Nyon Unchained: Forensic Analysis of Bosch's eBike Board Computers

Marcel Stachak[a], Julian Geus[a,*], Gaston Pugliese[a] and Felix Freiling[a]

[a] Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany

ARTICLE INFO

Keywords:
bike computer
mobile forensics
IoT forensics
storage acquisition

ABSTRACT

Modern eBike on-board computers are basically small PCs that not only offer motor control, navigation, and performance monitoring, but also store lots of sensitive user data. The Bosch Nyon series of board computers are cutting-edge devices from one of the market leaders in the eBike business, which is why they are especially interesting for forensics. Therefore, we conducted an in-depth forensic analysis of the two available Nyon models released in 2014 and 2021. On a first-generation Nyon device, Telnet access could be established by abusing a design flaw in the update procedure, which allowed the acquisition of relevant data without risking damage to the hardware. Besides the user's personal information, the data analysis revealed databases containing user activities, including timestamps and GPS coordinates. Furthermore, it was possible to forge the data on the device and transfer it to Bosch's servers to be persisted across their online service and smartphone app. On a current second-generation Nyon device, no software-based access could be obtained. For this reason, more intrusive hardware-based options were considered, and the data could be extracted via chip-off eventually. Despite encryption, the user data could be accessed and evaluated. Besides location and user information, the newer model holds even more forensically relevant data, such as nearby Bluetooth devices.

1. Introduction

Despite constant—and partly spectacular (Garfinkel, 2013)—advances (Casey, 2011; Freiling et al., 2018) in the area of digital forensic science, the analysis of concrete digital devices remains a cumbersome undertaking (Garfinkel, 2010). While much standardization has occurred, especially with respect to interfaces to devices (e.g., JTAG, USB) and storage (e.g., SATA, MMC), the variation of devices themselves has increased. A specific driving force behind this development is the proliferation of *special-purpose mobile devices*, i.e., devices specially built to fulfil a certain task or purpose. The forensic analysis of such devices is the focus of this paper.

In their *non*-mobile form, such special-purpose devices usually run under the heading of *embedded systems* and have appeared in the form of point-of-sale terminals, cash machines, or information displays. In contrast to general-purpose *mobile* devices like smartphones, special-purpose mobile devices are *mobile* embedded systems with a specific purpose. Examples are navigation systems, fitness trackers, or robot vacuum cleaners. The *Internet of Things* (IoT), as a buzzword, is often used to characterize special-purpose embedded systems.

An interesting type of special-purpose mobile devices has been on the market for almost 10 years: Bike computers are the information hub for electric bikes, also known as "pedelecs". According to TheRoundup.org (2023), pedelecs have a rapid increase in sales, which is caused by the broad target market that covers all age groups, and since they are often a cheaper and a more environmental friendly alternative to cars. Their ever-growing numbers and increasing relevancy in modern day transportation further supports the importance of considering their accompanying bike computers as sources of forensically relevant evidence. One of the market leaders of digital technology in Germany is the company Bosch. Bosch nowadays speaks about "connected biking". Especially their most feature-rich series of bike computers, called *Nyon*, might store lots of user-specific data and is therefore the focus of our analysis.

1.1. Related Work

Research in the field of non-classical PC forensics is necessarily based on specific case studies and many such studies have been performed. One of the earliest and most well-known studies, with more focus on reverse engineering than on forensic analysis, was performed when Huang (2003) "hacked the Xbox". On the mobile side of gaming consoles and with focus on forensic analysis, Barr-Smith et al. (2021) analyzed the portable video gaming console Nintendo Switch. The authors used an exploit to gain access to the device and acquire a memory dump. They meticulously analyzed the data and were able to identify numerous forensically interesting traces.

In the field of IoT forensics, Youn et al. (2021) analyzed Amazon's AI speaker Echo Show. The authors extracted the data via chip-off and also took smartphone data from the companion app into account. In contrast, Villarreal et al. (2022) developed a non-destructive method to acquire data from the flash memory chip of Amazon's Echo Dot devices.

Cars and their built-in infotainment systems are an interesting case since those parts of the system that coordinate the brakes and steering have a special purpose while the generic parts of the automotive computing system that are responsible for entertainment typically allow general-purpose computations. Automotive digital forensics (Strandberg et al.,

*Corresponding authors.
Email addresses: marcel.stachak@fau.de (M. Stachak); julian.geus@fau.de (J. Geus); gaston.pugliese@fau.de (G. Pugliese); felix.freiling@fau.de (F. Freiling)
ORCID(s): 0009-0001-8270-1964 (J. Geus); 0000-0002-8279-8401 (F. Freiling)

2023) has been of special interest for many years. While earlier work by Hoppe et al. (2012) focused on extracting route information from the car itself, Ebbers et al. (2021) attempted to extract the same data from the companion apps of car manufacturers.

Due to the heterogeneity of these devices, it is hard to come up with practical yet generic analysis guidelines. Gómez et al. (2021) proposed a generic investigation methodology for IoT devices, while Buquerin et al. (2021) evaluated digital forensic processes in the automotive domain and developed a general process for automotive digital forensic investigations. Especially because of the widespread usage of encryption, as well as other security measures, Fukami et al. (2021) proposed a model for data acquisition from mobile devices with particular emphasis on smartphones. In special-purpose mobile devices, the security might not be as advanced, therefore, this model only partially applies. Still, specific-purpose mobile devices are usually feature rich, often run a full Linux or Android operating system, but do not allow running custom software like apps. A generic data acquisition strategy for more feature-restricted devices is still an open research question.

1.2. Contributions

Generally, we are not aware of any work that has analyzed eBike computers for their forensic or security aspects. Therefore, the contributions of this paper are as follows:

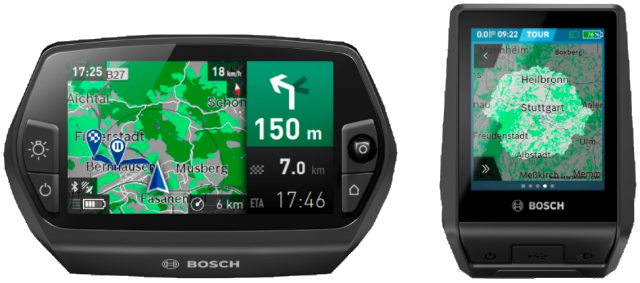
- To the best of our knowledge, we are the first to forensically analyze the first- and second-generation versions of the Bosch Nyon eBike board computer.
- We identified a design flaw in the update process of the Nyon 2014 that enables data acquisition without the need for intrusive hardware-based options.
- We developed a data acquisition methodology for special-purpose mobile devices, including an assessment of the forensic requirements.
- We conducted an in-depth data analysis, highlighting the forensic value of special-purpose mobile devices.

