

# EXHIBIT 7

## Application of U.S. Patent No. 8,116,749 to Google Wireless Devices<sup>\*,\*\*</sup>

---

\* The term “Google Wireless Devices” refers to the Google phones, tablets, and laptops identified in Plaintiff’s operative complaint and those identified herein, as well as all other products made, used, sold, offered for sale and/or imported by Google (as defined in the Complaint) or one of Google’s affiliated companies, that have the features shown in this chart, or substantially similar features.

\*\* This claim chart is meant to be illustrative for purposes of meeting Plaintiff’s pleading obligations and should not be construed as limiting or binding.

## US. Patent No. 8,116,749 versus Google Wireless Devices using Find Hub

**1[pre].** A method for exchange of information between one or more applications executing on at least a first wireless device and a second wireless device, the method comprising the steps of:

Google makes, uses, sells, offers to sell, and/or import wireless devices (*e.g.*, phones, tablets, and computers, accessories) (“Google Wireless Devices”) that implement the method described below.

# Pixel. The only phone engineered by Google.



<https://store.google.com/category/phones?hl=en-US&pli=1>

## Connectivity and Location

Wi-Fi 7 (802.11be) with 2.4GHz+5GHz+6GHz, 2x2+2x2 MIMO

Bluetooth® v5.3 with dual antennas for enhanced quality and connection

NFC

Google Cast

Dual Band GNSS

GPS, GLONASS, Galileo

## Network<sup>26</sup>

5G mmWave + Sub 6GHz<sup>27</sup> Model G2YBB

GSM/EDGE: Quad-band (850, 900, 1800, 1900 MHz)

UMTS/HSPA+/HSDPA: Bands 1,2,4,5,6,8,19

LTE: Bands

B1/2/3/4/5/7/8/12/13/14/17/18/19/20/25/26/28/29/30/38/40/41/48/66/71

5G Sub-6<sup>27</sup>: Bands

n1/2/3/5/7/8/12/14/20/25/26/28/29/30/38/40/41/48/66/70/71/77/78

5G mmWave<sup>27</sup>: Bands n258/260/261

eSIM

## Operating System

Launched with Android 14

[https://store.google.com/product/pixel\\_9\\_specs?hl=en-US](https://store.google.com/product/pixel_9_specs?hl=en-US)

Find your phone > Be ready to find a lost Pixel phone

Due to the latest device launch, we expect to receive higher contact volume than normal. To check if your question is already answered, go to the [Pixel Phone Help Center](#).

### Be ready to find a lost Pixel phone

You can set up Find Hub so you're prepared if you lose your phone, tablet, Wear OS watch, headphones, or something that has a tracker tag attached.

If your device is already lost, [learn how to find, secure, or erase it](#).

**Important:** Some of these steps work only on Android 9.0 and up. [Learn how to check your Android version](#).

#### Find your phone

- [Be ready to find a lost Pixel phone](#)
- [Find, secure, or erase a lost Pixel phone](#)
- [How Find Hub protects your data](#)
- [Fix issues with Find Hub](#)
- [Find Hub acceptable use policy](#)
- [Share & manage devices with Find Hub](#)

<https://support.google.com/pixelphone/answer/9338817?>

## Find, secure, or erase a lost Android device

If you lose an Android device or Wear OS watch, you can find, secure, or erase it remotely. You can also help a friend find, secure, or erase their lost device with the [Find Hub app](#).

If you've added a Google Account to your device, Find Hub is automatically turned on. By default, your device is set to the "With network in high-traffic areas only" setting so that it stores encrypted recent locations with Google and helps find offline devices as part of a crowdsourced network of Android devices. To get help from the network finding your items on your Android device, set a PIN, pattern, or password. Your device's most recent location is available to the first account activated on the device.

<https://support.google.com/accounts/answer/6160491>

## Find Hub Network Accessory Specification



On this page ▾

GATT Specification

Authentication

Operations

Advertised frames

Ephemeral identifier (EID) computation

...

v1.3

The Find Hub Network (FHN) accessory specification defines an end-to-end encrypted approach for tracking beaconing Bluetooth Low Energy (BLE) devices. This page describes FHN as an extension to the Fast Pair specification. Providers should enable this extension if they have devices that are compatible with FHN and are willing to enable location tracking for those devices.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn>

Pixel Buds 2a [Overview](#) Tech specs Compare

Unbelievable  
sound.  
Unreal value.



### How can I find my Pixel Buds 2a?

You can easily locate your Pixel Buds 2a. Find Hub will pinpoint the precise location of your earbuds on a map.<sup>6</sup> And when you get close, you can give your earbuds a ring to find them faster. The earbuds will ring even louder while in the charging case, making them easier to locate.<sup>7</sup>

### Audio

Each earbud:

Custom-designed 11 mm dynamic speaker driver

Active Noise Cancellation with Silent Seal™ 1.5

Transparency mode

Active in-ear pressure relief

Case:

Ringtone speaker for Find Hub

[https://store.google.com/product/pixel\\_buds\\_2a?hl=en-US](https://store.google.com/product/pixel_buds_2a?hl=en-US)

### Connectivity

Each earbud:

Bluetooth 5.4

Super Wideband<sup>3</sup>

[https://store.google.com/product/pixel\\_buds\\_2a\\_specs?hl=en-US](https://store.google.com/product/pixel_buds_2a_specs?hl=en-US)

1[a]: at the first wireless device,


Google Find Hub Service involves the first wireless device, providing initial identification information to a central server, said initial identification information (e.g., Ephemeral Identifier

providing initial identification information to a central server, said initial identification information having been collected by the first wireless device from the second wireless device via a first, direct, short range local wireless link between the second and first wireless devices,

(EID) advertised via Bluetooth / BLE as part of the Find Hub Network (FHN) frame) having been collected by the first wireless device from the second wireless device via a first, direct, short range local wireless link between the second and first wireless devices.

For example, a first Google Wireless Device may receive, using Bluetooth / BLE, a plurality of short range transmissions (*e.g.* BLE advertisements) when the first Google Wireless Device is located within a detection range of a second Google Wireless Device that implements the Google Find Hub Service. Each of the Bluetooth / BLE transmissions (*e.g.*, advertised FHN (Find Hub Network) frames) includes an Ephemeral Identifier (EID). The Ephemeral identifier (EID) is generated from an Ephemeral Identity Key (EIK). The first Google Wireless Device encrypts location report(s) using the received Ephemeral Identifier (EID) and sends the encrypted location reports (and other associated details) to Google server(s).

