Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile

Guillaume Celosia, Mathieu Cunche

Fingerprinting Bluetooth-Low-Energy Devices Based on the Generic Attribute Profile

Guillaume Celosia
Univ Lyon, INSA Lyon, Inria, CITI
F-69621 Villeurbanne, France
guillaume.celosia@insa-lyon.fr

Mathieu Cunche
Univ Lyon, INSA Lyon, Inria, CITI
F-69621 Villeurbanne, France
mathieu.cunche@insa-lyon.fr

ABSTRACT

Bluetooth Low Energy (BLE) is a short range wireless technology included in many consumer devices such as smartphones, earphones and wristbands. As part of the Attribute (ATT) protocol, discoverable BLE devices expose a data structure called Generic Attribute (GATT) profile that describes supported features using concepts of services and characteristics. This profile can be accessed by any device in range and can expose users to privacy issues.

In this paper, we discuss how the GATT profile can be used to create a fingerprint that can be exploited to circumvent anti-tracking features of the BLE standard (i.e. MAC address randomization). Leveraging a dataset of more than 13000 profiles, we analyze the potential of this fingerprint and show that it can be used to uniquely identify a number of devices. We also shed light on several issues where GATT profiles can be mined to infer sensitive information that can impact privacy of users. Finally, we suggest solutions to mitigate those issues.

CCS CONCEPTS

• **Networks** → **Network privacy and anonymity**; • **Security and privacy** → *Mobile and wireless security*.

KEYWORDS

Privacy; Bluetooth Low Energy; Tracking; Address randomization.

1 INTRODUCTION

Bluetooth is a widespread radio communication standard operating on the 2.4GHz ISM band. Bluetooth Low Energy (BLE) was introduced [21, Vol 6] in 2010 by the Bluetooth Special Interest Group (SIG) as a variant targeted towards battery-powered Internet of Things (IoT) applications such as fitness trackers, headphones and smartwatches. According to the Bluetooth SIG, more than two billion devices supporting BLE have been shipped in 2018 [25].

To protect users against tracking [14], the Bluetooth Core Specification version 4.0 introduced the LE Privacy feature [21, Vol 3, Part C, sec. 10.7] that defines the use of temporary and random link layer identifiers. However, several works [4, 6] have discovered flaws in its implementation showing that devices can still be tracked despite LE Privacy provisions. BLE has also been subjected to attacks aiming to infer sensitive information on users [7, 9].

In this paper, we focus on the Generic Attribute (GATT) profile exposed by connectable BLE devices as part of the mandatory Attribute (ATT) protocol. This profile presents a description of features supported by a device through concepts of services and characteristics. Moreover, as most of its elements are readable without authentication, a GATT profile can be easily collected by any device in range. We show that an attacker can use the content of a GATT profile to compromise the privacy of the device owner through tracking and inference of sensitive information.

Our contributions are outlined as follows:

- Based on a dataset of more than 13000 profiles, we demonstrate that the content of a GATT profile can be leveraged to build a fingerprint that can be used to single-out the device and undermine the LE Privacy provisions (Section 5);
- We identify that some services and characteristics can be exploited to infer sensitive information on the user (Section 6);
- We provide a set of recommendations to mitigate the issues uncovered in this work (Section 7).

Finally, we discuss related work in Section 8, and give concluding remarks in Section 9.

2 BACKGROUND

2.1 BLE protocol

BLE is a radio communication standard [24] operating on the 2.4GHz ISM band. In BLE, devices can endorse two main roles: Central and Peripheral. A Peripheral can be connectable or not depending on if it accepts or not incoming connection requests. A Central device can connect to a connectable Peripheral to communicate with it. For instance, a smartphone can connect to a smartwatch to send notifications and collect sensor readings.

BLE features a discovery mechanism that allows Central devices to discover nearby Peripheral. As part of this mechanism, connectable Peripheral periodically broadcast advertisement packets to announce their presence.

2.2 BLE addressing and privacy

BLE devices are identified by a Bluetooth device address, a 48-bit identifier that can be found within the payload of advertisement packets. As part of its privacy feature (called *LE Privacy*), BLE has introduced random addresses in addition to the globally unique MAC address [1, sec. 8.2]. Thus, there are 4 types of device address in BLE: Public, Random Static, Random Non-resolvable and Random Resolvable. The Public device address corresponds to the MAC address uniquely allocated to the device by the manufacturer. Other device address types are acting as pseudonyms as they are randomly generated and can change during the lifetime of a device.

Based on their temporal persistence, we classify those device address types into two categories:

- **Stable addresses**: device addresses that are used by a device indefinitely or for an extended period of time (i.e. Public and Random Static addresses);

- **Private addresses**: device addresses that are supposed to change frequently[1] (i.e. Random Non-resolvable and Random Resolvable addresses).

2.3 GATT

In BLE, the Attribute (ATT) protocol is a Client/Server stateless protocol based on *attributes* where devices can endorse each role regardless of their BLE role (Peripheral or Central).

Data exposed by a Server are presented in a GATT profile which is a hierarchical structure of attributes allowing the transfer of information between a Client and a Server. Within a GATT profile, attributes can be either *services* or *characteristics* and are identified by a universally unique identifier (UUID). In the hierarchical structure, conceptually related *characteristics* are grouped below a same *service* (see Figure 1).

In addition to their UUID, *characteristics* are made up of an attribute handle, a set of properties and a value. The handle specifies the position of the characteristic in the profile while the value holds the actual data of the characteristic. Properties are metadata that specify which ATT operations (read, write, etc.) can be executed on each particular attribute and with which specific security requirements (encryption, authentication).

A *service* is identified by its UUID and is associated with two handles (*Handle Start* and *Handle End*) that specify a range of *characteristics* that are hierarchically dependent from this service.