1.3. Outline

In Section 2, we provide background information on Bosch's Nyon devices, including hardware and software details. In Section 3, we address challenges for data acquisition on mobile devices, and propose a methodology adapted to the Nyon devices. Based on this methodology, the analysis results for the first-generation Nyon device are reported in Section 4, and for the second-generation Nyon device in Section 5. After discussing our findings in Section 6, we conclude the paper in Section 7.

2. Bosch Nyon Computers

Nyon devices are the premium class of portable bike computers by Bosch for electronic bicycles with supported "eBike drives". Since 2014, Bosch released two models of the Nyon computer which are shown in Figure 1. Both of them are connected to the electronics of the eBike through a docking station located on the bike's handlebar.



(a) Nyon 2014

(b) Nyon 2021

Figure 1: Bosch Nyon computers (© *Robert Bosch GmbH*).

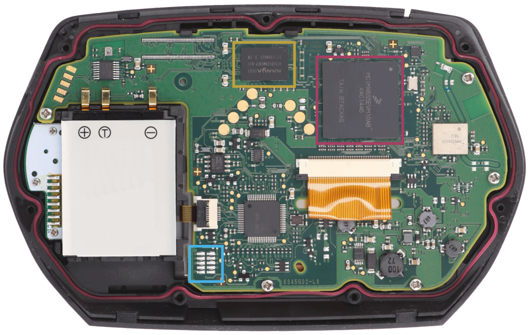


Figure 2: Main logic board (MLB) of the Nyon 2014 highlighting the CPU, DRAM, and a possible debug interface.

2.1. Bosch Nyon 2014

The first generation of the Bosch Nyon computer was released in 2014 and offered with 1 GB ("BUI270") and 8 GB ("BUI275") of internal storage. The graphical user interface on the 4.3-inch non-touch screen (480x270 pixel) can be controlled via an analog joystick and three buttons.

Figure 2 shows the main logic board (MLB) of the "BUI275" version of the Nyon 2014 after removing the rear casing, and reveals the following components:

- i.MX 6Solo CPU (MCIMX6S5EVM10AB) by NXP Semiconductors (2018) with 32-bit ARM Cortex-A9,
- two DDR3(L) SDRAM chips (NT5CB128M16FP-DII) by Nanya Technology (2015) with 128 MB each,
- eMMC with 8 GB (FBGA code "JWB18") by Micron Technology (2018).

The open-source software utilized in "eBike Systems" products can be determined on Bosch's license website (Robert Bosch GmbH, 2023b). Thereby, it is revealed that the Nyon 2014, for instance, is based on Linux kernel 3.0.35, uses U-Boot as bootloader, and incorporates parts of the Android Open Source Project (AOSP). During our analysis of the Nyon 2014 in Section 4, we further identified the Linux distribution to be "Poky Linux", a reference distribution of the Yocto Project (2023) developed for embedded systems.

The micro USB port of the Nyon 2014 can be utilized to charge the device or to connect it to a PC running Bosch's "DiagnosticTool", the debugging and diagnosis software for eBike manufacturers or retailers. An Internet connection can be established with the built-in Wi-Fi module to synchronize

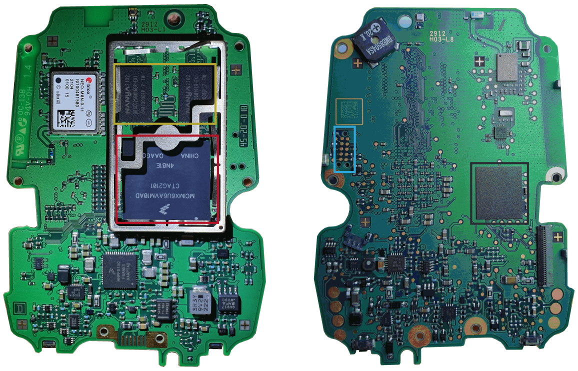


Figure 3: Main logic board (MLB) of the Nyon 2021 highlighting the CPU, DRAM, a possible debug interface, and eMMC.

data to Bosch's cloud or to download map data. Via Bluetooth, the Nyon can be paired with a smartphone running the "Bosch eBike Connect" app to obtain mobile Internet access. Bluetooth can also be utilized to connect supported peripherals which are, however, limited due to the missing support for Bluetooth Low Energy (BLE).

2.2. Bosch Nyon 2021

The second-generation and latest Nyon device is known as "BUI350" and was released in 2021. Its more compact design and portrait-format 3.2-inch touch screen (370x454 pixel) distinguishes the Nyon 2021 computer from its first-generation predecessor (cf. Figure 1).

Figure 3 shows the front and back side of the MLB, and highlights the following identified components:

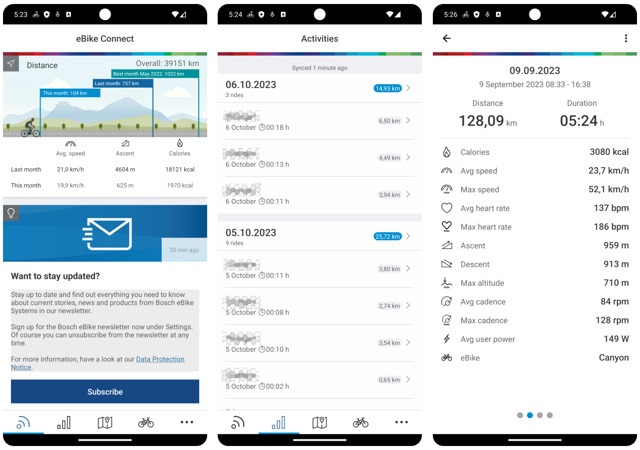
- i.MX 6Solo CPU (MCIMX6U8DVM10AD) by NXP Semiconductors (2018) w/ 32-bit ARM Cortex-A9,
- two DDR3(L) SDRAM chips (NT5CC256N16ER-EKI) by Nanya Technology (2022) w/ 256 MB each,
- eMMC with 8 GB (THGBMJG6C1LBAIL) by Kioxia Corp. (2019).

Bosch's license website (Robert Bosch GmbH, 2023b) discloses the use of Linux kernel 4.19.44, and our analysis in Section 5 again confirmed the usage of the Poky Linux distribution (cf. Section 2.1). Contrary to its predecessor, however, the Nyon 2021 has more advanced security mechanisms activated: The Linux kernel feature "dm-verity" checks the integrity of selected partitions on boot to ensure an uncompromised OS. "AppArmor" allows fine-grained privilege settings, restricts access to certain files or hardware devices for applications, and user data is encrypted using "Linux Unified Key Setup" (LUKS) with "cryptsetup". These security measures are similar to those of modern smartphones.

Connectivity-wise, the Nyon 2021 features a micro-AB USB port, Wi-Fi (2.4 GHz) with the now outdated 802.11b/g/n standard, and Bluetooth for the communication with the companion app. Due to its BLE support, the spectrum of pairable peripheral devices is broader compared to the first Nyon.

