos 3 model can be connected to a PC using a rechargeable USB provided by the manufacturer. Since Amazfit OS is based on Android, it is compatible with the ADB. Therefore, the inside of the device can be accessed using the ADB, as shown in Fig. 2.

Moreover, the directory structure, file name, etc., can be obtained in the non-rooting state. However, as mentioned above, because a permission issue occurs (as shown in Fig. 3), data acquisition is restricted.

The Android-based Amazfit OS used by Stratos 3 can be rooted similarly as for an Android smartphone. The device rooting was performed by flashing the firmware with root privileges. After acquiring the root privilege, an image dump was performed, and the file system image was analyzed using FTK Imager, as shown in Fig. 4. Ext4 was used as the file system, implying that the deleted files can be recovered [38].

Table 2 shows the data acquired in Stratos 3. Forensic artifacts can be acquired from five files, and the data include those pertaining to the device settings, user data, and activities.

– **setting.db**

```
C:₩Users₩jino>adb shell
shell@watch:/ $ cd
acct/                   init.bluetooth.rc       mnt/
cache/                  init.environ.rc         proc/
charger                 init.factory.rc         root/
config/                 init.invensense.rc      sbin/
d/                      init.iptables.rc        sdcard/
data/                   init.rc                 storage/
default.prop            init.recovery.watch.rc  sys/
dev/                    init.trace.rc           system/
etc/                    init.usb.rc             ueventd.rc
file_contexts           init.watch.rc           ueventd.watch.rc
fstab.watch             init.watch.usb.rc       vendor/
init                    init.zygote32.rc
```

**Fig. 2** ADB access to Amazfit Stratos 3

```
C:₩Users₩jino>adb pull init.bluetooth.rc
adb: error: failed to copy 'init.bluetooth.rc' to '.₩init.bluetooth.rc': remote Permission denied
```

**Fig. 3** Permission issue in non-rooting state

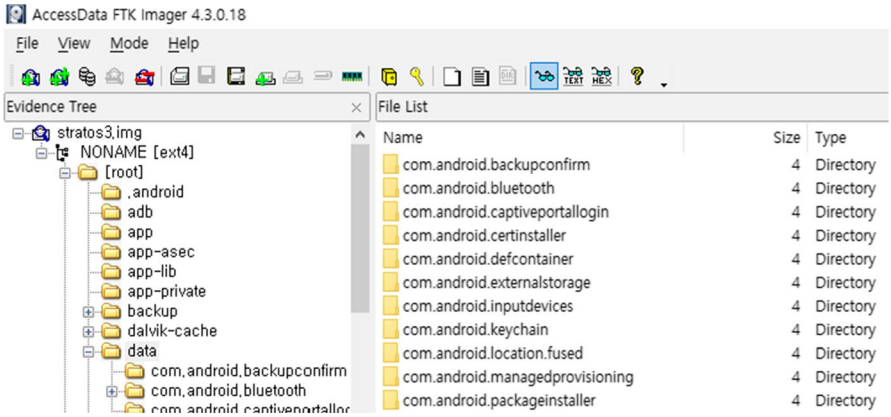


**Fig. 4** Internal data analysis using FTK Imager

setting.db file in the /data/com.android.providers.settings/databases/ path has seven tables and contains values for common device settings. Among the seven tables, the "global" and "secure" tables contain forensic artifacts. In "global" table, the smartwatch settings including the device name and last charging time are saved as shown in Fig. 5. As shown Fig. 5, the last charging time was 1,604,385,466,000. Through a Unix timestamp conversion, it can be seen that it was last charged at 6:37:46 (GMT + 0000) on November 3, 2020.

In "Secure" table, as shown in Fig. 6, it was confirmed that information related to the device communication has stored. Device_bound_time represents the estimated time when the smartphone was connected, and it was found to be connected on November 03, 2020, 04:55:43 in the same manner as described above. In addition, the Bluetooth name and Bluetooth address were stored.