Vendors are free to define their own services and characteristics, but the Bluetooth SIG has already defined a number of them [26]. For instance, the Bluetooth SIG has defined the `Device Information` service that contains the `Model Number String`, `Software Revision String` and `System ID` characteristics.

The BLE protocol features security mechanisms such as encryption and authentication that can be used in the ATT protocol. As such, certain values of characteristics may only be accessed by an authenticated Client [24, Vol 3, Part C, sec. 10.3]. Note that, the value is the only element protected by this feature; the list of services and characteristics as well as the associated metadata do not require authentication to be accessed.

3 METHODOLOGY

Our study is based on a dataset of BLE GATT profiles from 13295 distinct Bluetooth device addresses collected over 5 months by the authors during commute, work and leisure times. This dataset, presented in Table 2 and Table 3, was divided into two parts depending on the nature of the address used by devices (Stable or Private). Devices using Private addresses may be observed several times under a different pseudonym. Thus, in the Private part of the dataset, the number of actual devices is expected to be smaller than the reported number of distinct device addresses.

This dataset was collected using a Raspberry Pi 3 single-board computer equipped with four CSR v4.0 Bluetooth USB dongles. One of those dongles continuously scans for advertising Peripheral using the `bluepy` [13] python library. The 3 remaining dongles try to connect to discovered connectable Peripheral prior to enumerate attributes of their GATT profiles using our custom multi-threaded

[1]The Bluetooth Core Specification [24, Vol 3, Part C, App. A] recommends to renew Random Non-resolvable and Random Resolvable addresses at most every 15 minutes.

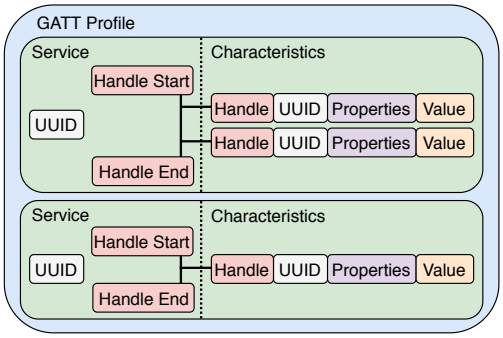


Figure 1: Structure of a GATT profile. Services are composed of a UUID along with two handles (`Handle Start` and `Handle End`) delimiting the hierarchically dependent characteristics. Characteristics are each constituted of a handle, a UUID, a set of properties and a value containing data.

version of the `bleah` [15] python tool. Each GATT profile is then structured as a json string and stored for further analysis.

Dataset anonymization: To limit the privacy risks associated with the collected data, we applied modifications to prevent user re-identification. We focused on attributes corresponding to identifiers (Stable device addresses, device names, etc.) and on temporal data (timestamps). The 24 least significant bits (the NIC) of Stable addresses have been pseudonymized through keyed-hashing[2]. String identifiers potentially containing names of users were sanitized by searching and key-hashing substrings that were matching names. Finally, the temporal information has been transformed from absolute (date and time) to relative (time elapsed since the beginning of the collection campaign).

4 ATTACKER MODEL

We consider an active attacker which monitors the BLE advertising channels to detect nearby connectable Peripheral, connect to them and collect their GATT profiles using several ATT *Read By Type Request*. Furthermore, we assume that the device used by the attacker has not been paired with any Peripheral: it cannot authenticate itself and access protected values of characteristics. As described in Section 3, those assumptions can be satisfied using off-the-shelf hardware and open-source software. In addition, we found that a full GATT profile can be collected in a matter of seconds (see Table 4). On the target side, we assume that the device has its BLE interface turned on, is in communication range and is discoverable.

Based on the collected GATT profile, the attacker can have two objectives: 1) generate a fingerprint of the device in order to track it despite its address randomization scheme and 2) infer sensitive information on the device and its owner.

5 GATT PROFILES FINGERPRINTING

Following the approach of Vanhoef et al. [28], we study how much identifying information can be found in GATT profiles. In particular, we study how services and characteristics can be used to create a fingerprint of the device. In case this fingerprint is unique enough, it can be used to track a device despite the address randomization.

[2]The key used during this process has been erased.

5.1 GATT fingerprint artifacts

The GATT profile of a BLE device is a data structure that can be easily accessed and that includes a number of data elements that can be used for fingerprinting. First, the number of possible components is large: online GATT specifications [26] describe a list of 40 services and 226 characteristics that can be complemented by vendors with their own custom elements. In total, we found 263 distinct services and 1086 distinct characteristics in the dataset. In addition, characteristics are associated with a value that can contain up to 512 bytes of data [24, Vol 3, Part F, sec. 3.2.9]. All those elements are also accompanied by metadata: handles and properties respectively represented by 2 bytes and 8 flags.

Second, the content of a profile will vary depending on the device type as the GATT profile reflects the features and the characteristics of the device. For instance, the `Cycling Power Measurement` characteristic will be only included in cycle devices, while only weight scales will expose the `Weight Scale Feature` characteristic.

Finally, values associated with characteristics may vary from one device to another as they can reflect the device state or identity. For instance, this is the case of identifiers such as the `Device Name` and `Model Number String`.

Overall, a GATT profile is a data structure containing a large number of elements that are subject to variation between devices, and thus hold a potential for fingerprinting.