2.3. Smartphone App and Web Service

Nyon computers are accompanied by the smartphone app "Bosch eBike Connect" which is available for Android and iOS. As shown in Figure 4, the app focuses on essential aspects related to users' cycling activities. While the first tab contains a monthly statistic on distances, average speeds, ascents, and calories (Figure 4a), the second tab provides an overview of recent trips on a daily basis (Figure 4b), including a detail view for each of them which reveals additional information on speed, heart rate, altitude, or cadence (Figure 4c). Moreover, users can plan trips and transfer them to the Nyon, as well as synchronize data with Bosch's cloud.



5:23

eBike Connect

Distance

Overall: 39151 km

Best month May 2022: 1022 km

Last month: 757 km

This month: 104 km

	Avg. speed	Ascent	Calories
Last month	21,0 km/h	4604 m	18121 kcal
This month	19,9 km/h	625 m	1970 kcal

30 min ago

Want to stay updated?

Stay up to date and find out everything you need to know about current stories, news and products from Bosch eBike Systems in our newsletter.

Sign up for the Bosch eBike newsletter now under Settings. Of course you can unsubscribe from the newsletter at any time.

For more information, have a look at our Data Protection Notice.

Subscribe

(a) Monthly statistics

5:24

Activities

Synced 1 minute ago

06.10.2023 — 3 rides — 14,93 km

[redacted] 6 October 00:18 h — 6,50 km

[redacted] 6 October 00:13 h — 4,49 km

[redacted] 6 October 00:11 h — 3,94 km

05.10.2023 — 9 rides — 25,72 km

[redacted] 5 October 00:11 h — 3,80 km

[redacted] 5 October 00:08 h — 2,74 km

[redacted] 5 October 00:10 h — 3,54 km

[redacted] 5 October 00:02 h — 0,65 km

(b) Recent trips

5:26

09.09.2023

9 September 2023 08:33 - 16:38

Distance	Duration
128,09 km	05:24 h

Calories	3080 kcal
Avg speed	23,7 km/h
Max speed	52,1 km/h
Avg heart rate	137 bpm
Max heart rate	186 bpm
Ascent	959 m
Descent	913 m
Max altitude	710 m
Avg cadence	84 rpm
Max cadence	128 rpm
Avg user power	149 W
eBike	Canyon

(c) Trip details

Figure 4: Screenshots of the eBike Connect app.

Another service by Bosch available for registered users is the "eBike Connect" website (Robert Bosch GmbH, 2023a). Although the functional scope of the website and the app are almost identical, the former additionally enables the export of trip data as GPX files, including all coordinates.

Both the companion app and the web service require an "eBike Connect" online account which can be registered for free by providing country/language, first name, last name, email address, and a password.

3. Data Acquisition Methodology

In classical digital forensics, the established paradigm of acquiring data is to perform a bitwise copy of a hard drive. This approach is easy to execute, ensures high data integrity, especially when the hard drive is mounted read-only, can be easily verified and repeated, and provides us with a comprehensive data set. Since we are dealing with a special-purpose mobile device, a lot of the same problems encountered in mobile forensics also apply. Those challenges are described by Fukami et al. (2021), which are mainly the widespread use of encryption and inaccessible key material, rendering acquisition by direct memory access hard or sometimes even impossible. This fact necessitates the use of imperfect acquisition methods, like the device's backup mechanisms, as described by Geus et al. (2023). However, the Nyon devices might not be as advanced as modern smartphones when it comes to security, and therefore, we could face only some of the same challenges.

Before we define a general acquisition methodology, we first investigate the main priorities that have to be taken into account for forensic data acquisition, as well as the challenges with those in mobile forensics:

- **Integrity of Data**: One of the key points is to ensure the integrity of the data during the process. Therefore, the acquisition method has to be chosen carefully. However, in the mobile forensics sector, the data's integrity cannot always be ensured. Hence, an imperfect best-effort approach is sometimes required.

- **Amount of Data**: Obviously, an important goal of data acquisition is to extract a comprehensive data set. The best-case scenario is a bitwise copy of the entire memory device. Not only would we be able to analyze all the files of the user and the OS, but we could also restore deleted data without having to worry about data alterations. With mobile devices, however, this is not easily possible due to the integrated hardware, which necessitates intrusive methods, and the widespread use of encryption. It is also possible to create a bitwise copy from within the host OS which, however, requires OS access and the necessary permissions.

- **Repeatability**: Another key factor, which needs to be considered, is the repeatability of the process. Especially with mobile devices, where no software-based access is possible, hardware-based methods are a viable option. These, however, can be heavily intrusive and might involve, besides the dismantling of the device, the usage of soldering, or even the removal of the flash memory chip. By executing such processes, the device or flash memory is likely to be damaged, which might lead to data loss. Therefore, utilization of such methods needs to be considered carefully.

- **Ease-of-Use**: Time is always an important factor in an investigation, and even if a method is theoretically possible, its practical execution might entail a lot of effort. Gaining privileged access to the OS of a mobile device, for example, might include reversing and exploiting vulnerabilities, which is often very time-consuming and probably out-of-scope for most investigations. This obviously heavily depends on the situation. Sometimes screenshots of a chat history might be sufficient, and other times a detailed analysis of a database file or restoring deleted files are essential to solving the case.

For mobile forensics, most of those criteria have a conflict of interest. High data integrity is ensured when we directly access the storage medium, which for mobile devices means possibly destructive acquisition methods that are not easy to execute. Copying data from the OS might be simple, but the amount of data could be limited, and the integrity cannot be ensured. Therefore, a sufficient compromise has to be found, where we try to take all those points into account.

Concerning the amount of extractable data, three categories of data acquisition for mobile devices, as described by Casey and Turnbull (2011), are differentiated. The simplest way of gathering data is *manual acquisition*, where the device is used as intended and important information is preserved by screenshots. For the *logical acquisition*, some form of interaction with the file system, usually by using the OS interface, is necessary. The simplest form can be a publicly accessible directory with shared data when the device is plugged into a PC (e.g., the camera directory of smartphones) up to a full file system copy when root access to the device's OS is possible. The *physical acquisition* is the most complete form of data acquisition, since a bitwise copy of the memory device is created. For mobile devices, such an image can either be extracted from within the OS or by directly accessing the flash memory through hardware-based methods.

By analyzing this information, we define a data acquisition sequence that considers all criteria as well as possible. It is sequentially executed for both Nyon devices in the order of the following subsections until we acquire a sufficient amount of data.

3.1. Manual Acquisition

First, we used manual acquisition to evaluate how much information can be accessed using the capabilities of the user interface. This process is very easy to use and easily repeatable but might alter data on the device, which is why measures such as disabling the Internet connection and thoroughly documenting the process are essential. Since we did not expect to acquire a lot of information from this process, at least one other acquisition method is executed to obtain files from the device.