– **external.db**

external.db is located in /data/data/com.android.providers.media/databases and contains file information saved in the device through a PC connection. external.db has 12 tables, and forensic artifacts is stored in the "file" table. Except for the "file" table, no important data were found, and some tables had no values. In "file" table, as shown in Fig. 7, the name, size, type, and date of the saved media file are stored.

– **Box.xml**

Smartwatch user information is stored in the Box.xml file in the /data/com.huami.watch.hmwatchmanager/shared_prefs/ path. As shown in Fig. 8, this file contains the birthday (month and year), height, gender (male: 1, female: 2), and wrist wearing smartwatch (left: 1, right: 2) information of the smartwatch user. In addition, user data such as weight and last charging time are stored. The last charging time of the Box.xml file has the same value as that of the setting.db file.

– **allday_heartrate.db**

Information regarding the user's heart rate is stored in the allday_heartrate.db file in the /data/com.huami.watch.health/databases/ path. allday_heartrate.db has two tables, and the "allday_heart_rate" table contains forensic artifacts.

**Table 2** Data acquired from Amazfit Stratos 3

| Path | File name | Information |
|---|---|---|
| /data/com.android.providers.settings/databases/ | Setting.db | Device name, last charging time, MAC address |
| /data/data/com.android.providers.media/databases | External.db | Media file information |
| /data/com.huami.watch.hmwatchmanager/shared_prefs/ | Box.xml | Device setting information |
| /data/com.huami.watch.health/databases/ | Allday_heartrate.db | Heart rate |
| /data/com.huami.watch.newsport/databases/ | Sport_data.db | Activity time, heart rate during activity |

| | | | |
|---|---|---|---|
| 39 | 42 | low_battery_sound_timeout | 0 |
| 40 | 43 | wifi_scan_always_enabled | 0 |
| 41 | 44 | heads_up_notifications_enabled | 1 |
| 42 | 45 | device_name | Amazfit Stratos 3 |
| 43 | 46 | guest_user_enabled | 1 |
| 44 | 47 | volte_vt_enabled | 1 |
| 45 | 48 | captive_portal_detection_enabled | 0 |
| 46 | 49 | wifi_watchdog_on | 1 |
| 47 | 50 | adb_enabled | 1 |
| 48 | 51 | network_scoring_provisioned | 1 |
| 49 | 52 | audio_safe_volume_state | 3 |
| 50 | 53 | device_provisioned | 1 |
| 51 | 59 | airplane_mode_on | 0 |
| 52 | 63 | zen_mode | 0 |
| 53 | 64 | bluetooth_on | 0 |
| 54 | 69 | last_charging_time | 1604385466000 |

**Fig. 5** "Global" table data in "setting.db"

| | | | |
|---|---|---|---|
| 38 | 49 | device_bound_time | 1604379343782 |
| 39 | 48 | device_bound | 1 |
| 40 | 37 | default_input_method | com.huami.watch.input/.HuamiIME |
| 41 | 54 | bluetooth_name | Amazfit-2CF3 |
| 42 | 44 | bluetooth_address | D8:80:3C:3F:2C:F3 |
| 43 | 45 | bluetooth_addr_valid | 1 |

**Fig. 6** "Secure" table data in "setting.db"

| | _id | _data | _size | format | parent | date_added | date_modified | mime_type | title |
|---|---|---|---|---|---|---|---|---|---|
| | F··· | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 534 | 534 | /storage/emulated/0/.ble/icon/··· noname1.jpg | 33453 | 14347 | 13 | 1604380768 | 1604380753 | image/png | com.benyhe.GragonBlastFree |
| 535 | 535 | /storage/emulated/0/.springchannel/··· | 759027 | 12288 | 183 | 1604380768 | 1604629455 | NULL | com.huami.watch |
| 536 | 536 | /storage/emulated/0/Download/무제 1.jpg | 528384 | 14337 | 8 | 1604381321 | 1604381321 | image/jpeg | 무제 1 |

**Fig. 7** "file" table data in "external.db"

As shown in Fig. 9, the "allday_heart_rate" table stores the heart rate of the user measured through the heart rate measurement application. The data are stored in Unix day format, and as shown, the value of 18,569 corresponds to November 3, based on Unix day conversion.