To create the fingerprint of a BLE device, we considered the following artifacts:

- List of *services*, including for each service:
  - **Handles (start-end)**: the handle range associated with the service (two 16-bit identifiers)
  - **UUID**: the UUID associated with the service (128-bit identifier)
- List of *characteristics*, including for each characteristic:
  - **Handle**: the handle associated with the characteristic (16-bit identifier)
  - **UUID**: the UUID associated with the characteristic (128-bit identifier)
  - **Properties**: the properties of the characteristic (8 bits)
  - **Value**: the value of the characteristic (from 0 to 512 bytes)

5.2 Fingerprinting evaluation

We use the collected dataset to evaluate the fingerprinting potential offered by the GATT profiles. Previously presented artifacts were extracted from the dataset and stored in a database in which each fingerprint is associated with a device address and a timestamp. Then, the resulting database was processed to compute fingerprinting metrics: entropy (Section 5.3) and anonymity sets (Section 5.4).

Impact of random addresses on evaluation: The dataset includes records from devices using random addresses (Private addresses). A device using the address randomization scheme can be observed multiple times under different pseudonyms and thus the corresponding fingerprint will be counted multiple times instead of one. This overcounting will have an impact on the privacy metrics: the entropy will be reduced and the size of the anonymity set will be increased. Therefore, values reported for the Private part of the dataset should be considered as an underestimation of the fingerprinting potential.

5.3 Empirical entropy

Leveraging the dataset, we evaluate the quantity of information brought by the services and characteristics. The entropy is a metric used to measure the amount of identifying information brought by an element of the fingerprint [8]. The database of fingerprints was processed to compute an empirical evaluation of the entropy of each artifact i using the following formula:

$$H_i = -\sum_{j \in E_i} f_{i,j} * \log_2 f_{i,j} \quad (1)$$

where E_i is the domain of possible values for an artifact i and $f_{i,j}$ is the frequency (i.e. probability) of the value j for the artifact i in the dataset. Note that, the absence of an artifact was also considered as a possible value.

Table 1 presents the entropy for the 8 most common services and characteristics exposed in the dataset as well as for the overall profile. The *Entropy* column presents the amount of identifying bits provided by the artifacts. The *Stability* column presents the fraction of devices observed several times for which the value of the artifact is constant throughout the dataset. Finally, the *Affected devices* column presents the fraction of devices that include this artifact in their GATT profiles.

A first observation is the high stability of the fingerprint: the overall fingerprint is stable in more than 95% of the cases. The entropy of single artifacts is typically comprised between 0 and 2 bits. However, some artifacts such as the `Device Name` and the `Model Number String` characteristic can bring up to 3.152 bits of information. Indeed, those artifacts are in fact identifiers.

Variations can be also observed between the types of device address: the `Device Name` brings less information for Private than for Stable addresses. Actually, we observed that for devices using Private addresses, this characteristic is often configured to carry a generic value[3]. This is likely a deliberate choice done for privacy reasons. However, developers appear to have overlooked the `Model Number String` as it appears to be a high source of information for Private addresses (2.757 bits).

Overall, characteristics appear to bring more information than services (4.380 bits against 2.111 bits). This is explained by the fact that characteristics hold more artifacts than services. When considering the full fingerprint, which includes both the characteristics and the services, we can observe that the entropy is the same as with the characteristics alone. This is due to the fact that artifacts of a service (handles and UUID) are fully determined by artifacts of its characteristics (remind that characteristics are hierarchically dependent from services). In other words, services do not bring additional information with regard to characteristics.

5.4 Anonymity sets

To further study the fingerprinting potential of GATT profiles, we used the concept of *anonymity set*, which is defined as a set of entities that share the same fingerprint. From a privacy point of view, the larger the anonymity set the better.

Aided by the `kmap` [11] python tool, we computed the anonymity sets for the fingerprints contained in the dataset. Figures 2a and 2b

[3]For instance, the value of the `Device Name` characteristic is `iPhone` for both an *Apple iPhone 6* and an *Apple iPhone 8* smartphone.

Table 1: Empirical entropy computed from the dataset for services and characteristics exposed within GATT profiles. For each item: the entropy brought by the attribute, the percentage of devices for which this item is stable over time, and the percentage of devices that include this item in their GATT profiles.

		Entropy (bits)			Stability* (%)			Affected devices (%)		
		All	Stable	Private	All	Stable	Private	All	Stable addr.	Private addr.
Services <Hdle Start, Hdle End, UUID>	Generic Access	0.750	0.606	0.245	100	100	100	99.01	92.99	100
	Generic Attribute	0.728	0.560	0.248	100	100	100	97.34	81.35	99.97
	Apple Continuity	0.680	0.317	0.462	100	100	100	84.74	6.48	97.64
	Apple Nearby Service	0.720	0.306	0.499	100	100	100	84.32	4.36	97.50
	Device Information	1.425	0.551	1.037	100	100	100	69.74	55.90	72.02
	Battery Service	0.943	0.328	0.879	100	100	100	57.86	5.58	66.48
	Current Time Service	0.871	0.277	0.866	100	100	100	57.08	0.05	66.48
	Apple Media Service	0.835	0.277	0.831	100	–	100	57.07	0	66.48
	Overall	2.111	0.803	1.330	99.28	100	99.20	–	–	–
Characteristics <Hdle, UUID, Prop, Value>	Device Name	1.913	1.191	0.731	100	100	100	99.65	97.56	100
	Appearance	1.148	0.625	0.578	100	100	100	98.90	92.40	99.97
	Service Changed	0.766	0.566	0.290	100	100	100	97.34	81.35	99.97
	Apple Continuity	0.680	0.317	0.462	100	100	100	84.74	6.48	97.64
	Apple Nearby	0.720	0.306	0.499	100	100	100	84.32	4.36	97.50
	Manuf. Name String	1.422	0.538	1.053	99.82	100	99.81	69.38	53.40	72.02
	Model Number String	3.152	0.564	2.757	99.82	100	99.81	69.32	52.98	72.02
	Battery Level	1.020	0.395	0.879	100	100	100	58.65	11.16	66.48
	Overall	4.380	1.294	3.092	97.84	95.71	98.08	–	–	–
	Overall (services + characteristics)	4.380	1.294	3.092	97.84	95.71	98.08	–	–	–

* Stability values have been computed only from device addresses that we observed multiple times.

respectively show the distributions of the set sizes for Stable and Private addresses.