3.2. OS-based Acquisition

Afterwards, we tried to gain access to the OS of the device, which would enable us to copy data from the file system. If we were able to also acquire root privileges, both logical and physical acquisition could be executed from within the OS. We call this step *OS-based acquisition*, which has the advantage of possibly gaining access to the entire data of the device with minimal risk of damage or data loss. The process of gaining access to the OS, however, may vary in complexity depending on the method used. Additionally, data integrity cannot be guaranteed, because the OS might be compromised, or background processes may concurrently change data, especially when an Internet or Bluetooth connection is established.

3.3. Hardware-based Acquisition

If an OS-based acquisition was not possible, direct hardware access is considered. As a physical acquisition technique, hardware-based acquisition has the advantage of providing us with a complete data set. The integrity of the acquired data is also very high since we do not rely on

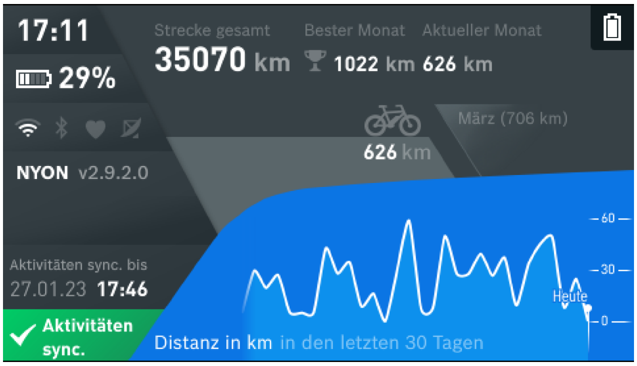


Figure 5: Dashboard of Nyon 2014.

the OS of the device. However, those methods require special hardware tools as well as knowledge and practice to minimize the risk of damage. Therefore, we first try to use debug interfaces to access the memory chip's data, before we actually consider the more destructive chip-off technique.

4. Forensic Analysis of the Nyon 2014

In this section, we forensically analyze the 8 GB version ("BUI275") of the first-generation Nyon computer (cf. Section 2.1). We start by evaluating a suitable acquisition method for the Nyon 2014 (cf. Section 3). Then, we analyze the acquired data to identify forensically relevant traces, and evaluate the possibility of data tampering.

4.1. Data Acquisition

The possibilities for data acquisition based on connectivity and hardware features of the Nyon 2014 are discussed in the following and follow the data acquisition sequence presented in the Sections 3.1–3.3. Finally, the method deemed most suitable for data acquisition is executed and the amount of extractable data is described in detail.

4.1.1. Manual Acquisition

The information provided by the user interface of the Nyon 2014 is limited. The dashboard depicted in Figure 5 shows a general overview of the user's monthly cycling statistics as well as the software version and the last synchronization with the Bosch cloud.

In the settings, the user can be identified by name and email address, but no other personal information is disclosed. Further, the SSIDs of the wireless networks the Nyon 2014 was connected to are accessible. In the navigation screen, recent destinations are available which, however, are lacking respective timestamps. This already concludes the amount of data obtainable by manual acquisition.

4.1.2. OS-based Acquisition

When connecting the Nyon 2014 to a PC via USB, accessing the device's file system is not possible, but Bosch's official "DiagnosticTool" is able to communicate with the device. The software enables certified bike shops and manufacturers to debug and update Nyon computers. It can be freely downloaded by anyone (Robert Bosch GmbH, 2023c) but requires a hardware token to run. The most interesting feature of Bosch's DiagnosticTool is the update process as it might reveal internals about the Nyon device, and altering the process might enable access to the OS. Hence, we analyzed the update process in further detail.

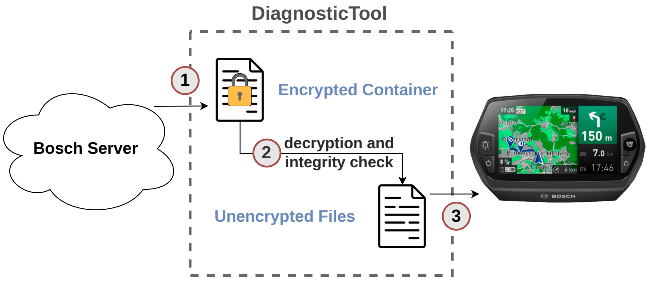


Figure 6: Update process of the Nyon 2014.

Update Process. Through black-box analysis, we determined that the update process basically consists of three steps which are illustrated in Figure 6:

1. The DiagnosticTool either (i) establishes a connection to Bosch servers to download the latest update container file, or (ii) offers to choose a local file. The update container file with the file extension `cff2` turned out to be a ZIP-compressed archive. By unzipping, we uncovered a `meta.xml` file that revealed the signing of the update container by Bosch. Furthermore, important parts of the extracted files were encrypted.

2. After the update container file was selected, the DiagnosticTool indicates to "prepare flash data", which seems to be the extraction and decryption process that takes place entirely on the PC running the software.

3. The last step is a reboot of the Nyon computer before the decrypted firmware files are transferred and installed, followed by a second reboot which concludes the process.

Exploiting a Design Flaw in Update Process. Through signing and encryption, the update process hinders us from forging our own update package. However, due to the fact that the decryption and integrity check take place on the PC and not on the Nyon, we were able to interfere with the process as follows to enable Telnet access:

1. By letting the program run in a sandbox which stores all files of the process locally, we gained access to the decrypted firmware files.

2. Through analysis of the update data, the Nyon's initialization script (`start.sh`) that executes software on startup was identified. We altered the script to start a Telnet server on the Nyon device.

3. The timeframe between steps 2 and 3 of the update process was utilized to exchange the original files with our forged ones. Therefore, we bypassed the integrity check and the decryption procedure, which is how our forged data was copied onto the device as a legitimate update.

This design flaw in the update process gave us root access to the Nyon 2014 using the Telnet protocol. To easily acquire the data, we started an FTP server on the device which enabled us to browse and copy data from the file system.

Forensic Soundness. Naturally, the changes applied to a system when installing an update might alter or delete forensically relevant user data or system files. While it is correct that parts of the OS are overwritten, the user data is usually not affected. This also applies in our case:

We discovered that the execution of the update process is accomplished by the "MFGTool" of NXP, the manufacturer of the Nyon 2014's CPU (cf. Section 2.1). This tool requires an XML file with instructions on how the process is executed. By inspecting the `ucl2_update_full.xml` file, we found that the forensically relevant user partition remains mostly untouched, except for two system directories that are temporarily stored on this partition to be restored after the system update: `/var/lib/connman/` which includes the Wi-Fi settings, and `/var/lib/bluego/` which contains information about connected Bluetooth devices. Hence, the user data is left unchanged, while the system partition is completely rewritten.

4.1.3. Hardware-based Acquisition

While being forensically sound, hardware-based acquisition usually entails the risk of damaging the hardware which might lead to an unusable device or even data loss. Further, on devices with encrypted user data, hardware-based acquisition is only practical if the encryption key can also be obtained. Our analysis showed no signs of storage encryption on the Nyon 2014, making hardware-based acquisition generally conceivable. But since less intrusive OS-based acquisition was possible, no hardware-based methods were executed. For cases, where non-invasive acquisition is not applicable, as for the Nyon 2021, we present a hardware-based acquisition in Section 5.1.3.