– **sport_data.db**

The sport_data.db file has 12 tables and stores sports data. There were no data in 11 tables except "heart_rate" table. As shown in Fig. 10, the "heart_rate" table

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="offlineSetWatchFaceInfo"></string>
    <long name="LastChargingTime" value="1604385466000" />
    <int name="month" value="5" />
    <int name="height" value="171" />
    <int name="gender" value="1" />
    <string name="uid">0</string>
    <int name="wear_hand" value="1" />
    <float name="weight" value="69.0" />
    <int name="year" value="1995" />
</map>
```

**Fig. 8** Data from "Box.xml"

Table: allday_heart_rate

| | _id | day_offset | average | reset_heart_rate | max_heart_rate | min_heart_rate | hight_percent |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 18569 | 81 | 0 | 86 | 77 | 0 |

**Fig. 9** "allday_heart_rate" table in the 'allday_heartrate.db'

Table: heart_rate

| | track_id | rate | step_freq | altitude | disdiff | pace | heart_quality | step_count | stride | run_time | heart_range | stroke_speed | time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1604381059000 | 0.0 | 0.0 | -20000.0 | 0.0 | 0 | 1000 | 0 | 0 | 1000 | 7 | 0.0 | 1604381060000 |
| 2 | 1604381059000 | 0.0 | 0.0 | -20000.0 | 0.0 | 0 | 1000 | 0 | 0 | 2000 | 7 | 0.0 | 1604381061000 |
| 3 | 1604381059000 | 100.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 3000 | 0 | 0.0 | 1604381062000 |
| 4 | 1604381059000 | 107.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 5000 | 1 | 0.0 | 1604381064000 |
| 5 | 1604381059000 | 109.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 6000 | 1 | 0.0 | 1604381065000 |
| 6 | 1604381059000 | 115.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 7000 | 1 | 0.0 | 1604381066000 |
| 7 | 1604381059000 | 112.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 8000 | 1 | 0.0 | 1604381067000 |
| 8 | 1604381059000 | 113.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 9000 | 1 | 0.0 | 1604381068000 |
| 9 | 1604381059000 | 112.0 | 0.0 | -20000.0 | 0.0 | 0 | 1 | 0 | 0 | 10000 | 1 | 0.0 | 1604381069000 |

**Fig. 10** "heart_rate" table in "sport_data.db"

saves the exercise start time, heart rate during exercise, and runtime. Therefore, fluctuations in heart rate over time can be measured.

### 4.2 B. Huawei Watch GT 2

In the Huawei Watch GT 2, the charging dock and smartwatch connection are composed of two pins, i.e., ground and power pins, as shown in Fig. 11. As these pins cannot transmit data, the device cannot be connected to a PC using the charging dock.

Huawei Watch GT 2 does not have an interface that can be connected to a PC, even in the PCB. Therefore, its internal data cannot be extracted by connecting it to

**Fig. 11** Huawei Watch GT 2's wired connection interface

a PC. In addition, the JTAG cannot be identified through PCB analysis; however, it contains Kington's 4 GB flash memory, as shown in Fig. 12. One should be able to extract the memory data using a reader compatible with flash memory.

#### 4.2.1 C. LG W7

The LG W7 model can be connected to a PC through USB for charging, and because the Android-based WearOS is used, the inside of the device can be accessed through the ADB, as shown in Fig. 13. Similar to Xiaomi, the file name and directory structure of LG W7 can be determined in a non-rooting state; however, we did not find any meaningful data through ADB because the privilege is limited. WearOS should be able to perform rooting by applying a methodology similar to that used in existing Android-based devices. However, rooting is not possible because the device's custom firm-

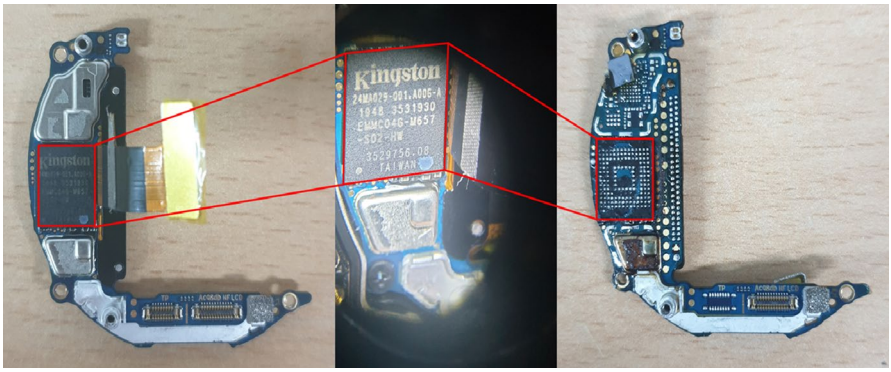


**Fig. 12** eMMC chip inside Huawei Watch GT 2