For Stable addresses, the anonymity sets are small with 94.75% of sets of size 1, meaning that those devices can be uniquely identified by their fingerprints. This is not critical, as those devices can be already identified through their Stable addresses. However, it demonstrates the potential for unique identification based on the GATT profile.

Moving to Private addresses, we observe that a smaller number of devices are uniquely identifiable (4.28%) and that 74.33% of devices are in anonymity sets of size 100 or more. This improvement could be explained by the fact that vendors have reduced the amount of identifying information included in GATT profiles of devices using Private addresses.

Focusing on devices using Private addresses, we found that a large number of them are *Apple iPhone* smartphones[4]. By dividing the Private part of the dataset between non-iPhone devices (Figure 2c) and *iPhones* (Figure 2d), we found that a majority of *iPhones* were sharing their fingerprints with many other devices: 85.49% are in anonymity sets of size 100 or more. On the other end, non-iPhone devices using Private addresses have less common fingerprints as 74.38% of them are in anonymity sets of size 10 or less, and 32.09% of them are unique.

A possible explanation to this phenomenon is that *Apple* distributes a large number of devices but focused on a small number of models (a single line of products with few variants per generation). Furthermore, the software running on those devices is homogeneous. It seems that a side effect of *Apple* commercial and technical policies is to reduce possibilities of uniquely identifying their devices based on technical characteristics.

6 INFERRING INFORMATION FROM GATT PROFILES

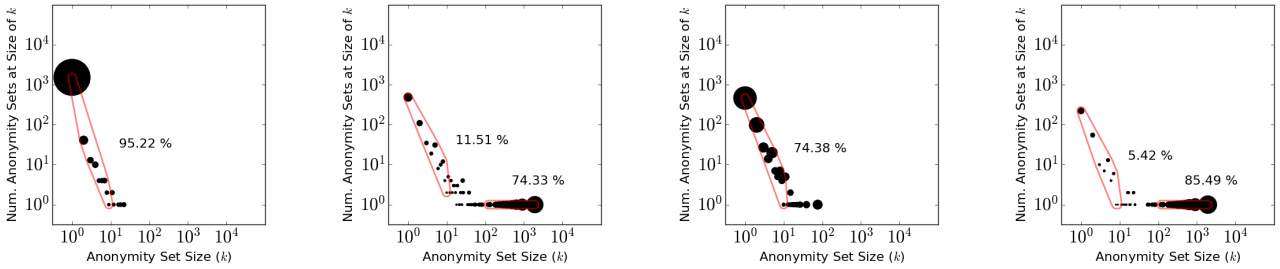
The content of a GATT profile can be leveraged to infer information on the device and its user. In this section, we present a number of elements found in GATT profiles that can be used to infer potentially sensitive information. In particular, we found that the value of characteristics was a rich source of information and was often readable without authentication (see Table 9).

We identified that information found in GATT profiles can be used to infer the following information: **device type**, **device model**, **device manufacturer** and **user's name**. All this information can threaten the privacy of the device owner. Information on the device model can lead to inventory attacks [9], and user's name can reveal the identity of the owner. Furthermore, we found that the value of some characteristics can hold identifiers, which can be leveraged for tracking despite the device address randomization.

6.1 Human readable identifiers

GATT profiles can include characteristics for which the value is a human readable string. For instance, this is the case of the `Device Name`, `Model Number String` and `Manufacturer Name String`. The `Device Name` characteristic is available and readable in more than 99% of the profiles, and often includes names of *manufacturer*,

[4]This device model identification is based on the values of the `Model Number String` and `Manufacturer Name String` characteristics along with the presence of *Apple* specific services (`Apple Continuity Service`, `Apple Nearby Service`, etc.).



(a) Stable addr. (1882 devices) **(b) Private addr. (11413 devices)** **(c) Private without *iPhones* (1489 devices)** **(d) Private *iPhones* only (9924 devices)**

Figure 2: Anonymity sets of GATT profiles in the dataset. The dot size is proportional to the number of devices in the set.

model and *user* such as `Polar M400` and `Alice's MacBook Pro`. Similarly, values carried by the `Model Number String` explicitly identify the device *model* [2]. For instance, the model number of an *Apple iPhone 8* is `iPhone10,4` while `iPad8,3` indicates a 11-inch *Apple iPad Pro*. Finally, the `Manufacturer Name String` directly reveals the *manufacturer* of the device.

6.2 Digital identifiers

Serial number strings: The `Serial Number String` characteristic carries a variable-length utf-8 string representing the serial number for a particular instance of the device. Leveraging the dataset, we found that the format of this identifier is often specific to a vendor[5] and thus can be leveraged to infer the *manufacturer*.

System IDs: The value of the `System ID` characteristic is a 64-bit structure which consists of a 40-bit manufacturer-defined identifier concatenated to a 24-bit Organizationally Unique Identifier (OUI). By definition, this OUI is issued by the IEEE Registration Authority to companies and thus can reveal the *manufacturer*. For instance, manufacturers such as *Xiaomi* and *Amazfit* include their OUI within the value of their `System ID` characteristic.