## Find Hub Network Accessory Specification

On this page 

- GATT Specification
  - Authentication
  - Operations
- Advertised frames
  - Ephemeral identifier (EID) computation
- ...

v1.3

The Find Hub Network (FHN) accessory specification defines an end-to-end encrypted approach for tracking beaconing Bluetooth Low Energy (BLE) devices. This page describes FHN as an extension to the Fast Pair specification. Providers should enable this extension if they have devices that are compatible with FHN and are willing to enable location tracking for those devices.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn>

## Google's Find My Device becomes Find Hub amid expansion

Sarah Perez

13 May 2025 • 1 min read



Android users will have more ways to find their devices and other items, Google announced on Tuesday during the Android Show, a week before [Google I/O 2025](#). The company says its Find My Device feature, which allows Android users to locate lost phones and other devices, will become known as "Find Hub" as it rolls out support for more partners, satellite-based finding capabilities, and airline partnerships.

<https://au.finance.yahoo.com/news/googles-device-becomes-hub-amid-171936846.html>

## Unique Features of Google Find My Device

### 1. Global Positioning: Leveraging a Vast Network of Devices

Unlike other Bluetooth tracker apps, Google's Find My Device capitalizes on its extensive network of over a billion Android devices worldwide. This vast ecosystem allows the service to utilize these devices as "signal nodes," aiding users in locating lost items.

- **How It Works:**

- When your tracker (such as those compatible with Google's network) disconnects from your phone, it enters a "lost mode."
- Nearby Android devices detect the tracker's Bluetooth signal and securely upload the location information to Google's servers.
- You can then view the item's last known location in real-time via the Find My Device app.

This global positioning capability ensures that even if your item is far from your phone, other Android devices can assist in locating it, offering coverage that surpasses traditional Bluetooth trackers.

<https://www.seinxon.com/blogs/blog-posts/google-find-my-device-vs-other-bluetooth-tracker-apps>

### Unwanted tracking prevention

Certified FHN devices must also meet the requirements in the implementation version of the cross-platform specification for [Detecting Unwanted Location Trackers \(DULT\)](#).

Relevant guidelines specific to FHN to be compliant with DULT spec:

- Any FHN compatible device must be registered in the Nearby Device Console, and have the "Find Hub" capability activated.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn>

Step 4: Find offline devices and devices without power 

1. On your device, open **Settings**.
2. Tap **Google** > **All Services** (if tabs exist) > **Find Hub**.
3. Tap **Find your offline devices**.
4. To help you find offline items with Find Hub, if you don't have one, set a PIN, pattern, or password on your Android device. [Learn how to set screen lock on your device](#).

**Find offline devices settings**

By default, your device is set to the "With network in high-traffic areas only" setting so that it stores encrypted recent locations with Google and helps find offline devices as part of a crowdsourced network of Android devices. You can change this setting at any time:

- **Off:** Your device's encrypted recent locations won't be stored and your Android device won't participate in the network. [What happens when you turn off offline finding](#).
- **Without network:** Your device won't participate in the network. You can still locate your offline devices with their encrypted recent locations that were stored when they were online. [Offline finding without the network](#).
- **With network in high-traffic areas only** (default): Locate your offline devices with their encrypted recent locations. If you have a PIN, pattern, or password set on your Android Device, the network will help you locate your device in areas like airports or busy footpaths. [Offline finding in high-traffic areas](#).
- **With network in all areas:** Locate your offline devices with their stored and encrypted recent locations. If you have a PIN, pattern, or password set on your Android device, the network will help you locate your device in high-traffic and low-traffic areas. [Offline finding in all areas](#).

<https://support.google.com/accounts/answer/3265955?hl=en&sjid=14156669169198587197-NC>

## Finding your offline devices

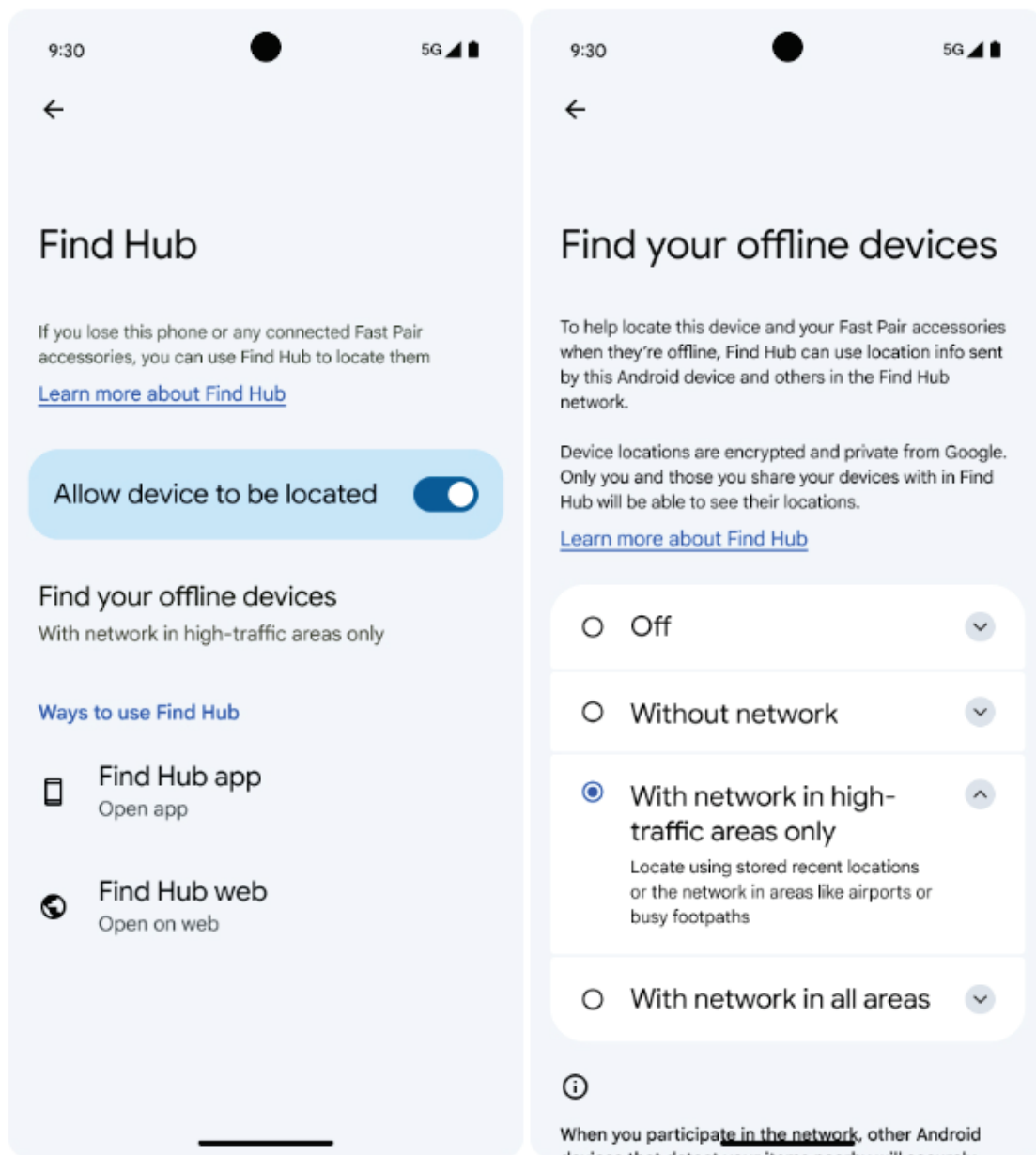
Your lost device may not always be online. To help you find your offline devices, Find Hub can also collect, store, and use encrypted location information sent by your Android device and others participating in the Find Hub network.