4.2. Data Evaluation

After obtaining root access via Telnet, we were able to browse the entire file system of the Nyon 2014 to identify forensically relevant data. Unsurprisingly, despite being a rather exotic distribution, the file system structure of "Poky" (cf. Section 2.1) follows the common Linux file system hierarchy.

Main Application. We found that a single executable called `Main`—stored in a directory of the same name—is responsible for all the functionalities and the user interface of the Nyon 2014. The binaries folder and its files are located in the user's home directory (`/home/appdata/`) which also contains most of the artifacts described below.

User. Personal information about the user is contained in the `Main/Apps/Settings/<USERID>/userObject.json` file located in the home directory. An excerpt of the JSON file is shown in Listing 1 which reveals, for instance, the user's name, email address, gender, date of birth, address information, and social media accounts.

```
{
    "user_id":                "1234567890123",
    "date_of_birth":          "2000-01-01",
    "first_name":             "Jane",
    "last_name":              "Doe",
    "gender":                 "female",
    "email":                  "janedoe@example.com",
    "home_address":           { [...] },
    "mobile_phone_number":    "[...]",
    "facebook":               null,
    "twitter":                null,
    [...]
}
```

Listing 1: `userObject.json` file contents

Cycling Statistics. The SQLite database file `EBikeCharts` contains a single table named **ChartsData** whose purpose is seemingly to store statistical data for diagrams. However, we only found occurrences of the file name in log files (yet). Each row is associated with a "UserId" (cf. Listing 1). The other columns store information related to the trip ("NTDistance", "Altitude"), the cyclist ("Speed", "driverCadence", "heartRate"), and the battery ("stateOfCharge", "Consumption", "Power"). As no timestamps are indicated in the table, the precise logging interval is unknown. However, since the values in the "NTDistance" column increase by steps of 25 (e.g., 66175 → 66200), it may be cautiously assumed that logging occurs each 25 meters when cycling.

Cycling Activities. Another SQLite database called `EBike` contains a total of seven tables whose rows are all associated with a "UserId" (cf. Listing 1) and "TimeStamp" (epoch):

The table **Activities** stores information related to, for instance, trips (e.g., "StopTime", "Distance"), hardware components of the bike (e.g., "DriveUnitSerial", "BatteryPackSerials"), or fitness and driving performance (e.g., "Calories", "Max. Speed"). The ambient temperature and air pressure are logged in the table **AmbientData**, and the table **BikeBattery** provides historical data on the battery's "stateOfCharge". Regarding the electric drive of the eBike, the table **DriveUnit** logs whether it "isMoving", the status of its "odometer", the present bike speed, and the associated torque, revolution, and power of the motor. Additionally, the table **Operational** logs whether the "BUI" (cf. product name in Section 2.1) was operational at given timestamps. Driver-related information can be found in the table **Driver**, such as "heartRate", "driverCadence", "driverTorque", or "driverPower". Finally, the table **Localization** contains the GPS coordinates of recent cycling activities indicated by latitude and longitude, as well as the corresponding "SensorAltitude". We noticed that trip data is deleted from the table as soon as it is synchronized with the Bosch cloud. In practice, the table may therefore contain geodata of multiple past trips if no synchronization occurred, or none at all.

Planned Trips. GPX files are stored in the folder `Main/Apps/Settings/<USERID>/gpx/` within the home directory. The parent directory is named after the user's ID (cf. Listing 1). These "GPS Exchange Format" files contain the waypoints of routes the user planned using the companion app or website (cf. Section 2.3), indicated by longitude and latitude values in an XML schema.

Bike Information. The `Settings.ini` file in `Main/Apps/Settings/<USERID>/` stores technical details about the user's eBike, such as part and serial numbers of components, or the Wi-Fi access token. Furthermore, the file includes timestamps that indicate at which point in time the recording of fitness and geo data were allowed, and the date of the last sync (encoded as Qt `@Variant` string).

Connectivity. As mentioned in Section 4.1.2, the two directories `/var/lib/connman/` and `/var/lib/bluego/` were saved and restored during the update process. The former contains a subdirectory for each Wi-Fi connection that has been established, including a `settings` file revealing the SSID, passphrase, network settings, and timestamp. The latter stores a file for every Bluetooth device, including general information and the device's name.

System Logs. Detailed system information can be found in the system's log files, which are located in the `/home/appdata/var/log/` directory. The scope of these log files covers, for instance, Bluetooth and Wi-Fi, system messages, USB diagnostics, and shutdown events.

Tampering. In principle, if obtaining unrestricted access to the OS like we did (cf. Section 4.1.2), data can be manipulated by anybody with physical access to the Nyon 2014. In an experiment, we forged a trip by adding additional GPS points and timestamps to the `EBike` database file. As expected, the trip was accepted by the device without any problems. Moreover, it was also synchronized with Bosch's cloud and thereby displayed in both Bosch's companion app and web service (cf. Section 2.3). Even though it might take some effort to forge a realistic trip with, for instance, reasonable altitude and distance values as well as timestamps, tampering with data on the Nyon 2014 is feasible if enough time and expertise is given.

5. Forensic Analysis of the Nyon 2021

We now analyze the Nyon 2021, the second-generation device in the series of bike computers that is still the most recent Nyon at the time of writing. Again, the acquisition methods presented in the Sections 3.1–3.3 are evaluated, whereby the most suitable method is executed. Afterwards, the acquired data is analyzed to identify relevant traces.

5.1. Data Acquisition

According to our acquisition methodology, we first evaluate manual acquisition, where we highlight the differences and similarities to the Nyon 2014 (cf. Section 4.1.1) before trying to acquire the actual data from the device. For the first-generation Nyon, we were able to extract the data logically by using a design flaw in the update process (cf. Section 4.1.2), which had a low footprint and enabled us to acquire the entire data set without risking damage to the device. Therefore, OS-based acquisition is considered for the Nyon 2021 as well, before the more intrusive hardware-based acquisition is evaluated.

5.1.1. Manual Acquisition

Similar to the first-generation Nyon (cf. Section 4.1.1), not much information can be gained through the user interface of the Nyon 2021: The settings only provide the user's name and email address, and the navigation screen only displays recently typed addresses. Regarding the latter, neither a timestamp nor an indication on whether those destinations were actually visited or only searched is visible.

A minor difference to the Nyon 2014 is that only destinations are displayed which have been typed in by the user or have been synchronized from the companion app or web service. Hence, the Nyon 2021 does not display destinations that were started from the app directly.

