Workgroup: Network Working Group
Internet-Draft:
draft-detecting-unwanted-location-trackers-01
Published: 20 December 2023
Intended Status: Informational
Expires: 22 June 2024
Authors: B. Ledvina (Apple) Z. Eddinger (Google) B. Detwiler (Apple) S. P. Polatkan (Google)

Detecting Unwanted Location Trackers

Abstract

This document lists a set of best practices and protocols for accessory manufacturers whose products have built-in location-tracking capabilities. By following these requirements and recommendations, a location-tracking accessory will be compatible with unwanted tracking detection and alerts on mobile platforms. This is an important capability for improving the privacy and safety of individuals in the circumstance that those accessories are used to track their location without their knowledge or consent.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-detecting-unwanted-location-trackers/.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 June 2024.

Table of Contents

1. Introduction

This document’s goal is to, in part, help protect the privacy of individuals from unwanted tracking by location-tracking accessories. Location-tracking accessories provide numerous benefits to consumers, but, as with all technology, it is possible for them to be misused. Misuse of location-tracking accessories can result in unwanted tracking of individuals or items for nefarious purposes such as stalking, harassment, and theft. This document is focused on

protecting people from misuse of location-tracking accessories. Formalizing a set of best practices for manufacturers will allow for scalable compatibility with unwanted tracking detection technologies on various smartphone platforms and improve privacy and security for individuals.

Unwanted tracking detection can both detect and alert individuals that a location tracker separated from the owner's device is traveling with them, as well as provide means to find and disable the tracker. This document outlines technical best practices for location tracker manufacturers, which will allow for their compatibility with unwanted tracking detection and alerting technology on platforms.

1.1. Conventions and Definitions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

Throughout this document, these terms have specific meanings:

* The term platform is used to refer to mobile device hardware and associated operating system. Examples of mobile devices are phones, tablets, laptops, etc.

* The term accessory is used to refer to any product intended to interface with a platform through the means described in this specification.

* The term owner device is a device that is associated with the accessory and can retrieve the accessory’s location.

* The term non-owner device refers to a device that may connect to an accessory but is not an owner device of that accessory.

* The term location-tracking accessory refers to any accessory that has location-tracking capabilities, including, but not limited to, crowd-sourced location, GPS/GNSS location, WiFi location, cell location, etc., and provides the location information back to the owner of the accessory via the internet, cellular connection, etc.

* The term location-enabled state refers to the state an accessory in where its location can be remotely viewed by its owner

* The term location-enabled advertisement payload refers to the Bluetooth (BT) advertisement payload that is advertised when an

accessory has recently, is currently, or will in the future provide location updates to its owner

*The term unwanted tracking (UT) refers to undesired tracking of a person, their property, or their belongings by a location-enabled accessory.

*The term unwanted tracking detection refers to the algorithms that detect the presence of an unknown accessory traveling with a person over time.

*The term unwanted tracking alert refers to notifying the user of the presence of an unrecognized accessory that may be traveling with them over time and allows them to take various actions, including playing a sound on the accessory if it’s in Bluetooth Low Energy (LE) range.

*The term platform-compatible method refers to a method of communication between the platform and the accessory/accessory manufacturers to exchange information, including, but not limited to, BT GATT protocol, BT advertisement, HTTP, etc.

2. Background

2.1. Applicability

These best practices are **REQUIRED** for location-enabled accessories that are small and not easily discoverable. For large accessories, such as a bicycle, these best practices are **RECOMMENDED**.

Accessories are considered easily discoverable if they meet one of the following criteria:

*The item is larger than 30 cm in at least one dimension.

*The item is larger than 18 cm x 13 cm in two of its dimensions.

*The item is larger than 250 cm^3 in three-dimensional space.

3. Requirements

3.1. Overview

This section details requirements and recommendations for best practices for location-enabled accessory manufacturers to allow unwanted tracking detection by platform makers.

3.2. Bluetooth Low Energy

The accessory **SHALL** use Bluetooth Low Energy (LE) as the transport protocol. This enables platforms to detect and connect to accessories.

3.2.1. Advertising

The accessory **SHALL** advertise using Bluetooth LE.

3.2.2. Connection

The accessory **MUST** support at least one non-owner unencrypted connection in a peripheral role. The connection interval of the Bluetooth LE link between the device and accessory **MAY** depend on the type of user interaction. Non-owner connections to the accessory **SHALL** be implemented using a platform-compatible method, e.g., BT GATT service.

3.3. Location Tracking

The location-enabled accessory has location capabilities via Bluetooth crowd-sourcing, GPS/GNSS location, WiFi location, cellular location, or by some other means. Furthermore, the accessory has a way to communicate its location to its owner via a network (e.g., cell network, crowd-sourced location via Bluetooth, etc.).

The accessory **SHALL** maintain an internal state that detects when its location is, or has been, available to the owner via a network. This state is called the location-enabled state.

Misuse of location-enabled accessories can occur when the owner’s device is not physically with the accessory. Thereby, the accessory **SHOULD** maintain a second internal state, denoted the near-owner state, which indicates if the accessory is connected to or nearby one or more of the owner’s devices. Near-owner state can take on two values, either near-owner mode or separated mode. Near-owner mode is denoted as the opposite of separated mode.

Figure 1 details the requirements and recommendations for advertising the location-enabled payload based on the location-enabled state and separated state.