Leveraging the power of a crowdsourced network of Android devices, the Find Hub network can help you find a wide range of items, including Android phones and tablets that are offline, Fast Pair accessories like compatible earbuds, and tracker tags that you can attach to physical assets like your wallet, keys, or bike.

The network has been developed with advanced safeguards, including end-to-end encryption, to help protect the privacy of everyone participating in the network.

## Controlling how your device participates in the network

You can control how your Android device participates in the network at any time by visiting “Find your offline devices” in the Find Hub settings and choosing between the following options:



**“With network in high-traffic areas only”**

By default, your Android device helps others find their items in higher-traffic areas. If you have a PIN, pattern, or password set on your Android device, you’ll also receive help finding your items in higher-traffic areas.

When the owner of a lost item requests its location, the Find Hub network will — by default — aggregate the location sent by your device with locations sent from other Android devices that also detected the lost item.

**What is aggregation?**

With aggregation, the Find Hub network waits until multiple Android devices have detected a lost item. Find Hub then shows the owner of the lost item an aggregated location calculated from the multiple location reports.

This helps people, including you, find items in higher-traffic areas where items are most often lost, like airports or busy footpaths, while helping protect the privacy of everyone whose Android devices share location info to the network.

**Important:** To get help from the network when finding your items, on your Android device, you must set a PIN, pattern, or password. Until then, the network uses your Android device to help others find their items. Your Android device also stores encrypted recent locations for itself and connected accessories with Google. You can read more about this function under [Without network](#). Find Hub uses the best location available, whether from your own device or crowdsourced from the broader network (if you have a lock screen set), to help you find your item.

**“With network in all areas”**

If you want the Find Hub network to help you find your lost items in lower-traffic areas, you can opt in to sharing location info through the network to help others find lost items even when your device is the only one that has detected and shared a location for the item. Users who turn on this option help each other find items in both higher-traffic and lower-traffic areas. This option may help you find your lost items more quickly.

**Important:** To get help from the network when finding your items, on your Android device, you must set a PIN, pattern, or password. Until then, when you select this option, the network uses your Android device to help others find their items. Your Android device also stores encrypted recent locations for itself and connected accessories with Google. You can read more about this function under [Without network](#). Find Hub uses the best location available, whether from your own device or crowdsourced from the broader network (if you have a lock screen set), to help you find your item.

<https://support.google.com/product-documentation/answer/14796936>

The Seeker then calculates a one-time authentication key to be used in a subsequent write request. The authentication key is calculated as described in tables 2 through 5. Depending on the operation being requested, the Seeker proves knowledge of one or more of the following keys:

- **Account key:** The 16-byte Fast Pair account key, as defined in the Fast Pair specification.
- **Owner account key:** The Provider chooses one of the existing account keys as the owner account key the first time a Seeker accesses the Beacon Actions characteristic. The chosen owner account key can't be changed until the Provider is factory reset. The Provider *must not* remove the owner account key when it runs out of free account key slots.
  - Providers that already support FHN when paired for the first time (or support it when paired after factory reset) choose the first account key, because this is the only existing account key when the Seeker reads the provisioning state during pairing.
  - Providers that gain FHN support after they're already paired (for example, through a firmware update) can choose any existing account key. It's reasonable to choose the first account key that is used to read the provisioning state from the beacon actions characteristic after the firmware update, assuming the user who performed the update is the Provider's current owner.
- **Ephemeral identity key (EIK):** A 32-byte key chosen at random by the Seeker when performing the FHN provisioning process. This key is used to derive cryptographic keys that are used for end-to-end encrypting location reports. The Seeker never reveals it to the backend.
- **Recovery key:** Defined as `SHA256(ephemeral identity key || 0x01)`, truncated to the first 8 bytes. The key is stored on the backend and the Seeker can use it to [recover the EIK](#), provided the user expresses consent by pressing a button on the device.
- **Ring key:** Defined as `SHA256(ephemeral identity key || 0x02)`, truncated to the first 8 bytes. The key is stored on the backend and the Seeker can use it only to [ring](#) the device.
- **Unwanted tracking protection key:** Defined as `SHA256(ephemeral identity key || 0x03)`, truncated to the first 8 bytes. The key is stored on the backend and the Seeker can use it only to activate [unwanted tracking protection mode](#).

## Advertised frames

After provisioning, the Provider is expected to advertise FHN frames at least once every 2 seconds. If Fast Pair frames are advertised, the Provider should interleave the FHN frames within the regular Fast Pair advertisements. For example, every two seconds, the Provider should advertise seven Fast Pair advertisements and one FHN advertisement.

The conducted Bluetooth transmit power for FHN advertisements should be set to at least 0 dBm.

★ **Note:** It is recommended to advertise the FHN frames even when the device is in low power mode, as this allows finding the device if lost. For some devices, it might not be possible to advertise when the device is turned off due to the battery life implications. In such cases, it is strongly recommended to periodically wake up the device and advertise for a short period of time (e.g. wake up every 10 minutes and advertise for 10 seconds).