PnP IDs: The `PnP ID` characteristic carries a set of values that are used to create a unique identifier for the device. Included in this characteristic is a *Vendor ID Source*, a *Vendor ID*, a *Product ID* and a *Product Version* field. The *Vendor ID Source* specifies the type of the *Vendor ID* value: a company identifier assigned by the Bluetooth SIG [23] or a value assigned by the USB Implementers Forum [10]. We observed that manufacturers such as *Gigaset* (company ID `0x0180`) and *Freebox* (USB ID `0x10eb`) include an identifiable PnP ID in their GATT profiles. Thus, the PnP ID can reveal the *manufacturer* but also the device *model* and *software version*. For instance, Product IDs of *Fitbit Surge* and *Fitbit Charge* are respectively `0x0010` and `0x0013` while the Product Version of the *Bose SoundSport Free* earphones is `0x0132` corresponding to the value carried by its `Software Revision String` (`1.3.2`).

6.3 Enumerated type values

Some characteristics are associated with enumerated values whose meanings are specified by the GATT specifications [26].

[5]For instance, *Ultimate Ears* and *Bose* respectively code their serial numbers following the `"^1...LZ0....800$"` and `"^07....[Z,P][6,7,8]"` regular expressions.

Appearances: The `Appearance` characteristic represents information about the external appearance of the device. Such a characteristic is readable in more than 99% of devices and can provide a broad description (e.g. `Generic Tag`) as well as a more specific one (e.g. `Running Walking Sensor: On-Shoe`). For instance, appearance values of *Apple TV* and *Garmin Forerunner 230* devices are respectively `Generic Media Player` and `Watch: Sports Watch`. Moreover, as reported in [6], certain appearance values indicate specific medical devices such as an `Insulin Pen` that could trivially betray a medical condition of the owner.

Sensor locations: The `Body Sensor Location` and `Sensor Location` characteristics are indicating where the device is located on the user. As such, they provide clues to infer the *type* of device: the presence of one of those characteristics indicates that the device is a sensor, and its value can further specify the type of sensor (see Table 5 and Table 6 for respectively a list of body sensor location and sensor location values).

6.4 Measurement values

During our study, we discovered that *Mio Global Alpha 2* smartwatches expose a readable value within their `Running Speed and Cadence (RSC) Measurement` characteristic revealing the physical activity of the user (i.e. walking or running). As a result, such a characteristic can constitute an additional source of information that could be used to profile or physically identify a user [7].

6.5 Names of services and characteristics

Beyond values carried by characteristics, names of services and characteristics can be leveraged as an indicator to reveal both the device *type* and *manufacturer*. For instance, the `Cycling Speed and Cadence (CSC) Feature` characteristic will be only included in cycle devices, while the presence of the `RSC Measurement` characteristic denotes running sensors. In addition, we found that UUID of attributes exposed within GATT profiles can be customized by manufacturers. Leveraging online specifications and codes such as the *Apple Notification Center Service* [3] and *Xiaomi Mi Band 2* [27] ones, it becomes possible to uncover meanings of custom UUIDs disclosing the corresponding *manufacturer* at the same time.

7 RECOMMENDATIONS

In light of the presented issues, we provide recommendations that should be considered by manufacturers of BLE devices, but also by the Bluetooth SIG to improve the Bluetooth Core Specification.

Restricting access to values of characteristics: We found that a number of characteristics are readable for unauthenticated Clients which expose the device to fingerprinting and inference of sensitive information. In many cases, it is not clear why this information is left openly available. A simple solution is to use the permission system of GATT to ensure that those values can only be read by authenticated Clients. This mechanism should be set by default on all characteristics and its removal should be justified by a valid requirement.

To support this statement, we leveraged the dataset to simulate the adoption of this measure. All values of characteristics were removed at the exception of mandatory ones (values of the `Device Name`, `Appearance` and `Service Changed` characteristics) and the entropy was re-evaluated (see Table 8). Removing those values has a significant impact on the fingerprinting potential: the overall entropy goes down from 4.380 to 2.765 bits.

In cases where this solution cannot be adopted, values should be as general as possible. For instance, the value of the `Model Number String` characteristic within *Apple* devices could be just `Apple` or `iOS` instead of `iPhone10,4` or `iPad8,3` (as reported by [16]).

Minimizing exposure of GATT profile: The current version of Bluetooth Core Specification [24, Vol 3, Part G, sec. 8.1] specifies the following: *"the list of services and characteristics that a device supports is not considered private or confidential information, and therefore the service and characteristic discovery procedures shall always be permitted"*. We demonstrated that this is not the case: even if values are not readable, the list of services and characteristics available in a GATT profile can be used for fingerprinting and inference of sensitive information. A potential mitigation technique would be to minimize the exposure of the GATT profile. This could be done by setting access control properties to services and characteristics so that only authenticated Clients can access them. By default, an unauthenticated Client will only see basic services and characteristics (e.g. `Generic Attribute` and `Service Changed`), and only authenticated Clients will be able to see the full list.

8 RELATED WORK

The possibility of singling-out a device based on its technical characteristics and attributes has been explored by several works. This technique has been used in the context of Web browser in order to track users despite anti-tracking mechanisms [8, 12, 20]. In the context of wireless devices, fingerprinting has been used to identify technical characteristics of a device such as the version of the operating system or driver [5], as well as the device model [16]. The timing of frame transmissions has been leveraged in 802.11 networks to identify a device [19] and to track it over time [18]. In [17, 28], the authors have demonstrated that the content of 802.11 probe request frames can be used to fingerprint devices and defeat MAC address randomization. Our work shows that this problem is not limited to 802.11 and that BLE suffers from similar issues.

Recent works [4, 6, 16] have studied the advertisement mechanism of BLE and showed that the content of advertisement packets can be used to defeat address randomization. Our work shows that address randomization is also threatened by the content of GATT profiles.