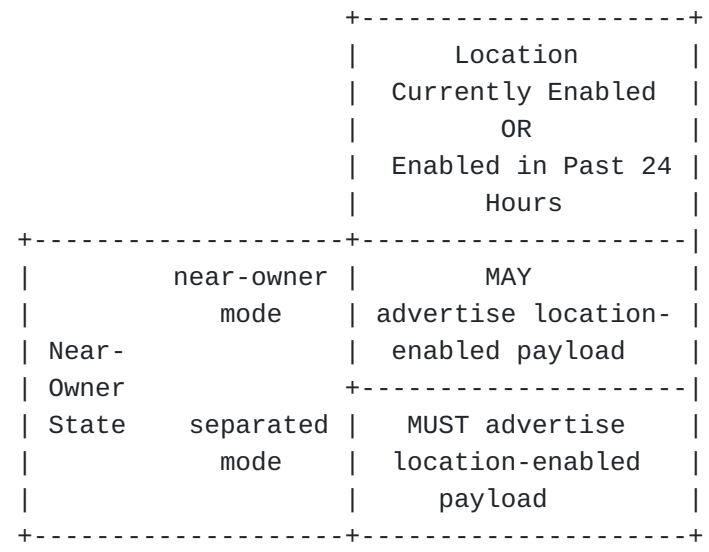

Figure 1: Requirements & Recommendations For Advertising Location-Enabled Payload

If the accessory maker chooses to continue advertising the location-enabled payload while in near-owner mode, setting the near-owner bit (Section 3.8) compensates for this.

3.4. Location-enabled Bluetooth LE Advertisement Payload

3.4.1. Overview

When in location-enabled state, the accessory **SHALL** advertise a Bluetooth LE format, denoted the location-enabled Bluetooth advertisement payload, that is recognizable to the platforms.

3.4.2. Location-enabled advertisement payload format

The payload format is defined in Table 1

Bytes	Description	Requirement
0-5	MAC address	**REQUIRED**
6-8	Flags TLV; length = 1 byte, type = 1 byte, value = 1 byte	**OPTIONAL**
9-12	Service Data TLV; length = 1 byte, type = 0x16, value = 0xFCB2	**REQUIRED**
13	Network ID	**REQUIRED**
14	Near-owner bit (1 bit, least significant bit) + reserved (7 bits)	**REQUIRED**
15-36	Proprietary company payload data	**OPTIONAL**

Table 1: Location-Enabled Payload Format

When Flags TLV are not added, the MAC address type needs to be set to random. This implies that if Bluetooth LE pairing is supported, the accessory **SHALL NOT** use its public address as its public identity when exchanging pairing keys at phase 3 (see Vol.3, Part H, Section 2.1 of the [BTCore5.4]) and it **SHALL** only use a static random address. Additionally, the LE advertisement needs to be connectable to allow for non-owner unencrypted connections to the accessory. Further details are discussed in Section 3.10.

Proprietary company payload data is both **OPTIONAL** and variable length.

3.4.3. Duration of advertising location-enabled advertisement payload

The accessory **SHALL** broadcast the location-enabled advertisement payload if location is available to the owner or was available any time within the past 24 hours. This allows unwanted tracking detection to operate both between and beyond the specific moments an accessory's location is made available to the owner.

3.4.4. Maximum duration after physical separation from owner to transition into separated mode

The accessory **SHALL** transition from near-owner mode to separated mode under the conditions listed in Table 2 below.

Preferred	Acceptable
The accessory has been physically separated from the owner device for more than 30 minutes	The accessory has been physically separated from the owner device for more than 30 minutes **AND** The owner of the accessory has received a more recent location update for that accessory after 30 minutes

Table 2: Advertising Policy

3.4.5. Maximum duration after reunification with owner to transition into near-owner mode

The accessory **SHALL** transition from separated to near-owner mode if it has reunited with the owner device for a duration no longer than 30 minutes.

3.5. MAC address

The Bluetooth LE advertisement payload **SHALL** contain an address in the 6-byte Bluetooth MAC address field which looks random to all parties while being recognizable by the owner device. The address type **SHALL** be set as a non-resolvable private address or as a static device address, as defined in Random Device Address in Vol 6, Part B, Section 1.3.2 of the [BTCore5.4].

The owner **MUST** be able to predict the MAC address or the payload advertised by the accessory at any given time in order to suppress unwanted tracking alerts caused by a device's owned accessory. See Owned Accessory Identification (Section 6.1) for additional details.

The address **SHALL** rotate periodically (see Rotation policy (Section 3.5.1)); otherwise if the same address is used for long periods of time, an adversary may be able to track a legitimate person carrying the accessory through local Bluetooth LE scanning devices. Same rules apply to any other identifiable content of the advertised payload.

A general approach to generate addresses meeting this requirement these properties is to construct them using a Pseudo-Random Function (PRF) taking as input a key established during the association of the accessory and either a counter or coarse notion of time. The counter or coarse notion of time allows for the address to change periodically. The key allows the owner devices to predict the sequence of addresses for the purposes of recognizing its associated accessories.

This construction allows accessories to define their own MAC address generation process while also providing a means to meet the requirements for owned accessory identification (Section 6.1).

3.5.1. Rotation policy

An accessory **SHALL** rotate its address on any transition from near-owner state to separated state as well as any transition from separated state to near-owner state.