```
narwhal:/ $ ls -al
ls: ./persist: Permission denied
ls: ./verity_key: Permission denied
ls: ./ueventd.rc: Permission denied
ls: ./init.zygote32.rc: Permission denied
ls: ./init.usb.rc: Permission denied
ls: ./init.usb.configfs.rc: Permission denied
ls: ./init.recovery.narwhal.rc: Permission denied
ls: ./init.rc: Permission denied
ls: ./init.environ.rc: Permission denied
ls: ./init: Permission denied
total 48
drwxrwxrwt  18 root   root     660 2020-12-03 11:14 .
drwxrwxrwt  18 root   root     660 2020-12-03 11:14 ..
dr-xr-xr-x  37 root   root       0 2020-12-03 11:14 acct
lrwxrwxrwx   1 root   root      50 1970-01-01 09:00 bugreports -> /data/user_de/0/com.android.shell/files/bugreports
drwxrwx---   6 system cache   4096 2000-01-01 09:00 cache
lrwxrwxrwx   1 root   root      13 1970-01-01 09:00 charger -> /sbin/charger
drwxr-xr-x   4 root   root       0 1970-01-01 09:00 config
lrwxrwxrwx   1 root   root      17 1970-01-01 09:00 d -> /sys/kernel/debug
drwxrwx--x  34 system system  4096 2000-01-01 09:00 data
lrwxrwxrwx   1 root   root      23 1970-01-01 09:00 default.prop -> system/etc/prop.default
drwxr-xr-x  15 root   root    2100 2020-12-03 11:14 dev
lrwxrwxrwx   1 root   root      11 1970-01-01 09:00 etc -> /system/etc
dr-xr-x---   3 root   system 16384 1970-01-01 09:00 firmware
drwxr-xr-x  10 root   system   220 2020-12-03 11:14 mnt
drwxr-xr-x   4 root   root    4096 2018-08-29 03:33 oem
dr-xr-xr-x 316 root   root       0 1970-01-01 09:00 proc
drwx------   2 root   root      40 2019-02-20 03:07 root
drwxr-x---   2 root   root     120 1970-01-01 09:00 sbin
lrwxrwxrwx   1 root   root      21 1970-01-01 09:00 sdcard -> /storage/self/primary
drwxr-xr-x   4 root   root      80 2020-12-03 11:14 storage
dr-xr-xr-x  12 root   root       0 2020-12-03 11:14 sys
drwxr-xr-x  14 root   root    4096 2009-01-01 17:00 system
drwxr-xr-x  10 root   root    4096 2009-01-01 17:00 vendor
```

**Fig. 13** LG W7 internal data accessed through "adb shell" command

ware does not exist. In addition, through PCB analysis, a chip presumed to be a flash memory was discovered, as shown in Fig. 14; as such, data acquisition through chip-off is likely to be possible.

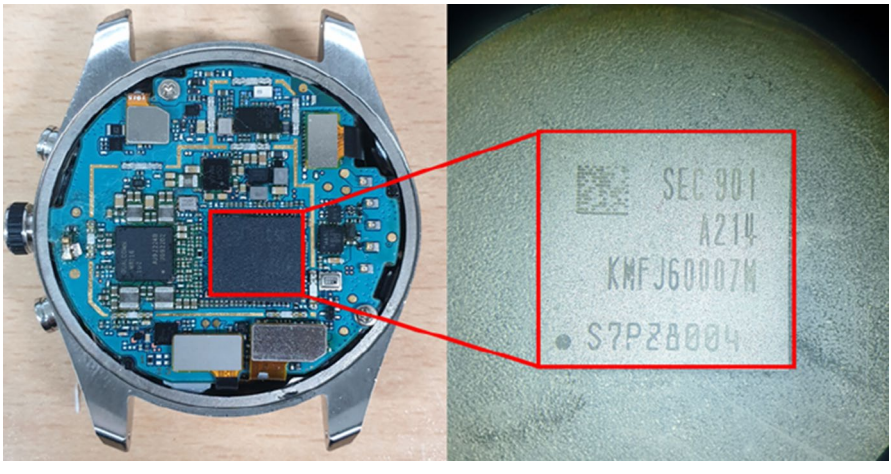


**Fig. 14** eMMC chip inside LG W7

### 4.3 D. Fitbit Charge 4

Fitbit Charge 4 models contain Fitbit firmware. Fitbit Charge 4 cannot be connected to a PC using a charging USB; however, it can be connected through Bluetooth. However, if the Fitbit application for Windows is not installed, then the PC cannot recognize the Fitbit. Therefore, internal access to the shell is difficult. This device is a smart band that is smaller and lighter than other smartwatches and provides relatively limited functions. Therefore, the usage of the JTAG will be more restricted owing to the small-sized board used, as shown in Fig. 15; furthermore, a flash memory for chip-off could not be identified.