5.1.2. OS-based Acquisition

Similar to its predecessor, the Nyon 2021 can be connected to a PC for debugging purposes or installing updates via Bosch's DiagnosticTool. In contrast to the Nyon 2014, however, we were not able to interfere with the update process anymore. Apparently, the design flaw reported in Section 4.1.2 was recognized and fixed by Bosch for the second-generation Nyon. In the revised update process, the Nyon 2021 is now being recognized as a storage medium on the PC, and the DiagnosticTool just copies over an encrypted update file. This file contains the encrypted file system image ("SquashFS") which gets decrypted on the device itself, leaving us without a race-window in the process.

As known from Android devices, we identified the debug mode of the Nyon 2021 by clicking 17 times on "SW-Version" in the settings menu. After activating the switch labeled "Debug", the Nyon 2021 was connected to a PC and recognized as a "RNDIS" device, i.e., as a USB network device ("Remote Network Driver Interface Specification"). Accordingly, a new network device with the IP address "172.16.35.101" and an open port "5001" was recognized. In Windows, the new device is called "Bosch Service Bridge", which seems to be a debug service. The AppArmor file for the "Diagnosticsservices" binary, which seems to be the counterpart on the device, suggests that it can be used to spawn a shell. Since the usage of this mechanism for data acquisition would require time-consuming reverse engineering and possibly exploitation, the feasibility remains unknown.

In summary, the OS-based acquisition revealed no possibility to access the Nyon 2021. In accordance with our acquisition methodology, we therefore opted for a more intrusive hardware-based acquisition, as explained below.

5.1.3. *Hardware-based Acquisition*

As a last resort, hardware-based methods were executed to acquire data from the Nyon 2021. Since it is the least destructive solution, we first attempted access over potential hardware debug interfaces, followed by a destructive chip-off procedure to obtain and read the eMMC chip.

Active JTAG or UART debug interfaces are useful to obtain data from IoT devices (Gordon et al., 2019). In our case, a possible hardware debug interface was identified on the MLB and highlighted in Figure 3. We soldered wires to the pre-tinned pads, and connected them to a logic analyzer and JTAGulator. As the pads only showed either constant or no supply voltages, this approach led to no result.

After all non-destructive approaches in our methodology were exhausted, we executed a chip-off. The basic idea of this method is to remove a chip from the printed circuit board (PCB) by liquifying the solder through heat, and extracting the data using special adapters suitable for the respective chip type. However, as described by Fukami et al. (2017), this procedure might lead to data loss because of the high temperatures that the chip is exposed to, and surrounding components might also be damaged. Therefore, a reassembly of the device to a working condition is unlikely.

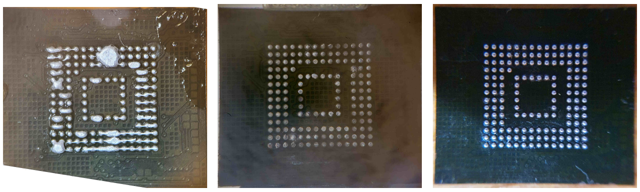
In Figure 7, we show the various stages of the eMMC throughout the chip-off procedure, which were as follows:

1. Before heating the chip using a hot air station, we placed heat-resistant tape around the edges and over the surrounding components to reduce damage. Afterwards, the chip was preheated to 250 °C for about 3 minutes. Generally speaking, the temperature set on the hot air station, the temperature arriving at the chip package, and the temperature reaching the solder underneath the chip, are not identical. Then, while continuously adding flux, the temperature was increased to 380 °C until the solder liquified, and the chip could be removed. As shown in Figure 7a, the chip was covered with excessive solder and flux.

2. The chip was cleaned using a soldering iron and desoldering braid. Contaminants were removed using isopropyl alcohol. The result can be seen in Figure 7b.

3. We reballed the chip using a stencil for the respective BGA-153 socket, soldering paste, and hot air. The result is shown in Figure 7c.

4. Using the Easy JTAG Plus (2.53 rev. 2), an eMMC adapter for the BGA-153 socket, and the software "EasyJtag Classic Suite" (v3.7.0.24), we successfully obtained access to the data on the chip. In total, 8 partitions were extracted, as well as unallocated memory.



(a) After removal (b) Cleaned (c) Reballed

Figure 7: The removed eMMC chip after the chip-off process.

5.2. Data Evaluation

The most important of the extracted partitions are described in Table 1. Besides the boot image, the first partition contained files with partition names, which stored their offset, size, hash, and salt values. Those files are probably used for the "dm-verity" feature which checks the consistency of important system partitions on startup. The names that are provided in the table, helped to attribute the partitions to their purposes. Most of the partitions are not forensically relevant, since they cannot change due to "dm-verity".

Userdata Partition. The unallocated memory extracted from the eMMC turned out to be the LUKS-encrypted userdata partition. The header of the respective dump file stored information about the key material used for encryption, which is 32-bit in size. Since we already identified keys on other partitions, we tried our luck and searched for 32-bit files on all partitions. Hereby, we found a file named `crypto_keyfile.bin` that was successfully used to decrypt and mount the userdata partition. As expected, this partition contained all the forensically relevant data in two directories, namely `system/` and `users/`. A selection is presented below.

Known Wireless Networks. Previously connected Wi-Fi networks are stored in the `WifiManagerSettings.json` file, incl. their names and passwords in plain text (see Listing 2).

```
{
    "id":       "\"Galaxy Note10+0c95\"",
    "psk":      "[PLAINTEXT_PASSWORD]",
    "security": "WPA2"
}
```

Listing 2: `WifiManagerSettings.json` (Excerpt, Nyon 2021).

Last Known Location. The last known location, indicated by latitude, longitude, altitude, speed, and timestamp, can be obtained from the `gnssSettings.json` file (see Listing 3).

```
"lastPosition":{
    "latitude":     [REDACTED],
    "longitude":    [REDACTED],
    "altitude":     311.0,
    "speed":       0.32667222619056702,
    "timestamp":    1686737311649,
    [...]}
```

Listing 3: `gnssSettings.json` (Excerpt, Nyon 2021).

START	SIZE	NAME	INFO
0x400000	16 MB	-	contains files with partitioning scheme and partition names
0x2800000	1 GB	`bui3xx-image`	system partition with the root file system
0x42900000	192 MB	`bui3xx-systemconfig`	log files and system config
0x4ea00000	336 MB	`bui3xx-recovery`	system recovery partition
0x6ac00000	5 GB	-	skobbler map data for navigation, key for userdata partition
0x1ac900000	50 MB	-	contains test images and key material
0x1af900200	550 MB	-	unallocated memory, turned out to be encrypted userdata partition

Table 1
Important partitions with their offset, size, names (according to the partition description files), and content information.

Nearby and Trusted Bluetooth Devices. A log file of the "Chromium Embedded Framework" (CEF), responsible for the user interface, is located in `system/webfs/logs/cef_debug.log` and reveals, among other things, information about discovered and trusted Bluetooth devices.