When in near-owner state, the accessory **SHALL** rotate its address every 15 minutes. This is a privacy consideration to deter tracking of the accessory by non-owners when it is in physical proximity to the owner.

When in a separated state, the accessory **SHALL** rotate its address every 24 hours. This duration allows a platform's unwanted tracking algorithms to detect that the same accessory is in proximity for some period of time, when the owner is not in physical proximity.

3.6. Service data TLV

The Service data TLV with a 2-byte UUID value of 0xFCB2 provides a way for platforms to easily scan for and detect the location-enabled Bluetooth advertisement.

3.7. Network ID

The 1-byte Network ID **SHALL** be set based on a registered value for the manufacturer, as defined in Manufacturer Network ID Registry (Section 10.1).

3.8. Near-owner bit

It is important to prevent unwanted tracking alerts from occurring when the owner of the accessory is in physical proximity of the accessory, i.e., it is in near-owner mode. In order to allow suppression of unwanted tracking alerts for an accessory advertising the location-enabled advertisement with the owner nearby, the accessory **MUST** set the near-owner bit to be 1 when the near-owner state is in near-owner mode, otherwise the bit is set to 0. Table 3 specifies the values of this bit.

The near-owner bit **MUST** be the least significant bit.

Near-owner Bit Value	Near-owner state
1	Near-owner mode
0	Separated mode

Table 3: Near-Owner Bit

3.9. Bluetooth LE advertising interval

The detection rate performance has a dependency on the BLE advertising interval used. A maximum advertising interval of 4 seconds **SHALL** be used; for the best detection rate, the advertising interval **SHOULD** be less than or equal to 2 seconds.

3.10. Accessory Connections

The accessory non-owner service UUID **SHALL** be 15190001-12F4-C226-88ED-2AC5579F2A85. This service **SHALL** use GATT over LE. The non-owner accessory service **SHALL** be instantiated as a primary service. The accessory non-owner characteristic UUID **SHALL** be 8E0C0001-1D68-FB92-BF61-48377421680E.

3.10.1. Byte transmission order

The characteristic used within this service **SHALL** be transmitted with the least significant octet first (that is, little endian).

3.10.2. Maximum transmission unit

Data fragmentation and reassembly is not defined in this document; therefore, the accessory **SHALL NOT** request an MTU (Maximum Transmission Unit) smaller than the maximum length of its write

responses for the opcodes defined in Non-owner controls (Section 3.12.4) and Section 3.11.1. In other words, all opcode response data must fit within a single write operation.

3.11. Accessory Information

The following accessory information **MUST** be persistent through the lifetime of the accessory: Product data (Section 3.11.1.1), Manufacturer name (Section 3.11.1.2), Model name (Section 3.11.1.3), Accessory category (Section 3.11.1.4), Accessory capabilities (Section 3.11.1.6), Network ID (Section 3.7), and Battery Type (Section 3.11.1.9).

3.11.1. Opcodes

The opcodes for accessory information are defined in Table 4.

Opcode	Opcode value	Operands	GATT subprocedure	Requirement
Get_Product_Data	0x003	None	Write; To Accessory	**REQUIRED**
Get_Product_Data_ Response	0x803	Product Data (Section 3.11.1.1)	Indications; From Accessory	**REQUIRED**
Get_Manufacturer_ Name	0x004	None	Write; To Accessory	**REQUIRED**
Get_Manufacturer_ Name_Response	0x804	Manufacturer Name (Section 3.11.1.2)	Indications; From Accessory	**REQUIRED**
Get_Model_Name	0x005	None	Write; To Accessory	**REQUIRED**
Get_Model_Name_ Response	0x805	Model Name (Section 3.11.1.3)	Indications; From Accessory	**REQUIRED**
Get_Accessory_ Category	0x006	None	Write; To Accessory	**REQUIRED**
Get_Accessory_ Category_Response	0x806	Accessory Category (Section 3.11.1.4)	Indications; From Accessory	**REQUIRED**
Get_Protocol_ Implementation_Version	0x007	None	Write; To Accessory	**REQUIRED**
Get_Protocol_ Implementation_Version_ Response	0x807	Protocol Implementation Version (Section 3.11.1.5)	Indications; From Accessory	**REQUIRED**
Get_Accessory_ Capabilities	0x008	None	Write; To Accessory	**REQUIRED**
	0x808			**REQUIRED**

Opcode	Opcode value	Operands	GATT subprocedure	Requirement
Get_Accessory_Capabilities_Response		Accessory Capabilities (Section 3.11.1.6)	Indications; From Accessory	
Get_Network_ID	0x009	None	Write; To Accessory	**REQUIRED**
Get_Network_ID_Response	0x809	Network ID (Section 3.7)	Indications; From Accessory	**REQUIRED**
Get_Firmware_Version	0x00A	None	Write; To Accessory	**REQUIRED**
Get_Firmware_Version_Response	0x80A	Firmware version (Section 3.11.1.8)	Indications; From Accessory	**REQUIRED**
Get_Battery_Type	0x00B	None	Write; To Accessory	**OPTIONAL**
Get_Battery_Type_Response	0x80B	Battery Type (Section 3.11.1.9)	Indications; From Accessory	**OPTIONAL**
Get_Battery_Level	0x00C	None	Write; To Accessory	**OPTIONAL**
Get_Battery_Level_Response	0x80C	Battery Level (Section 3.11.1.10)	Indications; From Accessory	**OPTIONAL**
RESERVED	0x00D - 0x05F			
RESERVED (Response)	0x80D - 0x85F			

Table 4: Accessory Information Opcodes

These opcodes **SHALL** be available when the accessory is in separated state. These opcodes **SHALL NOT** be available when the accessory is in the near-owner state. When any opcode is not available, the accessory **SHALL** return the Invalid_command error as the ResponseStatus in Command_Response. If an optional opcode is not available, the accessory **SHALL** return the Invalid_command error as the ResponseStatus in Command_Response. If any opcode value is commanded that is not supported by the accessory, it **SHALL** return the Invalid_command error as the ResponseStatus in the Command_Response. See Command Response (Section 3.12.4.1.1) for details.