Beyond tracking issues, BLE can be at the source of private information leakages. Das et al. demonstrated [7] how the BLE traffic from a fitness tracker can leak the physical activity of the wearer. In [9], Fawaz et al. discussed how information exposed in BLE advertisement packets can be leveraged as side information to infer sensitive attributes such as a medical condition. In this paper, we continue this research direction by presenting a detailed list of information leakages affecting BLE devices, that we illustrate with examples extracted from a large real-world dataset.

9 CONCLUSION

In this work, we studied data exposed within GATT profiles of BLE devices. First, we demonstrated that the content of a GATT profile can be leveraged to fingerprint a device. Indeed, identifiers and values composing this profile are diverse enough to act as a fingerprint. This fingerprint can be used to track a device even if it uses anti-tracking mechanisms such as the MAC address randomization. This contribution complements recent works [4, 6, 16, 17, 28] that have demonstrated the difficulties in the implementation of device address randomization.

Then, we showed how the data exposed in GATT profiles can be mined to infer information on the device such as its type, model, manufacturer and software version, which can be leveraged to threaten privacy of users.

To the best of our knowledge, there is no indication that those techniques are currently exploited in the wild. In the Web ecosystem, introduction of anti-tracking techniques has triggered [20] the deployment of fingerprinting techniques by Web trackers. The recent adoption of address randomization in wireless technologies [16, 28] could trigger a similar move by the industry of physical tracking. As a consequence, it is important to take this threat into account as soon as possible, by reviewing and complementing the Bluetooth Core Specification with additional requirements.

ACKNOWLEDGEMENTS

This work was supported by the SPIE ICS-INSA Lyon IoT chair and the SPARTA project. The authors would like to thanks Alexis Duque for his valuable insight on BLE.

REFERENCES

[1] 2014. IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture. *IEEE Std 802-2014 (Revision to IEEE Std 802-2001)* (June 2014), 1–74. https://doi.org/10.1109/IEEESTD.2014.6847097

[2] 2019. Models - The iPhone Wiki. (2019). https://www.theiphonewiki.com/wiki/Models Accessed: 2019-07-04.

[3] Apple. 2014. Apple Notification Center Service (ANCS) Specification. (2014). https://developer.apple.com/library/archive/documentation/CoreBluetooth/Reference/AppleNotificationCenterServiceSpecification/Specification/Specification.html#//apple_ref/doc/uid/TP40013460-CH1-SW8 Accessed: 2019-07-04.

[4] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking Anonymized Bluetooth Devices. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (2019), 50–65.

[5] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. 2008. Active Behavioral Fingerprinting of Wireless Devices. In *Proceedings of the First ACM*

Conference on Wireless Network Security (WiSec '08). ACM, New York, NY, USA, 56–61. https://doi.org/10.1145/1352533.1352543

[6] Guillaume Celosia and Mathieu Cunche. 2019. Saving Private Addresses: An Analysis of Privacy Issues in the Bluetooth-Low-Energy Advertising Mechanism. In *Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM.

[7] Aveek K Das, Parth H Pathak, Chen-Nee Chuah, and Prasant Mohapatra. 2016. Uncovering privacy leakage in BLE network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. ACM, 99–104.

[8] Peter Eckersley. 2010. How unique is your web browser ?. In *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 1–18.

[9] Kassem Fawaz, Kyu-Han Kim, and Kang G Shin. 2016. Protecting Privacy of BLE Device Users. In *USENIX Security Symposium*. 1205–1221.

[10] USB Implementers Forum. 2019. Membership Lookup & List. (2019). https://www.usb.org/members Accessed: 2019-07-04.

[11] Gabor Gyorgy Gulyas, Gergely Acs, and Claude Castelluccia. 2016. Near-optimal fingerprinting with constraints. *Proceedings on Privacy Enhancing Technologies* 2016, 4 (2016), 470–487.

[12] Gabor Gyorgy Gulyas, Doliere Francis Some, Nataliia Bielova, and Claude Castelluccia. 2018. To Extend or Not to Extend: On the Uniqueness of Browser Extensions and Web Logins. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society (WPES'18)*. ACM, New York, NY, USA, 14–27. https://doi.org/10.1145/3267323.3268959

[13] Ian Harvey. 2014. bluepy - a Bluetooth LE interface for Python. https://ianharvey.github.io/bluepy-doc/index.html Accessed: 2019-07-04.

[14] Markus Jakobsson and Susanne Wetzel. 2001. Security Weaknesses in Bluetooth. In *Topics in Cryptology - CT-RSA 2001 (Lecture Notes in Computer Science)*, David Naccache (Ed.). Springer Berlin Heidelberg, 176–191.

[15] Simone Margaritelli. 2017. This Is Not a Post About BLE, Introducing BLEAH. https://www.evilsocket.net/2017/09/23/This-is-not-a-post-about-BLE-introducing-BLEAH/ Accessed: 2019-07-04.

[16] Jeremy Martin, Douglas Alpuche, Kristina Bodeman, Lamont Brown, Ellis Fenske, Lucas Foppe, Travis Mayberry, Erik Rye, Brandon Sipes, and Sam Teplov. 2019. Handoff All Your Privacy – A Review of Apple's Bluetooth Low Energy Continuity Protocol. *Proceedings on Privacy Enhancing Technologies* 2019, 4 (2019), 34–53.

[17] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye, and Dane Brown. 2017. A study of MAC address randomization in mobile devices and when it fails. *Proceedings on Privacy Enhancing Technologies* 2017, 4 (2017), 365–383.

[18] Celestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. 2016. Defeating MAC Address Randomization Through Timing Attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16)*. ACM, New York, NY, USA, 15–20. https://doi.org/10.1145/2939918.2939930 event-place: Darmstadt, Germany.