System Logs. The database `system/db/analytics.db` logs timestamps of various system events in the table **analytics_events**, such as "BUI350_SYSTEM_WAKEUP", "BUI-350_BOOT_INFO", or "BUI350_START_NAVIGATION". If available, related parameters are indicated as well (e.g., "{"duration": "12:36"}" for "BUI350_TRIP_RESET").

The directory `system/webfs/logs/log/` contains several log files related to system activities, such as `system.log` or `wifi.log`. Additionally, the names of certain log files indicate software that is actively running on the Nyon 2021, such as "Tunnelblick" (OpenVPN), "CUPS" (Common Unix Printing System), "Adobe Acrobat Updater", "fsck", or "nginx".

User. All relevant user-specific data can be found in the `users/buiowner/data/system/db/` directory. Similar to the JSON file of the first Nyon generation (`userObject.json`), the `active-account.json` file contains user information, like user ID, name, address, and contact info.

In `user-settings.db`, a total of seven tables exist, but only **settings_app** and **settings_system** contain data. Judging by the values in the "stored_setting" column, the former table contains different information for the application, such as the battery level, or whether the developer mode was activated (cf. Section 5.1.2). The latter table, on the other hand, provides system settings like the language and unit of measurement, but also user-defined goals, such as weekly kilometers, or monthly calories.

Bike Information. The `bike-info.json` file contains a list of all registered eBikes, including their serial and part numbers, software and hardware version numbers, as well as detailed information about their battery packs.

Cycling Activities. The database file `NavStorage.sqlite` stores navigation and location data of the user. While the **Consumptions** table stores vectors indicating the energy consumption for the travel distance calculation, the **Locations** table stores user-defined places indicated by their address ("name"), a tuple of latitude and longitude values, and an epoch timestamp of the last modification. Indicated similarly, the planned routes of users can be found in the **Routes** table, and recent destinations in the **Recents** table.

The contents of another database file, `tracking.db`, are similar to the `EBike` database of the first-generation Nyon, even though they are now spread over 22 tables. More interestingly, we found that data synchronization on the Nyon 2021 seems to not have an impact on the stored data. Contrary to the Nyon 2014, where the data was deleted after synchronization with the Bosch cloud, the second-generation Nyon stored the last 100 trips.

6. Discussion

In general, the data we could acquire on both generations of Nyons was similar, albeit the data set on the latest device was slightly larger and the structure differed.

Forensic Implications. Probably the most obvious forensically interesting data is the information about the trips taken by the user. This information is available comprehensively on both devices, even though trips on the first Nyon are deleted after a synchronization. In particular, the wealth of detailed location data and timestamps is of utmost importance for forensics. In a hypothetical case, it might help to determine if a suspect visited a certain location, and at which time the visit took place.

Furthermore, both devices feature detailed information about the driving behavior of the user. While not being as relevant as the location information, it could help to determine the plausibility of the assumption that a certain person was actually the one who drove this bike. If, for example, a physically demanding cycling style is exhibited, and a fairly non-athletic person is the suspected driver, it could be an indication of a false assumption.

Personal information about the user could also be of interest since not only the name but also contact info and social media accounts are stored. If this data was not previously known or not disclosed, it might be of assistance in drawing a comprehensive picture of a suspect.

The data about wireless networks and Bluetooth devices within proximity can enhance our confidence in ascertaining the locations visited by the device's user. Of particular significance are nearby Bluetooth devices, which can potentially aid in determining whether an individual was actually the driver based on the registration of their phone as a Bluetooth device. Additionally, the examination of stored Wi-Fi passwords offer the option for network analysis.

Due to the unrestricted access to the OS of the first Nyon, we showed that data tampering is possible, which should be considered when interpreting traces. For example, a trip does not have to be completely made up to create a false alibi, because changing the timestamps might be enough.

Advice for Practitioners. From working on this project, we especially noticed many similarities between smartphones and the Nyons. However, some differences require distinct data acquisition methods. Since the security measures were not as advanced, hardware-based data acquisition considered outdated for modern smartphones retained their applicability. The Nyons also do not offer publicly accessible OS interfaces usable for data acquisition, such as shared folders, local backups, or publicly available debug tools.

Chip-off, as a well-known hardware-based option, is applicable on both devices, since the Nyon 2014's data was unencrypted, and the encryption key of the Nyon 2021 was easily accessible. But before actually executing such a destructive method, other options should be considered.

On the software side, manual gathering of data from the user interface did not result in a comprehensive data set, however in some cases this information might suffice. As we could show for the Nyon 2014, a thorough analysis of possible software-based access methods might be worth the effort. Particularly so, since it became possible to avoid more destructive options and evidence about the data's integrity due to the processes execution could be provided.

7. Conclusion and Future Work

We conducted a comprehensive forensic analysis of both existing Bosch Nyon eBike board computers to uncover valuable digital traces with forensic significance. Before acquiring the data, we examined both the hardware and software aspects of the devices. This involved highlighting the essential features by inspecting the main board with its hardware components and interfaces, as well as by analyzing software details and communication protocols.

Subsequently, we formulated a structured data acquisition methodology by first defining important forensic priorities concerning data acquisition and their challenges in the field of mobile forensics. By taking those requirements into account, a structured acquisition strategy was formulated.

Our efforts led to the successful data acquisition for both generations of Nyon eBike computers. For the first-generation Nyon, we acquired root access to the device's operating system, granting us comprehensive access to its data. In contrast, the increased security measures of the second-generation Nyon necessitated using hardware-based methods for data extraction. Due to our inability to access the device through hardware debug interfaces, we resorted to a chip-off procedure, which enabled the extraction of the data from the flash memory. The extracted user data was encrypted, but since the encryption key was stored on an unencrypted partition, decryption was possible.

By analyzing the files from both Nyon devices, we uncovered a wealth of forensically relevant information. While many similarities existed between the data of the two generations, the second-generation Nyon had a slightly broader spectrum of data. On both devices, detailed location records were available, including timestamps, driving behavior profiles, stored Wi-Fi credentials, Bluetooth logs, and system data. The most apparent applicability of this information are statements about a suspect's whereabouts and assessments about the plausibility of the individual's driving behavior.

In conclusion, our forensic analysis of Bosch Nyon eBike computers not only improves our understanding of these special-purpose mobile devices but also highlights their significance in the forensic context. By addressing the inherent challenges and successfully extracting valuable data, our study contributes to the evolving landscape of digital forensics, underscoring the importance of considering such devices in the investigative process.

A possible future direction is the analysis of the app and cloud data from the Nyon devices, as well as an evaluation of a broader spectrum of eBike board computers.

We have initiated a responsible disclosure process regarding our findings and are in contact with Bosch PSIRT.

Acknowledgements

We thank Jenny Ottmann as well as the anonymous reviewers for their helpful comments. This work has been supported partially by the Bavarian Ministry of Science and Arts as part of the project "Security in Everyday Digitization" (ForDaySec) and by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 "Cybercrime and Forensic Computing" (grant number 393541319/GRK2475/1-2019).