Opcodes should be structured as defined below.

Bytes	Description
0-1	Opcode value
2+	Operand

Table 5: Accessory Opcode Structure

3.11.1.1. Product data

The Product Data operand represents an 8-byte value that is intended to serve as a unique identifier for the accessory make and model. This value **SHALL** be determined during the onboarding process (Section 7).

Operand name	Data type	Count	Total Size (Bytes)	Description
Product Data	Uint8	8	8	See Product data (Section 3.11.1.1)

Table 6: Product Data Operand

3.11.1.2. Manufacturer name

The Manufacturer Name operand contains the name of the company whose brand will appear on the accessory, e.g., ”Acme”.

Operand name	Data type	Count	Total Size (Bytes)	Description
Manufacturer Name	UTF-8	64 (maximum)	64 (maximum)	Manufacturer name

Table 7: Manufacturer Name Operand

3.11.1.3. Model name

The Model Name operand contains the manufacturer specific model of the accessory.

Operand name	Data type	Count	Total Size (Bytes)	Description
Model Name	UTF-8	64 (maximum)	64 (maximum)	Model name

Table 8: Model Name Operand

3.11.1.4. Accessory category

The Accessory Category operand describes the category the accessory most closely resembles.

Operand name	Data type	Count	Total Size (Bytes)	Description
Accessory Category	Uint8	8	8	Byte 0: Uint8 value of Accessory Category Value (Section 4) Byte 1-7: Reserved

Table 9: Accessory Category Operand

3.11.1.5. Protocol implementation version

The Protocol Implementation Version operand contains a value indicating an implementation version of these protocols.

Operand name	Data type	Count	Total Size (Bytes)	Description
Protocol Implementation Version	Uint32	1	4	Byte 0 : revision version number Byte 1 : minor version number Byte 2-3 : major version number

Table 10: Protocol Implementation Version Operand

The Major.Minor.Revision value associated with this document is 1.0.0. The equivalent 4-byte value is 0x00010000.

3.11.1.6. Accessory capabilities

The Accessory Capabilities operand enumerates the various capabilities supported on the accessory as defined in Table 11.

Operand name	Data type	Count	Total Size (Bytes)	Description
Accessory Capabilities	Uint32	1	4	Bit 0 : Supports play sound (**REQUIRED**) Bit 1 : Supports motion detector UT Bit 2 : Supports identifier lookup by NFC Bit 3 : Supports identifier lookup by BLE

Table 11: Accessory Capabilities Operand

For example, an accessory supporting play sound, motion detector UT, and identifier look-up over BT will have the value set as 1011 in binary and 11 as Uint32.

3.11.1.7. Network ID

The Network ID operand contains the Network ID for the accessory. This is the same information that's in the BT advertisement header in Table 1.

Operand name	Data type	Count	Total Size (Bytes)	Description
Network ID	Uint8	1	1	Network ID

Table 12: Network ID Operand

3.11.1.8. Firmware version

The Firmware Version describes the current firmware version running on the accessory. The firmware revision string **SHALL** use the x[.y[.z]] format where :

- <x> is the major version number, required.
- <y> is the minor version number, required if it is non zero or if <z> is present.
- <z> is the revision version number, required if non zero.

The firmware revision **MUST** follow these rules:

- <x> is incremented when there is significant change; for example, 1.0.0, 2.0.0, 3.0.0, and so on.
- <y> is incremented when minor changes are introduced, such as 1.1.0, 2.1.0, 3.1.0, and so on.
- <z> is incremented when bug fixes are introduced, such as 1.0.1, 2.0.1, 3.0.1, and so on.
- Subsequent firmware updates can have a lower <y> version only if <x> is incremented.
- Subsequent firmware updates can have a lower <z> version only if <x> or <y> is incremented.

Major version **MUST** not be greater than (2^16 - 1). Minor and revision version **MUST** not be greater than (2^8 - 1). The value **MUST** change after every firmware update.

Operand name	Data type	Count	Total Size (Bytes)	Description
Firmware version	Uint32	1	4	Byte 0 : revision version number Byte 1 : minor version number Byte 2:3 : major version number

Table 13: Firmware Version Operand

As an example, a Major.Minor.Revision value of 1.0.0 has an equivalent 4-byte value of 0x00010000.

3.11.1.9. Battery type

The Battery type operand describes the battery type used in the accessory.

Operand name	Data type	Count	Total Size (Bytes)	Description
Battery Type	Uint8	1	1	0 = Powered 1 = Non-rechargeable battery 2 = Rechargeable battery

Table 14: Battery Type Operand

3.11.1.10. Battery level

The Battery level operand indicates the current battery level.