The FHN frame carries a public key used to encrypt location reports by any supporting client that contributes to the crowdsourcing network. Two types of elliptic curve keys are available: a 160-bit key that fits legacy BLE 4 frames, or 256-bit key that requires BLE 5 with extended advertising capabilities. The Provider's implementation determines which curve is used.

★ **Note:** Using 256-bit keys means that older phones that don't support BLE 5 can't scan and report for advertising devices, thus reducing the size of the network.

An FHN frame is structured as follows.

Octet	Value	Description
0	0x02	Length
1	0x01	Flags data type value
2	0x06	Flags data
3	0x18 or 0x19	Length
4	0x16	Service data data type value
5	0xAA	16-bit service UUID
6	0xFE	...
7	0x40 or 0x41	FHN frame type with unwanted tracking protection mode indication
8..27		20-byte ephemeral identifier
28		<a href="#">Hashed flags</a>

Table 9 shows the byte offsets and values for a 256-bit curve.

Octet	Value	Description
0	0x02	Length
1	0x01	Flags data type value
2	0x06	Flags data
3	0x24 or 0x25	Length
4	0x16	Service data data type value
5	0xAA	16-bit service UUID
6	0xFE	...
7	0x40 or 0x41	FHN frame type with unwanted tracking protection mode indication
8..39		32-byte ephemeral identifier
40		Hashed flags

**Table 9:** FHN frame supporting a 256-bit curve.

### Encryption with EID

★ **Note:** The following details for encryption and decryption with EIDs are only included for completeness of the specification. Only the Seeker implements these, not the Provider.

To encrypt a message  $m$ , a sighter (having read  $R_x$  from the beacon) would do the following:

1. Choose a random number  $s$  in  $F_p$ , as defined in the [EID computation](#) section.
2. Compute  $S = s * G$ .
3. Compute  $R = (R_x, R_y)$  by substitution in the curve equation and picking an arbitrary  $R_y$  value out of the possible results.
4. Compute the 256-bit AES key  $k = \text{HKDF-SHA256}((s * R)_x)$  where  $(s * R)_x$  is the  $x$  coordinate of the curve multiplication result. Salt isn't specified.
5. Let  $UR_x$  and  $LR_x$  be the upper and lower 80-bits of  $R_x$ , respectively, in big-endian format. In a similar way, define  $US_x$  and  $LS_x$  for  $S$ .
6. Compute  $\text{nonce} = LR_x || LS_x$ .
7. Compute  $(m', \text{tag}) = \text{AES-EAX-256-ENC}(k, \text{nonce}, m)$ .
8. Send  $(UR_x, S_x, m', \text{tag})$  to the owner, possibly through an untrusted remote service.

### Decryption of values encrypted with EID

The owner's client, which is in possession of the EIK and the rotation period exponent, decrypts the message as follows:

1. Given  $UR_x$ , obtain the beacon time counter value on which  $UR_x$  is based. This can be done by the owner's client computing  $R_x$  values for beacon time counter values for the recent past and near future.
2. Given the beacon time counter value on which  $UR_x$  is based, compute the anticipated value of  $r$  as defined in the [EID computation](#) section.
3. Compute  $R = r * G$ , and verify a match to the value of  $UR_x$  provided by the sighter.
4. Compute  $S = (S_x, S_y)$  by substitution in the curve equation and picking an arbitrary  $S_y$  value out of the possible results.
5. Compute  $k = \text{HKDF-SHA256}((r * S)_x)$  where  $(r * S)_x$  is the  $x$  coordinate of the curve multiplication result.
6. Compute  $\text{nonce} = \text{LR}_x \parallel \text{LS}_x$ .
7. Compute  $m = \text{AES-EAX-256-DEC}(k, \text{nonce}, m', \text{tag})$ .

#### Current ephemeral identifier (device information code 0x0B)

The Provider can use the *current ephemeral identifier (code 0x0B)* to report the current EID and clock value when the Provider is provisioned for FHN, to sync the Seeker in case of a clock drift (for example, due to drained battery). Otherwise, the Seeker initiates a more expensive and less reliable connection for this purpose.

Octet	Data Type	Description	Value
0	uint8	Device information event	0x03
1	uint8	Current ephemeral identifier	0x0B
2 - 3	uint16	Additional data length	0x0018 or 0x0024
4 - 7	byte array	Clock value	Example: 0x13F9EA80
8 - 19 or 31	byte array	Current EID	Example: 0x1122334455667788990011223344556677889900

**Table 13:** Device information event: clock sync.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn>

### **How does crowdsourcing work?**

Android devices participating in the Find Hub network use Bluetooth to scan for nearby items. If they detect your items, they securely send the location where they detected the items to Find Hub. Your Android device does the same to help others find their lost items when it detects them nearby.

### **End-to-end encryption**

The Find Hub network encrypts the locations of your items using a unique key that only you can access by entering your Android device's PIN, pattern, or password.

This end-to-end encryption, which is backed by the same technology used by Google Password Manager to secure your passwords, ensures that the locations of your items are private from Google. They're only visible to you and those you share your items with in Find Hub.

**Important:** If you haven't set a PIN, pattern, or password on your Android device, you must set one to take advantage of Find Hub network. Until then, by default, the network uses your Android device to help others find their items and your Android device stores encrypted recent locations for itself and connected accessories with Google to help you find your devices. To take advantage of the Find Hub network and have the best offline finding experience, set a PIN, pattern, or password on your Android device.

<https://support.google.com/product-documentation/answer/14796936?hl=en>

- **Data Safeguards:** We've implemented protections that help ensure the privacy of everyone participating in the network and the crowdsourced location data that powers it.

- **Location data is end-to-end encrypted.** When Android devices participating in the network report the location of a Bluetooth tag, the location is end-to-end encrypted using a key that is only accessible to the Bluetooth tag owner and anyone the owner has shared the tag with in the Find Hub app. Only the Bluetooth tag owner (and those they've chosen to share access with) can decrypt and view the tag's location. With end-to-end encrypted location data, Google cannot decrypt, see, or otherwise use the location data.
- **Private, crowdsourced location reports.** These end-to-end encrypted locations are contributed to the Find Hub network in a manner that does not allow Google to identify the owners of the nearby Android devices that provided the location data. And when the Find Hub network shows the location and timestamp to the Bluetooth tag's owner to help them find their belongings, no other information about the nearby Android devices that contributed the data is included.