References

Barr-Smith, F., Farrant, T., Leonard-Lagarde, B., Rigby, D., Rigby, S., Sibley-Calder, F., 2021. Dead man's switch: forensic autopsy of the nintendo switch, in: Proceedings of the Digital Forensics Research Conference Europe (DFRWS EU).

Buquerin, K.K.G., Corbett, C., Hof, H.J., 2021. A generalized approach to automotive forensics, in: Proceedings of the Digital Forensics Research Conference Europe (DFRWS EU).

Casey, E., 2011. Digital Evidence and Computer Crime - Forensic Science, Computers and the Internet, 3rd Edition. Academic Press. URL: http://www.elsevierdirect.com/product.jsp?isbn=9780123742681.

Casey, E., Turnbull, B., 2011. Digital evidence on mobile devices. Digital Evidence and Computer Crime 3, 1–44.

Ebbers, S., Ising, F., Saatjohann, C., Schinzel, S., 2021. Grand theft app: Digital forensics of vehicle assistant apps, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, pp. 1–6.

Freiling, F.C., Groß, T., Latzo, T., Müller, T., Palutke, R., 2018. Advances in forensic data acquisition. IEEE Des. Test 35, 63–74. URL: https://doi.org/10.1109/MDAT.2018.2862366, doi:10.1109/MDAT.2018.2862366.

Fukami, A., Ghose, S., Luo, Y., Cai, Y., Mutlu, O., 2017. Improving the reliability of chip-off forensic analysis of NAND flash memory devices, in: Proceedings of the Digital Forensics Research Conference Europe (DFRWS EU).

Fukami, A., Stoykova, R., Geradts, Z., 2021. A new model for forensic data extraction from encrypted mobile devices. Forensic Science International: Digital Investigation 38, 301169.

Garfinkel, S.L., 2010. Digital forensics research: The next 10 years. Digit. Investig. 7, S64–S73. URL: https://doi.org/10.1016/j.diin.2010.05.009, doi:10.1016/j.diin.2010.05.009.

Garfinkel, S.L., 2013. Digital media triage with bulk data analysis and bulk_extractor. Comput. Secur. 32, 56–72. URL: https://doi.org/10.1016/j.cose.2012.09.011, doi:10.1016/j.cose.2012.09.011.

Geus, J., Ottmann, J., Freiling, F., 2023. Systematic Evaluation of Forensic Data Acquisition using Smartphone Local Backup, in: Proceedings of the Digital Forensics Research Conference (DFRWS US).

Gómez, J.M.C., Mondéjar, J.C., Gómez, J.R., Martínez, J.L.M., 2021. Developing an iot forensic methodology. A concept proposal, in: Proceedings of the Digital Forensics Research Conference Europe (DFRWS EU).

Gordon, T., Kilgore, E., Wylds, N., Nowatkowski, M., 2019. Hardware reverse engineering tools and techniques, in: 2019 SoutheastCon, IEEE. pp. 1–6.

Hoppe, T., Kuhlmann, S., Kiltz, S., Dittmann, J., 2012. It-forensic automotive investigations on the example of route reconstruction on automotive system and communication data, in: Ortmeier, F., Daniel, P. (Eds.), Computer Safety, Reliability, and Security - 31st International Conference, SAFECOMP 2012, Magdeburg, Germany, September 25-28, 2012. Proceedings, Springer. pp. 125–136. URL: https://doi.org/10.1007/978-3-642-33678-2_11, doi:10.1007/978-3-642-33678-2_11.

Huang, A., 2003. Hacking the Xbox: An Introduction to Reverse Engineering. No Starch Press.

Kioxia Corp., 2019. Datasheet: eMMC THGBMJG6C1LBAIL (8GB). URL: https://datasheet.lcsc.com/lcsc/2004271813_KIOXIA-THGBMJG6C1LBAIL_C524518.pdf. Accessed: 2023-10-09.

Micron Technology, 2018. Datasheet: eMMC MTFC8GACAANA-4M IT (JWB18). URL: https://www.micron.com/products/managed-nand/emmc/part-catalog/mtfc8gacaana-4m-it. Accessed: 2023-10-09.

Nanya Technology, 2015. Datasheet: DDR3(L) SDRAM (NT5CB128M16FP-DII). URL: https://pdf1.alldatasheet.com/datasheet-pdf/view/1132525/NANYA/NT5CB128M16FP-DII/+01__7W8XLzx/1DdSZCIIvwpZDITePuab+/datasheet.pdf. Accessed: 2023-10-09.

Nanya Technology, 2022. Datasheet: DDR3(L) SDRAM (NT5CC256N16ER-EKI). URL: https://static6.arrow.com/aropdfconversion/54ca98a15fba86b32611a8fd857f21d97cc3239f/4gb_ddr3_e_die_component_datasheet.pdf. Accessed: 2023-10-09.

NXP Semiconductors, 2018. Datasheet: i.MX 6Solo CPU (MCIMX6S5EVM10AB). URL: https://www.nxp.com/docs/en/data-sheet/IMX6SDLCEC.pdf. Accessed: 2023-10-09.

Robert Bosch GmbH, 2023a. Bosch eBike Connect. URL: https://www.ebike-connect.com/. Accessed: 2023-10-01.

Robert Bosch GmbH, 2023b. Bosch eBike Systems Licences Products. URL: https://www.bosch-ebike.com/de/licences-products. Accessed: 2023-10-01.

Robert Bosch GmbH, 2023c. Index of /data/DiagnosisSoftware/Update. URL: http://bosch-ebike-updates.com/data/DiagnosisSoftware/Update/. accessed: 2023-10-01.

Strandberg, K., Nowdehi, N., Olovsson, T., 2023. A systematic literature review on automotive digital forensics: Challenges, technical solutions and data collection. IEEE Trans. Intell. Veh. 8, 1350–1367. URL: https://doi.org/10.1109/TIV.2022.3188340, doi:10.1109/TIV.2022.3188340.

TheRoundup.org, 2023. 51 Official Ebike Statistics & Facts. URL: https://theroundup.org/ebike-statistics/. Accessed: 2023-10-10.

Villarreal, A.M., Verma, R.K., Upton, O., Beebe, N.L., 2022. Nondestructive data acquisition methodology for iot devices: A case study on amazon echo dot version 2. IEEE Internet of Things Journal 10, 4375–4387.

Yocto Project, 2023. Poky. URL: https://www.yoctoproject.org/software-item/poky/. Accessed: 2023-10-09.

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, in: Proceedings of the Digital Forensics Research Conference (DFRWS APAC).

CRediT authorship contribution statement

Marcel Stachak: Conceptualization, Methodology, Investigation, Writing - Review and Editing. **Julian Geus:** Conceptualization, Methodology, Validation, Investigation, Writing - Original Draft, Writing - Review and Editing, Supervision. **Gaston Pugliese:** Investigation, Resources, Writing - Review and Editing, Supervision. **Felix Freiling:** Writing - Review and Editing, Supervision.