Operand name	Data type	Count	Total Size (Bytes)	Description
Battery Level	Uint8	1	1	0 = Full 1 = Medium 2 = Low 3 = Critically low

Table 15: Battery Level Operand

3.12. Non-Owner Finding

Once a user has been notified of an unknown accessory traveling with them, it is **REQUIRED** they have the means to physically locate the accessory. This is called non-owner finding of the accessory.

3.12.1. Hardware

This is a description of the **REQUIRED** and **RECOMMENDED** hardware to be incorporated into the accessory to enable non-owner finding.

3.12.2. Motion detector

The accessory **SHOULD** include a motion detector that can detect accessory motion reliably (for example, an accelerometer). If the accessory includes an accelerometer, it **MUST** be configured to detect an orientation change of ±10° along any two axes of the accessory.

3.12.2.1. Implementation

The details in this section apply to those accessories that include a motion detector. Values of the variables referenced are specified in Table 16.

After $T_{SEPARATED\_UT\_TIMEOUT}$ in separated state, the accessory **MUST** enable the motion detector to detect any motion within $T_{SEPARATED\_UT\_SAMPLING\_RATE1}$.

If motion is not detected within the $T_{SEPARATED\_UT\_SAMPLING\_RATE1}$ period, the accessory **MUST** stay in this state until it exits separated state.

If motion is detected within the $T_{SEPARATED\_UT\_SAMPLING\_RATE1}$ the accessory **MUST** play a sound. After first motion is detected, the movement detection period is decreased to $T_{SEPARATED\_UT\_SAMPLING\_RATE2}$. The accessory **MUST** continue to play a sound for every detected motion. The accessory **SHALL** disable the motion detector for $T_{SEPARATED\_UT\_BACKOFF}$ under either of the following conditions:

- Motion has been detected for 20 seconds at $T_{SEPARATED\_UT\_SAMPLING\_RATE2}$ periods.
- Ten sounds are played.

If the accessory is still in separated state at the end of $T_{SEPARATED\_UT\_BACKOFF}$, the UT behavior **MUST** restart.

A Bluetooth LE connection from an associated device **MUST** reset the separated behavior.

Name	Value	Description
$T_{SEPARATED\_UT\_SAMPLING\_RATE1}$	10 seconds	Sampling rate when motion detector is enabled in separated state.

Name	Value	Description
$T_{SEPARATED\_UT\_SAMPLING\_RATE2}$	0.5 seconds	Motion detector sampling rate when movement is detected in separated state.
$T_{SEPARATED\_UT\_BACKOFF}$	6 hours	Period to disable motion detector if accessory is in separated state.
$T_{SEPARATED\_UT\_TIMEOUT}$	random value between 8-24 hours chosen from a uniform distribution	Time span in separated state before enabling motion detector.

Table 16: Motion Detector Time Values

3.12.3. Sound maker

The accessory **MUST** include a sound maker (for example, a speaker) to play sound when in separated state, either periodically or when motion is detected.

It **MUST** also play sound when a non-owner tries to locate the accessory by initiating a play sound command from a non-owner device when the accessory is in range and connectable through Bluetooth LE. The sound maker **MUST** emit a sound with minimum 60 Phon peak loudness as defined by ISO 532-1:2017. The loudness **MUST** be measured in free acoustic space substantially free of obstacles that would affect the pressure measurement. The loudness **MUST** be measured by a calibrated (to the Pascal) free field microphone 25 cm from the accessory suspended in free space.

3.12.4. Non-owner controls

Non-owner controls **SHALL** use the same service and characteristic UUIDs as defined in Accessory Connections (Section 3.10).

These controls allow a non-owner to locate the accessory by playing a sound as well as fetch an encrypted payload used to retrieve the identifier of the device.

These opcodes are defined in Table 17.

Opcode	Opcode value	Operands	GATT subprocedure
Sound_Start	0x300	None	Write; To accessory
Sound_Stop	0x301	None	Write; To accessory

Opcode	Opcode value	Operands	GATT subprocedure
Command_Response	0x302	Command Response (Section 3.12.4.1.1)	Indications; From accessory
Sound_Completed	0x303	None	Indications; From accessory
Get_Identifier	0x404	None	Write; To accessory
Get_Identifier_Response	0x405	Identifier Payload (Section 3.12.4.2)	Indications; From accessory
RESERVED	0x304 - 0x35F		
RESERVED (Response)	0x405 - 0x45F		

Table 17: Non-Owner Controls Opcodes

Sound_Start and Sound_Stop **SHALL** only be available to the platform when the accessory is in the separated state.

In all other states, the accessory **SHALL** return the Invalid_command error as the ResponseStatus in Command_Response.

Get_Identifier **SHALL** only be available when in identifier read state; otherwise, it **MUST** send Command_Response (Section 3.12.4.1.1) with the Invalid_command as the ResponseStatus.

The identifier read state is discussed further in Identifier Payload (Section 3.12.4.2).

3.12.4.1. Play sound

The Sound_Start opcode is used to play sound on the sound maker of the accessory. The sound maker **MUST** play sound for a minimum duration of 5 seconds.

*The accessory **SHALL** confirm the start of the play sound procedure by sending a Command_Response (Section 3.12.4.1.1) with the corresponding CommandOpCode and a ResponseStatus value of Success.

*Once the play sound action is completed, the accessory sends the Sound_Completed message.

*The Sound_Stop opcode is used to stop an ongoing sound request.