### 4.4 E. Xiaomi Mi Band 4

The Xiaomi Mi Band 4 model uses its own firmware and cannot be connected to a PC using a charging dock. Therefore, its internal data cannot be acquired using a PC connection. As Mi Band 4 is as small as Fitbit Charge 4, the JTAG is not expected to function. However, as shown in Fig. 16, if a chip presumed to be a flash memory and a compatible reader for the chip exist, then data can be acquired.

## 5 Discussion

In this study, we proposed a framework for data acquisition of smart devices. To prove the proposed framework, an experiment was conducted on wearable devices. In the experiment, wearable devices from Xiaomi (Amazfit Stratos 3 and Mi Band 4), Huawei, LG, and Fitbit were used. For the devices other than Xiaomi's Amazfit Stratos 3, significant artifacts could not be acquired from a digital forensic perspective. In the case of Xiaomi's Amazfit Stratos 3, we use the charging dock to connect

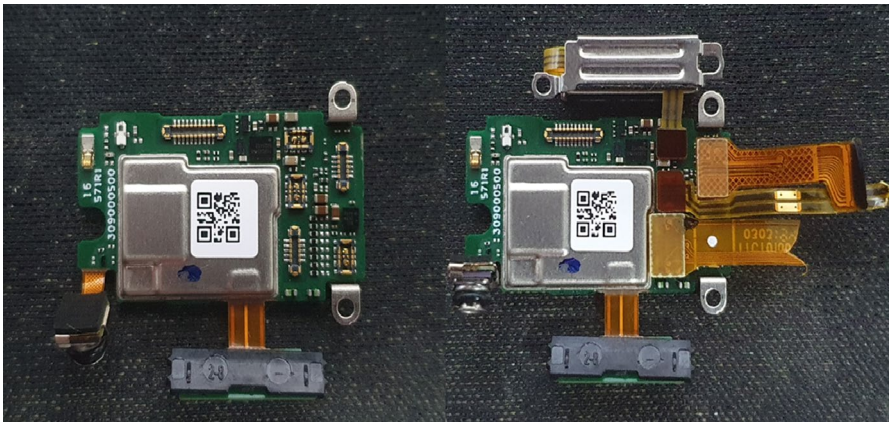

**Fig. 15** PCB Board of Fitbit Charge 4

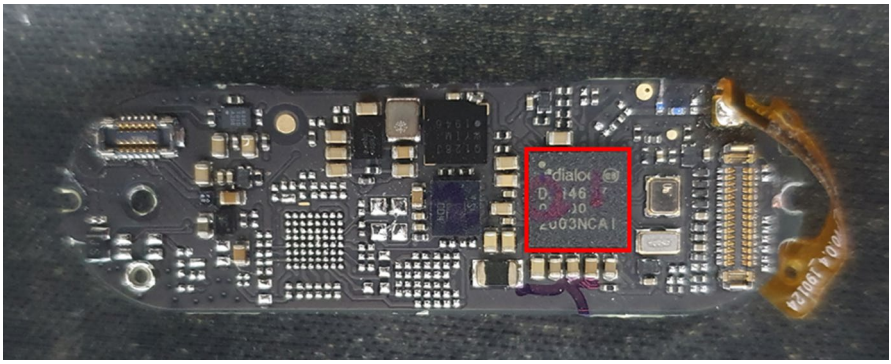

**Fig. 16** Xiaomi Mi Band 4 PCB Board

to the PC and use ADB to access the inside of the device. We were able to identify the file names and directory structures through ADB, but could not find any important data due to permission issues. Therefore, we tried to get the root privileges of Amazfit Stratos 3. We obtained root privileges through custom firmware and were able to extract forensic artifacts. Device information and user information could be obtained, and it is expected to be possible to identify the device user and prove an alibi based on heart rate and timestamp information. An overview of the information we have acquired is summarized in Table 2. In the case of the Huawei Watch GT 2, the charging dock provided by the manufacturer consists of two pins, so it can only charge the battery. We could not find hidden ports to connect to the PC, and we could not extract data through the PC connection. However, the eMMC chip was identified through PCB analysis and the possibility of data extraction through chip-off was confirmed. LG W7 was able to connect to a PC using the charging dock and access the inside of the device through ADB. However, we did not find any important data because of permission issues. We tried to get the root privileges of LG W7, but could not find a way to get root privileges. Afterward, the eMMC chip was identified through PCB analysis and the possibility of data extraction through chip-off was confirmed. Fitbit Charge 4 and Xiaomi Mi Band 4 were lighter than other smartwatches, so there were limitations in terms of data acquisition.

Overall, for software-based data acquisition of wearable devices, it must be connected to a PC. If the wearable device uses an Android-based operating system, it is possible to access the inside of the device through ADB. However, in the case of non-rooted devices, it was difficult to find meaningful forensic artifacts due to permission issues. If it is possible for a device to gain root privileges, it can acquire forensic artifacts. However, there is a limitation in that methods for acquiring root