<https://security.googleblog.com/2024/04/find-my-device-network-security-privacy-protections.html>

#### **Data processed by the network**

In addition to end-to-end encrypted locations, the Find Hub network processes data such as temporary device identifiers, timestamps when your device detects an item and when you request the location of your lost items, and info about the Fast Pair accessories that you have paired to your device or share with others. The Find Hub network uses this data for reasons like implementing features, delivering location info to the right person when an item is lost, and providing privacy and anti-abuse protections, such as the aggregation feature described below. Importantly, Google can't identify you when your Android device shares the location of a detected item.

Individuals using the Find Hub network to find their lost items don't receive any information from the network other than the location where their item was detected and approximately when their item was last seen.

	<a href="https://support.google.com/product-documentation/answer/14796936">https://support.google.com/product-documentation/answer/14796936</a>
<p><b>1[b]:</b> wherein the initial identification information is associated at the central server with an identity of a user or entity associated with the second wireless device, and wherein the initial identification information is provided to the central server, by the first wireless device, over a second wireless link;</p>	<p>In Google Wireless Devices, the initial identification information (<i>e.g.</i>, Ephemeral Identifier (EID) advertised via Bluetooth / BLE as part of the Find Hub Network (FHN) frame) is associated at the central server with an identity of a user or entity associated with the second wireless device, and wherein the initial identification information is provided to the central server, by the first wireless device, over a second wireless link.</p> <p>For example, the first Google Wireless Device receives an Ephemeral Identifier (EID) as part of the FHN (Find Hub Network) advertisement frame (transmitted over Bluetooth/BLE) from the second Google Wireless Device. The first Google Wireless Devices subsequently use the received Ephemeral identifier (EID) to encrypt location report(s) and upload the location report(s) (and other associated details) to Google server(s) over Wi-Fi / Cellular networks. The Google server(s) subsequently provides the uploaded location report(s), for the second Google Wireless Device, only to the specific owner of the second Google Wireless Device.</p> <p><b>Unique Features of Google Find My Device</b></p> <p><b>1. Global Positioning: Leveraging a Vast Network of Devices</b></p> <p>Unlike other Bluetooth tracker apps, Google's Find My Device capitalizes on its extensive network of over a billion Android devices worldwide. This vast ecosystem allows the service to utilize these devices as "signal nodes," aiding users in locating lost items.</p> <ul style="list-style-type: none"> <li>• <b>How It Works:</b> <ul style="list-style-type: none"> <li>◦ When your tracker (such as those compatible with Google's network) disconnects from your phone, it enters a "lost mode."</li> <li>◦ Nearby Android devices detect the tracker's Bluetooth signal and securely upload the location information to Google's servers.</li> <li>◦ You can then view the item's last known location in real-time via the Find My Device app.</li> </ul> </li> </ul> <p>This global positioning capability ensures that even if your item is far from your phone, other Android devices can assist in locating it, offering coverage that surpasses traditional Bluetooth trackers.</p> <p><a href="https://www.seinxon.com/blogs/blog-posts/google-find-my-device-vs-other-bluetooth-tracker-apps">https://www.seinxon.com/blogs/blog-posts/google-find-my-device-vs-other-bluetooth-tracker-apps</a></p>

### **How does crowdsourcing work?**

Android devices participating in the Find Hub network use Bluetooth to scan for nearby items. If they detect your items, they securely send the location where they detected the items to Find Hub. Your Android device does the same to help others find their lost items when it detects them nearby.

### **End-to-end encryption**

The Find Hub network encrypts the locations of your items using a unique key that only you can access by entering your Android device's PIN, pattern, or password.

This end-to-end encryption, which is backed by the same technology used by Google Password Manager to secure your passwords, ensures that the locations of your items are private from Google. They're only visible to you and those you share your items with in Find Hub.

**Important:** If you haven't set a PIN, pattern, or password on your Android device, you must set one to take advantage of Find Hub network. Until then, by default, the network uses your Android device to help others find their items and your Android device stores encrypted recent locations for itself and connected accessories with Google to help you find your devices. To take advantage of the Find Hub network and have the best offline finding experience, set a PIN, pattern, or password on your Android device.

<https://support.google.com/product-documentation/answer/14796936?hl=en>

- **Data Safeguards:** We've implemented protections that help ensure the privacy of everyone participating in the network and the crowdsourced location data that powers it.

- **Location data is end-to-end encrypted.** When Android devices participating in the network report the location of a Bluetooth tag, the location is end-to-end encrypted using a key that is only accessible to the Bluetooth tag owner and anyone the owner has shared the tag with in the Find Hub app. Only the Bluetooth tag owner (and those they've chosen to share access with) can decrypt and view the tag's location. With end-to-end encrypted location data, Google cannot decrypt, see, or otherwise use the location data.

- **Private, crowdsourced location reports.** These end-to-end encrypted locations are contributed to the Find Hub network in a manner that does not allow Google to identify the owners of the nearby Android devices that provided the location data. And when the Find Hub network shows the location and timestamp to the Bluetooth tag's owner to help them find their belongings, no other information about the nearby Android devices that contributed the data is included.

<https://security.googleblog.com/2024/04/find-my-device-network-security-privacy-protections.html>

### Data processed by the network

In addition to end-to-end encrypted locations, the Find Hub network processes data such as temporary device identifiers, timestamps when your device detects an item and when you request the location of your lost items, and info about the Fast Pair accessories that you have paired to your device or share with others. The Find Hub network uses this data for reasons like implementing features, delivering location info to the right person when an item is lost, and providing privacy and anti-abuse protections, such as the aggregation feature described below. Importantly, Google can't identify you when your Android device shares the location of a detected item.

Individuals using the Find Hub network to find their lost items don't receive any information from the network other than the location where their item was detected and approximately when their item was last seen.