*If the sound event is completed or was not initiated by the connected non-owner device, the accessory responds with the Invalid_state ResponseStatus code.

*If the accessory does not support the play sound procedure, it responds with Invalid_command ResponseStatus code.

*If a Sound_Start procedure is initiated when another play sound action is in progress, it rejects with Invalid_state error code.

*The accessory **SHALL** confirm the completion of the stop sound procedure by sending the Sound_Completed message.

3.12.4.1.1. Command Response

There are 2 components of the command response operands: CommandOpCode and ResponseStatus. The CommandOpCode operand indicates the procedure that the accessory is responding to and ResponseStatus operand indicates the status of the response. The accessory **SHALL** respond to any invalid opcode with Command_Response and Invalid_command as the ResponseStatus.

Operand name	Data type	Count	Total Size (Bytes)	Description
CommandOpCode	Uint16	1	2	The control procedure matching this response
ResponseStatus	Uint16	1	2	0x0000 Success 0x0001 Invalid_state 0x0002 Invalid_configuration 0x0003 Invalid_length 0x0004 Invalid_param 0xFFFF Invalid_command

Table 18: Command Response Operands

3.12.4.2. Identifier Payload

The Get_Identifier opcode is used to retrieve identifier lookup payload over Bluetooth LE. To enable this opcode, the accessory **MUST** be in the identifier read state. To enter the identifier read state, a user action on the accessory **MUST** be performed (for example, press and hold a button for 10 seconds). The identifier read state **MUST** be enabled for 5 minutes once the user action on the accessory is successfully performed. When the accessory is in this mode, it **MUST** respond with Get_Identifier_Response opcode and Identifier Payload operand.

Operand name	Data type	Count	Total Size (Bytes)	Description
Identifier Payload	UTF-8	defined by accessory	defined by accessory	The encrypted identifier, encoded as a hex string.

Table 19: Identifier Payload Over Bluetooth

It is **REQUIRED** that the encrypted identifier (which in some cases is the product serial number) be non-identifiable.

If the accessory is not in identifier read state, it **MUST** send Command_Response (Section 3.12.4.1.1) with the Invalid_command as the ResponseStatus. Further considerations for how these operands should be implemented are discussed in Design of encrypted identifier look-up (Section 8.1.1).

3.12.5. Alternate finding hardware

The accessory **SHOULD** provide alternate means to help find it, e.g. by vibrating or flashing lights, via a platform-compatible method. Future versions of this document will consider support for haptics and lights.

3.12.6. Recommended Finding Options

Table 20 lists two **RECOMMENDED** options on the set of technology in an accessory to make it findable.

	Option A	Option B
	Good	Better
Sound maker	X	X
Haptics		X
Lights		X

Table 20: Accessory Finding Hardware Options

3.12.7. Future hardware

Future technologies for finding **MAY** be considered in revisions of this document.

3.13. Disablement

The accessory **SHALL** have a way to be disabled such that its future locations cannot be seen by its owner. Disablement **SHALL** be done via some physical action (e.g., button press, gesture, removal of battery, etc.).

3.13.1. Disablement instructions

The accessory manufacturer **SHALL** provide both a text description of how to disable the accessory as well as a visual depiction (e.g. image, diagram, animation, etc.) that **MUST** be available when the platform is online and OPTIONALLY when offline. Disablement procedure or instructions CAN change with accessory firmware updates. These are provided as part of the onboarding process (Section 7).

3.14. Identification

The accessory **MUST** include a way to uniquely identify it - either via a serial number or other privacy-preserving solution. Guidelines for serial numbers only apply if the accessory supports identification via a serial number.

3.14.1. Serial number identification

If a serial number is available, it **SHALL** be printed and be easily accessible on the accessory. The serial number **MUST** be unique for each product ID.

3.14.2. Identifier retrieval capability

The identifier payload **SHALL** be readable either through NFC tap (see Identifier over NFC (Section 3.14.5)) or Bluetooth LE (see Identifier Retrieval over Bluetooth LE (Section 3.14.3)).

3.14.3. Identifier retrieval over Bluetooth LE

For privacy reasons, accessories that support identifier retrieval for identifiers not included in the advertising packet over Bluetooth LE **MUST** have a physical mechanism, for example, a button, that **SHALL** be required to enable the Get_Identifier opcode, as discussed in Identifier Payload (Section 3.12.4.2).

The accessory manufacturer **SHALL** provide both a text description of how to enable identifier retrieval over Bluetooth LE, as well as a visual depiction (e.g. image, diagram, animation, etc.) that **MUST** be available when the platform is online and OPTIONALLY when offline. The description and visual depiction CAN change with accessory firmware updates. These are provided as part of the onboarding process (Section 7).

3.14.4. Identifier retrieval from a server

For security reasons, the identifier payload returned from an accessory in the paired state **SHALL** be encrypted.

3.14.5. Identifier over NFC

For those accessories that support identifier retrieval over NFC, an associated accessory **SHALL** advertise the encrypted identifier encoded as a hex string. This string **SHALL** be an argument passed to the URL defined in the Product data registry which **SHALL** decrypt the identifier payload and return the identifier of the accessory in a form that can be rendered in the platform's HTML view.

The encrypted identifier when in associated state **SHALL** be an argument passed to this URL and it is **REQUIRED** that any metadata passed be non-identifiable.