privileges are different for each operating system and there are few known methods. Therefore, it is necessary to study a method for obtaining an optimal root privileges for each wearable device. For hardware-based data acquisition of wearable devices, pcb must be analyzed first. If a JTAG port is found, the disk image can be extracted through JTAG. However, in this experiment, the JTAG port could not be found, so it could not be performed. Additionally, if the wearable device uses NAND flash

memory, the chip-off technique can be used. In this experiment, we found NAND flash memory of LG W7 and Huawei Watch GT2, and it is presumed that data can be acquired if there is a compatible reader.

## 6 Conclusion

Smart cities are composed of smart devices that use the latest ICT technologies to efficiently manage cities and improve the quality of life. Smart devices store user information to provide convenient services, which can also be used as key evidence in actual investigations. Therefore, in this study, data acquisition and analysis were performed by applying the proposed smart device data acquisition framework to five wearable devices. In the Xiaomi device, device information such as device name, last charging/connecting time, and MAC address, as well as user input information such as heart rate and exercise time could be acquired. In the Huawei and LG devices, it was confirmed that although the inside of the device could be accessed using a PC, meaningful data could not be obtained, and that data could be obtained using chip-off technology. Smart bands such as Fitbit Charge 4 and Mi Band 4 are more compact than smartwatches and have fewer functions, rendering it difficult to apply a logical/physical forensic method on them.

In future studies, the acquisition of artifacts through a detailed analysis of operating systems in wearable devices such as Samsung's TizenOS and Apple's WatchOS, as well as WearOS and Amazfit OS, should be investigated. In addition, because smart devices generally use flash memory, further research regarding physical acquisition methods for smart devices such as chip-off and JTAG should be performed.

**Acknowledgements** This research was supported by the Energy Cloud R&D Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2019M3F2A1073385)

## References

1. Subramaniyaswamy V, Manogaran G, Logesh R, Vijayakumar V, Chilamkurti N, Malathi D, Senthilselvan N (2019) An ontology-driven personalized food recommendation in IoT-based healthcare system. J Supercomput 75(6):3184–3216
2. Ghahramani, M., Javidan, R., & Shojafar, M. (2020). A secure biometric-based authentication protocol for global mobility networks in smart cities. The Journal of Supercomputing, 1–27.
3. Chaudhry SA, Naqvi H, Farash MS, Shon T, Sher M (2018) An improved and robust biometrics-based three factor authentication scheme for multiserver environments. J Supercomput 74(8):3504–3520
4. Nicole Chavez (2017) Arkansas judge drops murder charge in Amazon Echo case. CNN. https://edition.cnn.com/2017/11/30/us/amazon-echo-arkansas-murder-case-dismissed/index.html. Accessed 2 December 2017