[19] Christoph Neumann, Olivier Heen, and Stephane Onno. 2012. An empirical study of passive 802.11 device fingerprinting. In *2012 32nd International Conference on Distributed Computing Systems Workshops*. IEEE, 593–602.

[20] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. 2013. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. In *2013 IEEE Symposium on Security and Privacy*. 541–555. https://doi.org/10.1109/SP.2013.43

[21] Bluetooth SIG. 2010. *Bluetooth Core Specification v4.0*. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=456433 Accessed: 2019-07-04.

[22] Bluetooth SIG. 2017. *GATT Specification Supplement v1.0*. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=429632 Accessed: 2019-07-04.

[23] Bluetooth SIG. 2019. Assigned Numbers - Company Identifiers. (2019). https://www.bluetooth.com/specifications/assigned-numbers/company-identifiers/ Accessed: 2019-07-04.

[24] Bluetooth SIG. 2019. *Bluetooth Core Specification v5.1*. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080 Accessed: 2019-07-04.

[25] Bluetooth SIG. 2019. *Bluetooth Market Update 2019*. Technical Report. https://www.bluetooth.com/wp-content/uploads/2018/04/2019-Bluetooth-Market-Update.pdf Accessed: 2019-07-04.

[26] Bluetooth SIG. 2019. GATT Specifications. (2019). https://www.bluetooth.com/specifications/gatt/ Accessed: 2019-07-04.

[27] Leo Soares. 2019. MiBand2 - Python library to work with Xiaomi MiBand 2. (2019). https://github.com/creotiv/MiBand2/blob/master/constants.py Accessed: 2019-07-04.

[28] Mathy Vanhoef, Celestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. 2016. Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. ACM, New York, NY, USA, 413–424. https://doi.org/10.1145/2897845.2897883

A APPENDIX

Table 2: Composition of the dataset of GATT profiles for each type of device address.

		Stable		Private	
		Public	Static	Non-res.	Res.
Device addr.	%	10.29	3.87	0.17	85.67
	#	1368	514	23	11390

Table 3: Top 10 services and characteristics in the dataset.

		Device addr. (%)		
		All	Stable	Private
Services	Generic Access	99.01	92.99	100
	Generic Attribute	97.34	81.35	99.97
	Apple Continuity	84.74	6.48	97.64
	Apple Nearby Service	84.32	4.36	97.50
	Device Information	69.74	55.90	72.02
	Battery Service	57.86	5.58	66.48
	Current Time Service	57.08	0.05	66.48
	Apple Media Service	57.07	0	66.48
	Apple NCS Service	57.07	0	66.48
	ISSC Transparent	3.86	27.26	0
	Overall*	99.17	94.16	100
Characteristics	Device Name	99.65	97.56	100
	Appearance	98.90	92.40	99.97
	Service Changed	97.34	81.35	99.97
	Apple Continuity	84.74	6.48	97.64
	Apple Nearby	84.32	4.36	97.50
	Manuf. Name String	69.38	53.40	72.02
	Model Number String	69.32	52.98	72.02
	Battery Level	58.65	11.16	66.48
	Current Time	57.30	1.65	66.48
	Apple MS Entity Att.	57.07	0	66.48
	Overall*	99.96	99.73	100

* Devices that include at least one of the top 10 service or characteristic.

Table 4: Average time to collect a GATT profile among different devices.

Device type	Device	Time (sec)
Lightbulb	Osram Smart+	6.531
Motion sensor	Eve Motion	6.468
Socket outlet	Eve Energy	5.919
Smartphone	Apple iPhone 8	4.354
Smartphone	Apple iPhone 6	4.259
Keyring	Nut	4.148
TV dongle	Google Chromecast	3.660
Fitness wristband	Fitbit Inspire	3.231
Presentation remote	Logitech Spotlight	2.860
Smartwatch	Apple Watch Series 3	2.853
Heart rate monitor	Polar H7	2.751
Fitness wristband	Fitbit Flex	2.552
Headset	Bose SoundLink Around-Ear II	2.181
Speaker	Divacore Ktulu2+	1.742
Keyring	Chipolo	1.426
	Average	3.662

Table 5: List of `Body Sensor Location` values extracted from [22, sec. 3.24.2.1].

Value	Description
0x01	Chest
0x02	Wrist
0x03	Finger
0x04	Hand
0x05	Ear Lobe
0x06	Foot

Table 6: List of `Sensor Location` values extracted from [22, sec. 3.148.2.1].

Value	Description
0x00	Other
0x01	Top of shoe
0x02	In shoe
0x03	Hip
0x04	Front Wheel
0x05	Left Crank
0x06	Right Crank
0x07	Left Pedal
0x08	Right Pedal
0x09	Front Hub
0x0a	Rear Dropout
0x0b	Chainstay
0x0c	Rear Wheel
0x0d	Rear Hub
0x0e	Chest
0x0f	Spider
0x10	Chain Ring

Table 7: Example of a GATT profile collected from an *Apple iPhone 8* smartphone.