3.15. Owner registry

Verifiable identity information of the owner of an accessory at time of association **SHALL** be recorded and associated with the identifier of the accessory, e.g., phone number, email address.

3.15.1. Obfuscated owner information

A limited amount of obfuscated owner information from the owner registry **SHALL** be made available to the platform along with a retrieved identifier. This information **SHALL** be part of the response of the identifier retrieval from a server which can be rendered in a platform's HTML view.

This **MUST** include at least one of the following:

- the last four digits of the owner's telephone number. e.g., (***) ***-5555

- an email address with the first letter of the username and entity visible, as well as the entire extension. e.g., b********@i*****.com

3.15.2. Persistence

The owner registry **SHOULD** be stored for a minimum of 25 days after an owner has unassociated an accessory. After the elapsed period, the data **SHOULD** be deleted.

3.15.3. Availability for law enforcement

Available ownership registry information **SHOULD** be produced in response to a valid law enforcement request seeking information related to the misuse of location-tracking accessories provided that the request is submitted pursuant to defined procedures for obtaining such information. Network providers **SHOULD** define their own

procedures for submission of valid legal requests from law enforcement.

4. Accessory Category Value

Accessory manufacturer's **MUST** pick an accessory category value that closest resembles their physical product. If none of the accessory categories provided in Table 21 match the physical product, Other **MUST** be chosen.

Accessory Category Name	Value
Location Tracker	1
Other	128
Luggage	129
Backpack	130
Jacket	131
Coat	132
Shoes	133
Bike	134
Scooter	135
Stroller	136
Wheelchair	137
Boat	138
Helmet	139
Skateboard	140
Skis	141
Snowboard	142
Surfboard	143
Camera	144
Laptop	145
Watch	146
Flash drive	147
Drone	148
Headphones	149
Earphones	150
Inhaler	151
Sunglasses	152
Handbag	153
Wallet	154
Umbrella	155
Water bottle	156
Tools or tool box	157
Keys	158
Smart case	159
Remote	160
Hat	161

Accessory Category Name	Value
Motorbike	162
Consumer electronic device	163
Apparel	164
Transportation device	165
Sports equipment	166
Personal item	167
Reserved for future use	2-127, 168+

Table 21: Accessory Category Values

5. Firmware Updates

The accessory **SHOULD** have a mechanism for the manufacturer to provide firmware updates.

5.1. Backwards Compatibility

5.1.1. Existing trackers

Existing trackers should be updated on a best-effort basis to implement the protocols and practices outlined above.

5.1.2. Advertisement backwards compatibility

The manufacturer **MAY** continue to use the company’s existing service UUID (Legacy ID) as registered in the Bluetooth SIG until November 1, 2024, after which all manufacturers **MUST** use the unwanted tracking service UUID to be detected for unwanted tracking. Trackers that use the company's Legacy ID **MUST** be able to have their firmware updated. This applies to new or updated trackers and any existing trackers that have the ability to have their firmware updated.

If the manufacturer wishes to use their Legacy ID, the Legacy ID **MUST** be registered with platforms. If a manufacturer is using a Legacy ID, Network ID in Table 1 **MAY** be omitted.

Manufacturers can register their service UUID by reaching out to the listed authors. Backwards compatibility requests must be submitted by January 15, 2024. Manufacturers who have registered their Network IDs will appear in a table below.

If using existing service UUIDs, rather than the unwanted tracking protocol UUID, detection performance might be degraded.

6. Platform Support for Unwanted Tracking

This section details the requirements and recommendations for platforms to be compatible with the accessory protocol behavior described in the document.

6.1. Owned Accessory Identification

Any platform that supports unwanted tracking **SHOULD** also provide the capability to suppress unwanted tracking alerts caused by an accessory associated with the owner device.

If an unwanted tracking alert occurs for an accessory and the platform does not already have the installed capability to prevent this alert for the owner of the accessory, then the platform **SHOULD** explain to the user how those capabilities can be acquired.

6.1.1. Implementation

Unwanted tracking **SHOULD** recognize an accessory associated to that owner device by matching the MAC address advertised as defined in Table 1, or some other part of the payload, against the one(s) expected during that time.

6.1.2. Platform Software Extension

Platforms **SHOULD** implement the owned accessory identification capability as a software extension to its unwanted tracking detection.

Accessory manufacturers **SHALL** provide this set of MAC addresses or matching rules to the platform. This set **MUST** account for the uncertainty involved with the MAC address (Section 3.5).

The Network ID in the advertisement payload, as specified in Table 1, **SHALL** be used to associate an accessory detected with the manufacturer's software extension.

6.1.3. Network Access

Network access **MUST NOT** be required in the moment that the platform performs owned accessory recognition.

6.1.4. Removal

The platform **MUST** delete any local identifying information associated with an accessory if the manufacturer's software is removed or if the platform unassociates from the accessory.

7. Onboarding

Accessory manufacturers **MUST** follow a minimum set of steps for their accessories to be detectable by platforms such as adding their Network ID value to the Manufacturer network ID Registry (Section 10.1).

During onboarding, a product data registry will be created that includes information such as:

* Product Data: an 8-byte string representing a unique identifier for a product. See Product Data (Section 3.11.1.1).

* Disablement Instructions: information on how a user can disable the tracker.

* Identifier Look-up Over Bluetooth Instructions: visual depictions for enabling identifier look-up over Bluetooth LE.