5. Tracy Connor (2017) Fitbit Murder Case: Richard Dabate Pleads Not Guilty in Wife's Death. NBC-NEWS. https://www.nbcnews.com/news/us-news/fitbit-murder-case-richard-dabate-pleads-not-guilty-wife-s-n752526 Accessed 30 Apr 2017
6. Cosic, J., Schlehuber, C., & Morog, D. (2021, April). Digital Forensic Investigation Process in Railway Environment. In 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1–6). IEEE

7. Jayaraman I, Panneerselvam AS (2021) A novel privacy preserving digital forensic readiness provable data possession technique for health care data in cloud. J Ambient Intell Humaniz Comput 12(5):4911–4924
8. Kim, S., Jo, W., & Shon, T. (2020). APAD: autoencoder-based payload anomaly detection for industrial IoE. Applied Soft Computing, 88, 106017
9. Kwon S, Yoo H, Shon T (2020) IEEE 1815.1-based power system security with bidirectional RNN-based network anomalous attack detection for cyber-physical system. IEEE Access 8:77572–77586
10. Chaudhry SA, Shon T, Al-Turjman F, Alsharif MH (2020) Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems. Comput Commun 153:527–537
11. Kim, H., Kim, S., Jo, W., Kim, K., & Shon, T. (2021). Unknown Payload Anomaly Detection Based on Format and Field Semantics Inference in Cyber-Physical Infrastructure Systems. IEEE Access
12. Iqbal, A., Mahmood, F., & Ekstedt, M. (2019). Digital forensic analysis of industrial control systems using sandboxing: A case of wampac applications in the power systems. energies, 12(13), 2598
13. Rondeau, C. M., Temple, M. A., & Lopez, J. (2019). Industrial IoT cross-layer forensic investigation. Wiley Interdisciplinary Reviews: Forensic Science, 1(1), e1322
14. Losavio, M. M., Chow, K. P., Koltay, A., & James, J. (2018). The Internet of Things and the Smart City: Legal challenges with digital forensics, privacy, and security. Security and Privacy, 1(3), e23
15. Sathwara, S., Dutta, N., & Pricop, E. (2018, June). IoT Forensic A digital investigation framework for IoT systems. In 2018 10th International Conference on Electronics, Computers And Artificial Intelligence (ECAI) (pp. 1–4). IEEE
16. Quick D, Choo KKR (2018) IoT device forensics and data reduction. IEEE Access 6:47566–47574
17. Feng, X., Dawam, E. S., & Amin, S. (2017). Digital forensics model of smart city automated vehicles challenges
18. Goudbeek, A., Choo, K. K. R., & Le-Khac, N. A. (2018, August). A forensic investigation framework for smart home environment. In 2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE) (pp. 1446–1451). IEEE
19. Do Q, Martini B, Choo KKR (2018) Cyber-physical systems information gathering: a smart home case study. Comput Netw 138:1–12
20. Iqbal, A., Olegård, J., Ghimire, R., Jamshir, S., & Shalaginov, A. (2020, December). Smart Home Forensics: An Exploratory Study on Smart Plug Forensic Analysis. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 2283–2290). IEEE