<https://support.google.com/android/answer/14796936?hl=en-en>

### Connectivity and Location

Wi-Fi 7 (802.11be) with 2.4GHz+5GHz+6GHz, 2x2+2x2 MIMO

Bluetooth® v5.3 with dual antennas for enhanced quality and connection

NFC

Google Cast

Dual Band GNSS  
GPS, GLONASS, Galileo

**Network**<sup>26</sup>

5G mmWave + Sub 6GHz<sup>27</sup> Model G2YBB

GSM/EDGE: Quad-band (850, 900, 1800, 1900 MHz)

UMTS/HSPA+/HSDPA: Bands 1,2,4,5,6,8,19

LTE: Bands

B1/2/3/4/5/7/8/12/13/14/17/18/19/20/25/26/28/29/30/38/40/41/48/66/71

5G Sub-6<sup>27</sup> : Bands

n1/2/3/5/7/8/12/14/20/25/26/28/29/30/38/40/41/48/66/70/71/77/78

5G mmWave<sup>27</sup> : Bands n258/260/261

eSIM

[https://store.google.com/product/pixel\\_9\\_specs?hl=en-US](https://store.google.com/product/pixel_9_specs?hl=en-US)

## Advertised frames

After provisioning, the Provider is expected to advertise FHN frames at least once every 2 seconds. If Fast Pair frames are advertised, the Provider should interleave the FHN frames within the regular Fast Pair advertisements. For example, every two seconds, the Provider should advertise seven Fast Pair advertisements and one FHN advertisement.

The conducted Bluetooth transmit power for FHN advertisements should be set to at least 0 dBm.

★ **Note:** It is recommended to advertise the FHN frames even when the device is in low power mode, as this allows finding the device if lost. For some devices, it might not be possible to advertise when the device is turned off due to the battery life implications. In such cases, it is strongly recommended to periodically wake up the device and advertise for a short period of time (e.g. wake up every 10 minutes and advertise for 10 seconds).

The FHN frame carries a public key used to encrypt location reports by any supporting client that contributes to the crowdsourcing network. Two types of elliptic curve keys are available: a 160-bit key that fits legacy BLE 4 frames, or 256-bit key that requires BLE 5 with extended advertising capabilities. The Provider's implementation determines which curve is used.

★ **Note:** Using 256-bit keys means that older phones that don't support BLE 5 can't scan and report for advertising devices, thus reducing the size of the network.

An FHN frame is structured as follows.

Octet	Value	Description
0	0x02	Length
1	0x01	Flags data type value
2	0x06	Flags data
3	0x18 or 0x19	Length
4	0x16	Service data data type value
5	0xAA	16-bit service UUID
6	0xFE	...
7	0x40 or 0x41	FHN frame type with unwanted tracking protection mode indication
8..27		20-byte ephemeral identifier
28		<a href="#">Hashed flags</a>

Table 9 shows the byte offsets and values for a 256-bit curve.

Octet	Value	Description
0	0x02	Length
1	0x01	Flags data type value
2	0x06	Flags data
3	0x24 or 0x25	Length
4	0x16	Service data data type value
5	0xAA	16-bit service UUID
6	0xFE	...
7	0x40 or 0x41	FHN frame type with unwanted tracking protection mode indication
8..39		32-byte ephemeral identifier
40		Hashed flags

**Table 9:** FHN frame supporting a 256-bit curve.

### Encryption with EID

★ **Note:** The following details for encryption and decryption with EIDs are only included for completeness of the specification. Only the Seeker implements these, not the Provider.

To encrypt a message  $m$ , a sighter (having read  $R_x$  from the beacon) would do the following:

1. Choose a random number  $s$  in  $F_p$ , as defined in the [EID computation](#) section.
2. Compute  $S = s * G$ .
3. Compute  $R = (R_x, R_y)$  by substitution in the curve equation and picking an arbitrary  $R_y$  value out of the possible results.
4. Compute the 256-bit AES key  $k = \text{HKDF-SHA256}((s * R)_x)$  where  $(s * R)_x$  is the  $x$  coordinate of the curve multiplication result. Salt isn't specified.
5. Let  $UR_x$  and  $LR_x$  be the upper and lower 80-bits of  $R_x$ , respectively, in big-endian format. In a similar way, define  $US_x$  and  $LS_x$  for  $S$ .
6. Compute  $\text{nonce} = LR_x || LS_x$ .
7. Compute  $(m', \text{tag}) = \text{AES-EAX-256-ENC}(k, \text{nonce}, m)$ .
8. Send  $(UR_x, S_x, m', \text{tag})$  to the owner, possibly through an untrusted remote service.

	<p><b>Decryption of values encrypted with EID</b></p> <p>The owner's client, which is in possession of the EIK and the rotation period exponent, decrypts the message as follows:</p> <ol style="list-style-type: none"> <li>Given <math>UR_x</math>, obtain the beacon time counter value on which <math>UR_x</math> is based. This can be done by the owner's client computing <math>R_x</math> values for beacon time counter values for the recent past and near future.</li> <li>Given the beacon time counter value on which <math>UR_x</math> is based, compute the anticipated value of <math>r</math> as defined in the <a href="#">EID computation</a> section.</li> <li>Compute <math>R = r * G</math>, and verify a match to the value of <math>UR_x</math> provided by the sighter.</li> <li>Compute <math>S = (S_x, S_y)</math> by substitution in the curve equation and picking an arbitrary <math>S_y</math> value out of the possible results.</li> <li>Compute <math>k = \text{HKDF-SHA256}((r * S)_x)</math> where <math>(r * S)_x</math> is the <math>x</math> coordinate of the curve multiplication result.</li> <li>Compute <math>\text{nonce} = \text{LR}_x \    \ \text{LS}_x</math>.</li> <li>Compute <math>m = \text{AES-EAX-256-DEC}(k, \text{nonce}, m', \text{tag})</math>.</li> </ol> <p><b>Current ephemeral identifier (device information code 0x0B)</b></p> <p>The Provider can use the <i>current ephemeral identifier (code 0x0B)</i> to report the current EID and clock value when the Provider is provisioned for FHN, to sync the Seeker in case of a clock drift (for example, due to drained battery). Otherwise, the Seeker initiates a more expensive and less reliable connection for this purpose:</p> <table border="1"> <thead> <tr> <th>Octet</th> <th>Data Type</th> <th>Description</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>uint8</td> <td>Device information event</td> <td>0x03</td> </tr> <tr> <td>1</td> <td>uint8</td> <td>Current ephemeral identifier</td> <td>0x0B</td> </tr> <tr> <td>2 - 3</td> <td>uint16</td> <td>Additional data length</td> <td>0x0018 or 0x0024</td> </tr> <tr> <td>4 - 7</td> <td>byte array</td> <td>Clock value</td> <td>Example: 0x13F9EA80</td> </tr> <tr> <td>8 - 19 or 31</td> <td>byte array</td> <td>Current EID</td> <td>Example: 0x1122334455667788990011223344556677889900</td> </tr> </tbody> </table> <p><b>Table 13:</b> Device information event: clock sync.</p> <p><a href="https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn">https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn</a></p>	Octet	Data Type	Description	Value	0	uint8	Device information event	0x03	1	uint8	Current ephemeral identifier	0x0B	2 - 3	uint16	Additional data length	0x0018 or 0x0024	4 - 7	byte array	Clock value	Example: 0x13F9EA80	8 - 19 or 31	byte array	Current EID	Example: 0x1122334455667788990011223344556677889900
Octet	Data Type	Description	Value																						
0	uint8	Device information event	0x03																						
1	uint8	Current ephemeral identifier	0x0B																						
2 - 3	uint16	Additional data length	0x0018 or 0x0024																						
4 - 7	byte array	Clock value	Example: 0x13F9EA80																						
8 - 19 or 31	byte array	Current EID	Example: 0x1122334455667788990011223344556677889900																						
<p><b>1[c]:</b> at the second wireless device, upon an occurrence of a predetermined event coordinated with said central server, within a specific application on the second wireless device, providing modified identification information</p>	<p>Google Find Hub Service involves the second wireless device, upon an occurrence of a predetermined event coordinated with said central server, within a specific application on the second wireless device, providing modified identification information (e.g., Ephemeral Identifier (EID) advertised via Bluetooth/BLE as part of the Find Hub Network (FHN) frame) over the first, direct, short range local wireless link in place of the initial identification information, such that the modified identification is associated at the central server with said identity of a user or entity associated with the second device.</p> <p>For example, in Google Find Hub Service, the Ephemeral identifiers (EID) are sent over Bluetooth/BLE as part of the FHN advertisement frames from the second Google Wireless Device to the first Google Wireless Device. At the second Google Wireless Device, the EID is rotated as per the Rotation period exponent that is set to 10 (which corresponds to 1024 seconds). Therefore, after every 1024 seconds, a new EID is generated and transmitted from the second Google Wireless Device to first Google Wireless Device as part of the FHN advertisement frame. The new EID is also generated from the same Ephemeral Identity Key (EIK). The first Google Wireless Device receives the new EID and encrypts the location report(s) using the received new EID. The encrypted location report(s) (and other associated details) are subsequently uploaded to</p>																								