Handles	Service > Characteristics (*UUID*)	Properties	Value
0001 -> 0005	Generic Access (*00001800-0000-1000-8000-00805f9b34fb*)		
0003	Device Name (*00002a00-0000-1000-8000-00805f9b34fb*)	Read	iPhone
0005	Appearance (*00002a01-0000-1000-8000-00805f9b34fb*)	Read	Generic Phone
0006 -> 0009	Generic Attribute (*00001801-0000-1000-8000-00805f9b34fb*)		
0008	Service Changed (*00002a05-0000-1000-8000-00805f9b34fb*)	Indicate	–
000a -> 000e	Apple Continuity Service (*d0611e78-bbb4-4591-a5f8-487910ae4366*)		
000c	Continuity Characteristic (*8667556c-9a37-4c91-84ed-54ee27d90049*)	Notify, Write, Extended properties	–
000f -> 0013	Apple Nearby Service (*9fa480e0-4967-4542-9390-d343dc5d04ae*)		
00011	Nearby Characteristic (*af0badb1-5b99-43cd-917a-a77bc549e3cc*)	Notify, Write, Extended properties	–
0014 -> 0017	Battery Service (*0000180f-0000-1000-8000-00805f9b34fb*)		
00016	Battery Level (*00002a19-0000-1000-8000-00805f9b34fb*)	Notify, Read	–
0018 -> 001d	Current Time Service (*00001805-0000-1000-8000-00805f9b34fb*)		
001a	Current Time (*00002a2b-0000-1000-8000-00805f9b34fb*)	Notify, Read	–
001d	Local Time Information (*00002a0f-0000-1000-8000-00805f9b34fb*)	Read	–
001e -> 0022	Device Information (*0000180a-0000-1000-8000-00805f9b34fb*)		
0020	Manufacturer Name String (*00002a29-0000-1000-8000-00805f9b34fb*)	Read	Apple Inc.
0022	Model Number String (*00002a24-0000-1000-8000-00805f9b34fb*)	Read	iPhone10,4
0023 -> 002c	Apple Notification Center Service (*7905f431-b5ce-4e99-a40f-4b1e122d00d0*)		
0025	Control Point (*69d1d8f3-45e1-49a8-9821-9bbdfdaad9d9*)	Write, Extended properties	–
0028	Notification Source (*9fbf120d-6301-42d9-8c58-25e699a21dbd*)	Notify	–
002b	Data Source (*22eac6e9-24d6-4bb5-be44-b36ace7c7bfb*)	Notify	–
002d -> 0038	Apple Media Service (*89d3502b-0f36-433a-8ef4-c502ad55f8dc*)		
002f	Remote Command (*9b3c81d8-57b1-4a8a-b8df-0e56f7ca51c2*)	Notify, Write, Extended properties	–
0033	Entity Update (*2f7cabce-808d-411f-9a0c-bb92ba96c102*)	Notify, Write, Extended properties	–
0037	Entity Attribute (*c6b2f38c-23ab-46d8-a6ab-a3a870bbd5d7*)	Read, Write, Extended properties	–

Table 8: Empirical entropy of characteristics exposed within GATT profiles in the dataset, without the values (only handles, UUIDs and properties are considered).

		Entropy (bits)		
		All	Stable addr.	Private addr.
Characteristics <Hdle, UUID, Prop>	Device Name (+ value*)	1.913	1.191	0.731
	Appearance (+ value*)	1.148	0.625	0.578
	Service Changed (+ value*)	0.766	0.566	0.290
	Apple Continuity	0.680	0.317	0.462
	Apple Nearby	0.720	0.306	0.499
	Manuf. Name String	1.372	0.517	1.031
	Model Number String	1.327	0.491	1.030
	Battery Level	0.985	0.361	0.879
	Overall	2.764	1.208	1.564
	Overall (services + characteristics)	2.765	1.208	1.564

* Value of this characteristic is kept during the entropy computation because the Bluetooth Core Specification specifies it as *mandatory*.

Table 9: List of Bluetooth SIG-defined characteristics values that have been observed as readable at least once in the dataset. Percentages reported for readable values are fractions of devices for which this value was readable.

	Readable values					
	All		Stable		Private	
	%	#	%	#	%	#
Device Name	99.49	13182	99.18	1821	99.54	11361
Appearance	99.48	13082	98.56	1714	99.62	11368
Service Changed	0.02	2	0.13	2	0	0
Manufacturer Name String	99.48	9177	99.40	999	99.49	8178
Model Number String	99.36	9158	99.40	991	99.36	8167
Battery Level	2.45	191	90.95	191	0	0
Current Time	0.41	31	100	31	0	0
Peripheral Preferred Connection Parameters	99.90	1051	99.90	1037	100	14
Software Revision String	97.04	1017	97.68	1010	50	7
Hardware Revision String	95.95	996	96.58	989	50	7
Serial Number String	97.12	979	97.79	972	50	7
Firmware Revision String	94.99	835	95.72	828	50	7
System ID	97.51	822	98.31	815	50	7
PnP ID	98.02	741	98.02	741	0	0
Peripheral Privacy Flag	99.47	561	100	561	0	0
IEEE 11073-20601 Regulatory Certification Data List	99.64	550	99.64	550	0	0
Reconnection Address	0.56	3	0.56	3	0	0
Central Address Resolution	98.08	307	98.84	85	97.80	222
HTTP Entity Body	100	42	0	0	100	42
Body Sensor Location	70	28	100	28	0	0
Alert Level	7.50	3	8.57	3	0	0
Alert Notification Control Point	100	30	100	30	0	0
Heart Rate Control Point	3.33	1	3.33	1	0	0
Report	14.29	2	14.29	2	0	0
URI	100	14	0	0	100	14
Report Map	42.86	6	42.86	6	0	0
Uncertainty	7.14	1	0	0	7.14	1
HID Information	42.86	6	42.86	6	0	0
Altitude	100	14	0	0	100	14
HTTP Headers	100	14	0	0	100	14
Location Name	100	14	0	0	100	14
RSC Feature	100	9	100	9	0	0
RSC Measurement	11.11	1	11.11	1	0	0
Sensor Location	100	8	100	8	0	0
Protocol Mode	25.00	2	25.00	2	0	0
Tx Power Level	60	3	100	3	0	0
Resolvable Private Address Only	100	4	100	4	0	0
Scan Refresh	33.33	1	33.33	1	0	0
Boot Mouse Input Report	50	1	50	1	0	0
CSC Feature	100	1	100	1	0	0