* Identifier Look-up: a method to retrieve the obfuscated owner information and possibly identifier.

* Product Name: a string representing the accessory make and model associated with the Product Data string.

Additional details will follow in 2024 to specify formats for disablement instructions and product images.

7.1. Network providers

Companies that have their own accessory-locating networks will need to create infrastructure to support the scaled retrieval of disablement instructions and product images. Additional information for network providers will be updated in 2024.

8. Security Considerations

8.1. Obfuscated owner information look-up

Obfuscated owner information look-up is required to display important information to users who encounter an unwanted tracking notification. It helps them tie the notification to a specific physical device and recognize the accessory as belonging to a friend or relative. Displaying an identifier (or serial number) may be one method to allow for partial user information look up.

However, the identifier is unique and stable, and the partial user information can further make the accessory identifiable. Therefore, identifier (if used) and obfuscated owner information **SHOULD NOT** be made directly available to any requesting devices. Instead, several security- and privacy-preserving steps **SHOULD** be employed.

The obfuscated owner information and identifier look-up **SHALL** only be available in separated mode for an associated accessory. When requested through any long range wireless interface like Bluetooth, a user action **MUST** be required for the requesting device to access the

obfuscated owner information and identifier. Over NFC, it **MAY** be acceptable to consider the close proximity as intent for this flow.

To uphold privacy and anti-tracking features like the Bluetooth MAC address randomization, the accessory **MUST** only provide non-identifiable data to non-owner requesting devices. One approach is for the accessory to provide encrypted and unlinkable information that only the accessory network service can decrypt. With this approach, the server can employ techniques such as rate limiting and anti-fraud to limit access to the identifier. In addition to being encrypted and unlinkable, the encrypted payload provided by the accessory **SHOULD** be authenticated and protected against replay. The replay protection is to prevent an adversary using a payload captured once to monitor changes to the partial information associated with the accessory, while the authentication prevents an adversary from impersonating any accessory from a single payload.

8.1.1. Design of encrypted identifier look-up

One way to design this encryption is for the accessory to contain a public key for the accessory network server. For every request received by a device nearby, the accessory would use the public key and a public key encryption scheme (ie: RSA-OAEP, ECIES, or HPKE) to encrypt a set of fields including the identifier, a monotonic counter or one time token and a signature covering both the identifier and counter or token. The signature can be either a public key signature or symmetric signature, leveraging a key trusted by the network server which **MAY** be established at manufacturing time or when the user sets up the accessory. Some additional non-identifiable metadata **MAY** be sent along with this encrypted payload, allowing the requesting device to determine which accessory network service to connect to for the decryption, and for the service to know which decryption key and protocol version to use.

9. Privacy Considerations

9.1. Obfuscated owner information

In many circumstances when unwanted tracking occurs, the individual being tracked knows the owner of the location-tracker. By allowing the retrieval of an obfuscated email or phone number when in possession of the accessory, as described in Section 3.15.1, this provides the potential victim with some level of information on the owner, while balancing the privacy of accessory owners in the arbitrary situations where they have separated from those accessories.

9.2. Identifier look-up

An identifier both physically on the device, as well as retrievable over NFC or Bluetooth LE, can aid recourse actions in the case of unwanted tracking. While retrieval of the identifier over NFC implies having physical possession of the accessory, the same conclusion can not be made for Bluetooth given its wireless range. The procedure required for identifier look-up over Bluetooth LE intends to strike a balance between the privacy of the owner and ability to empower potential victims, by requiring both the accessory to be in separated state as well as a physical action be performed to enable the identifier retrieval.

9.3. Location-enabled payload

9.3.1. Stable identifiers

Rotating the mac address of the location-enabled payload, as described in Section 3.5, balances the risk of nefarious stable identifier tracking with the need for unwanted tracking detection. If the address were permanently static, then the accessory would become infinitely trackable for the life of its power source. By requiring rotation, this reduces the risk of a malicious actor having the ability to piece together long stretches of longitudinal data on the whereabouts of an accessory.

9.3.2. Proprietary company payload data

Accessory manufacturers **SHOULD** evaluate the contents of the proprietary company payload data in Table 1 to ensure it does not introduce additional privacy risk through the broadcast of stable identifiers or unencrypted sensitive data.

10. IANA Considerations

Eventually an IANA will create a new registry group called "Unwanted Tracking Protocols (UTP)". This group includes the "Manufacturer Network ID" registry.

10.1. Manufacturer Network ID Registry

New entries are assigned only for values that have received Expert Review, per Section 4.5 of [RFC8126].

An entry in this registry contains the following fields:

* Manufacturer Name: the name of an organization that is producing a location-tracker accessory

*Network ID: a 1-byte value specifying the Network ID associated with the Manufacturer Name

10.1.1. Temporary Registry

Until this an IANA registry is available, the values in this registry are listed in Table 22.

Network ID	Manufacturer
0x00	Reserved
0x01	Apple Inc.
0x02	Google LLC
0xFF	Reserved

Table 22: Manufacturer Registry

11. Normative References

[BTCore5.4] "Bluetooth Core Specification v5.4", 31 January 2023, <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=556599>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

Authors' Addresses

Brent Ledvina
Apple

Email: bledvina@apple.com

Zach Eddinger
Google

Email: zae@google.com

Ben Detwiler

Apple

Email: bdetwiler@apple.com

Siddika Parlak Polatkan
Google

Email: siddikap@google.com