over the first, direct, short range local wireless link in place of the initial identification information, such that the modified identification information is associated at the central server with said identity of a user or entity associated with the second device; and

Google server(s). The Google server(s) provides the uploaded location report(s), encrypted with the new Ephemeral Identifier (EID), only to the same specific owner of the second Google Wireless Device.

### Advertised frames

After provisioning, the Provider is expected to advertise FHN frames at least once every 2 seconds. If Fast Pair frames are advertised, the Provider should interleave the FHN frames within the regular Fast Pair advertisements. For example, every two seconds, the Provider should advertise seven Fast Pair advertisements and one FHN advertisement.

The conducted Bluetooth transmit power for FHN advertisements should be set to at least 0 dBm.

★ **Note:** It is recommended to advertise the FHN frames even when the device is in low power mode, as this allows finding the device if lost. For some devices, it might not be possible to advertise when the device is turned off due to the battery life implications. In such cases, it is strongly recommended to periodically wake up the device and advertise for a short period of time (e.g. wake up every 10 minutes and advertise for 10 seconds).

The FHN frame carries a public key used to encrypt location reports by any supporting client that contributes to the crowdsourcing network. Two types of elliptic curve keys are available: a 160-bit key that fits legacy BLE 4 frames, or 256-bit key that requires BLE 5 with extended advertising capabilities. The Provider's implementation determines which curve is used.

★ **Note:** Using 256-bit keys means that older phones that don't support BLE 5 can't scan and report for advertising devices, thus reducing the size of the network.

### Ephemeral identifier (EID) computation

A random is generated by AES-ECB-256 encrypting the following data structure with the ephemeral identity key:

Octet	Field	Description
0 - 10	Padding	Value = 0xFF
11	K	Rotation period exponent
12 - 15	TS[0]...TS[3]	Beacon time counter, in 32-bit big-endian format. The K lowest bits are cleared.
16 - 26	Padding	Value = 0x00
27	K	Rotation period exponent
28 - 31	TS[0]...TS[3]	Beacon time counter, in 32-bit big-endian format. The K lowest bits are cleared.

**Table 10:** Construction of a pseudorandom number.

★ **Note:** The device is assumed to have a 32-bit time counter in seconds.

★ **Note:** Rotation period exponent is fixed and set to 10, corresponding to 1024 seconds.

## Encryption with EID

★ **Note:** The following details for encryption and decryption with EIDs are only included for completeness of the specification. Only the Seeker implements these, not the Provider.

To encrypt a message  $m$ , a sighter (having read  $R_x$  from the beacon) would do the following:

1. Choose a random number  $s$  in  $F_p$ , as defined in the [EID computation](#) section.
2. Compute  $S = s * G$ .
3. Compute  $R = (R_x, R_y)$  by substitution in the curve equation and picking an arbitrary  $R_y$  value out of the possible results.
4. Compute the 256-bit AES key  $k = \text{HKDF-SHA256}((s * R)_x)$  where  $(s * R)_x$  is the  $x$  coordinate of the curve multiplication result. Salt isn't specified.
5. Let  $UR_x$  and  $LR_x$  be the upper and lower 80-bits of  $R_x$ , respectively, in big-endian format. In a similar way, define  $US_x$  and  $LS_x$  for  $S$ .
6. Compute  $\text{nonce} = LR_x || LS_x$ .
7. Compute  $(m', \text{tag}) = \text{AES-EAX-256-ENC}(k, \text{nonce}, m)$ .
8. Send  $(UR_x, S_x, m', \text{tag})$  to the owner, possibly through an untrusted remote service.

## Decryption of values encrypted with EID ⇄

The owner's client, which is in possession of the EIK and the rotation period exponent, decrypts the message as follows:

1. Given  $UR_x$ , obtain the beacon time counter value on which  $UR_x$  is based. This can be done by the owner's client computing  $R_x$  values for beacon time counter values for the recent past and near future.
2. Given the beacon time counter value on which  $UR_x$  is based, compute the anticipated value of  $r$  as defined in the [EID computation](#) section.
3. Compute  $R = r * G$ , and verify a match to the value of  $UR_x$  provided by the sighter.
4. Compute  $S = (S_x, S_y)$  by substitution in the curve equation and picking an arbitrary  $S_y$  value out of the possible results.
5. Compute  $k = \text{HKDF-SHA256}((r * S)_x)$  where  $(r * S)_x$  is the  $x$  coordinate of the curve multiplication result.
6. Compute  $\text{nonce} = LR_x || LS_x$ .
7. Compute  $m = \text{AES-EAX-256-DEC}(k, \text{nonce}, m', \text{tag})$ .