21. Dorai, G., Houshmand, S., & Baggili, I. (2018, August). I know what you did last summer: Your smart home Internet of Things and your iPhone forensically ratting you out. In Proceedings of the 13th International Conference on Availability, Reliability and Security (pp. 1–10)
22. Philomin, S., Singh, A., Ikuesan, A., & Venter, H. (2020). Digital forensic readiness framework for smart homes. In International Conference on Cyber Warfare and Security (pp. 627-XVIII). Academic Conferences International Limited
23. Awasthi A, Read HO, Xynos K, Sutherland I (2018) Welcome pwn: almond smart home hub forensics. Digit Investig 26:S38–S46
24. Azhar, M. H. B., & Bate, S. B. L. (2019). Recovery of Forensic Artefacts from a Smart Home IoT Ecosystem. IARIA
25. Kim S, Park M, Lee S, Kim J (2020) Smart home forensics—data analysis of IoT devices. Electronics 9(8):1215
26. Hutchinson, S., Yoon, Y. H., Shantaram, N., & Karabiyik, U. Internet of Things Forensics in Smart Homes: Design, Implementation and Analysis of Smart Home Laboratory
27. Yıldırım, İ., Bostancı, E., & Güzel, M. S. (2019, September). Forensic Analysis with Anti-Forensic Case Studies on Amazon Alexa and Google Assistant Build-In Smart Home Speakers. In 2019 4th International Conference on Computer Science and Engineering (UBMK) (pp. 1–3). IEEE
28. Youn, M. A., Lim, Y., Seo, K., Chung, H., & Lee, S. (2021). Forensic Analysis for AI Speaker with Display Echo Show 2nd Generation as a Case Study. Digital Investigation
29. Jo W, Shin Y, Kim H, Yoo D, Kim D, Kang C, Shon T (2019) Digital forensic practices and methodologies for AI speaker ecosystems. Digit Investig 29:S80–S93
30. Li S, Choo KKR, Sun Q, Buchanan WJ, Cao J (2019) IoT forensics: Amazon echo as a use case. IEEE Internet Things J 6(4):6487–6497
31. Chung H, Park J, Lee S (2017) Digital forensic approaches for Amazon Alexa ecosystem. Digit Investig 22:S15–S25

32. Shin, Y., Kim, H., Kim, S., Yoo, D., Jo, W., & Shon, T. (2020). Certificate Injection-Based Encrypted Traffic Forensics in AI Speaker Ecosystem. Forensic Science International: Digital Investigation, 33, 301010
33. Kang S, Kim S, Kim J (2020) Forensic analysis for IoT fitness trackers and its application. Peer-to-Peer Net Appl 13(2):564–573
34. Odom NR, Lindmar JM, Hirt J, Brunty J (2019) Forensic inspection of sensitive user data and artifacts from smartwatch wearable devices. J Forensic Sci 64(6):1673–1686
35. Gregorio J, Alarcos B, Gardel A (2019) Forensic analysis of nucleus RTOS on MTK smartwatches. Digit Investig 29:55–66
36. MacDermott, Á., Lea, S., Iqbal, F., Idowu, I., & Shah, B. (2019, June). Forensic analysis of wearable devices: Fitbit, Garmin and HETP Watches. In 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1–6). IEEE
37. Becirovic, S., & Mrdovic, S. (2019, September). Manual IoT Forensics of a Samsung Gear S3 Frontier Smartwatch. In 2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM) (pp. 1–5). IEEE
38. Lee, S., Jo, W., Eo, S., & Shon, T. (2019). ExtSFR: scalable file recovery framework based on an Ext file system. Multimedia Tools and Applications, 1–19