## ID rotation

A resolvable (RPA) or non-resolvable (NRPA) BLE address must be used for advertising FHN frames. RPA is required for LE Audio (LEA) devices and is recommended for other devices, with the exception of locator tags that don't use bonding.

Fast Pair advertisement, FHN advertisement and the corresponding BLE address(es) should rotate at the same time. Rotation should happen every 1024 seconds on average. The precise point at which the beacon starts advertising the new identifier must be randomized within the window.

The recommended approach to randomize the rotation time is to set it to the next anticipated rotation time (if no randomization was applied) plus a positive randomized time factor in the range of 1 to 204 seconds.

When the device is in unwanted tracking protection mode, the BLE address of the FHN advertisement should be fixed, but the RPA for FP non-discoverable advertisement (such as Fast Pair) must keep rotating. It's acceptable to use different addresses for the different protocols.

Upon receiving a capability update request (0x0601), if the Provider has enabled support for FHN tracking, it should respond as shown in table 12.

Octet	Data Type	Description	Value
0	uint8	Device capability sync event	0x06
1	uint8	FHN tracking	0x03
2 - 3	uint16	Additional data length	0x0007
4	uint8	FHN provisioning state	0x00 if unprovisioned; 0x01 if provisioned by any account
5 - 10	byte array	The current BLE MAC address of the device	<i>varies</i>

**Table 12:** Device capability sync event: added tracking capability.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn#eid-computation>

## Unique Features of Google Find My Device

### 1. Global Positioning: Leveraging a Vast Network of Devices

Unlike other Bluetooth tracker apps, Google's Find My Device capitalizes on its extensive network of over a billion Android devices worldwide. This vast ecosystem allows the service to utilize these devices as "signal nodes," aiding users in locating lost items.

- **How It Works:**

- When your tracker (such as those compatible with Google's network) disconnects from your phone, it enters a "lost mode."
- Nearby Android devices detect the tracker's Bluetooth signal and securely upload the location information to Google's servers.
- You can then view the item's last known location in real-time via the Find My Device app.

This global positioning capability ensures that even if your item is far from your phone, other Android devices can assist in locating it, offering coverage that surpasses traditional Bluetooth trackers.

<https://www.seinxon.com/blogs/blog-posts/google-find-my-device-vs-other-bluetooth-tracker-apps>

### **How does crowdsourcing work?**

Android devices participating in the Find Hub network use Bluetooth to scan for nearby items. If they detect your items, they securely send the location where they detected the items to Find Hub. Your Android device does the same to help others find their lost items when it detects them nearby.

### **End-to-end encryption**

The Find Hub network encrypts the locations of your items using a unique key that only you can access by entering your Android device's PIN, pattern, or password.

This end-to-end encryption, which is backed by the same technology used by Google Password Manager to secure your passwords, ensures that the locations of your items are private from Google. They're only visible to you and those you share your items with in Find Hub.

**Important:** If you haven't set a PIN, pattern, or password on your Android device, you must set one to take advantage of Find Hub network. Until then, by default, the network uses your Android device to help others find their items and your Android device stores encrypted recent locations for itself and connected accessories with Google to help you find your devices. To take advantage of the Find Hub network and have the best offline finding experience, set a PIN, pattern, or password on your Android device.

<https://support.google.com/product-documentation/answer/14796936?hl=en>

- **Data Safeguards:** We've implemented protections that help ensure the privacy of everyone participating in the network and the crowdsourced location data that powers it.

- **Location data is end-to-end encrypted.** When Android devices participating in the network report the location of a Bluetooth tag, the location is end-to-end encrypted using a key that is only accessible to the Bluetooth tag owner and anyone the owner has shared the tag with in the Find Hub app. Only the Bluetooth tag owner (and those they've chosen to share access with) can decrypt and view the tag's location. With end-to-end encrypted location data, Google cannot decrypt, see, or otherwise use the location data.

- **Private, crowdsourced location reports.** These end-to-end encrypted locations are contributed to the Find Hub network in a manner that does not allow Google to identify the owners of the nearby Android devices that provided the location data. And when the Find Hub network shows the location and timestamp to the Bluetooth tag's owner to help them find their belongings, no other information about the nearby Android devices that contributed the data is included.

<https://security.googleblog.com/2024/04/find-my-device-network-security-privacy-protections.html>

	<p><b>Data processed by the network</b></p> <p>In addition to end-to-end encrypted locations, the Find Hub network processes data such as temporary device identifiers, timestamps when your device detects an item and when you request the location of your lost items, and info about the Fast Pair accessories that you have paired to your device or share with others. The Find Hub network uses this data for reasons like implementing features, delivering location info to the right person when an item is lost, and providing privacy and anti-abuse protections, such as the aggregation feature described below. Importantly, Google can't identify you when your Android device shares the location of a detected item.</p> <p>Individuals using the Find Hub network to find their lost items don't receive any information from the network other than the location where their item was detected and approximately when their item was last seen.</p> <p><a href="https://support.google.com/android/answer/14796936?hl=en-en">https://support.google.com/android/answer/14796936?hl=en-en</a></p>
<p><b>1[d]:</b> at the first wireless device, collecting said modified identification information.</p>	<p>Google Find Hub Service involves the first wireless device, collecting said modified identification information. For example, the FHN (Find Hub Network) frames, including the new Ephemeral Identifier (EID), advertised by the second Google Wireless Device associated with the new EID are received by the first Google Wireless Device over Bluetooth / BLE.</p> <p><b>Advertised frames</b></p> <p>After provisioning, the Provider is expected to advertise FHN frames at least once every 2 seconds. If Fast Pair frames are advertised, the Provider should interleave the FHN frames within the regular Fast Pair advertisements. For example, every two seconds, the Provider should advertise seven Fast Pair advertisements and one FHN advertisement.</p> <p>The conducted Bluetooth transmit power for FHN advertisements should be set to at least 0 dBm.</p> <p>★ <b>Note:</b> It is recommended to advertise the FHN frames even when the device is in low power mode, as this allows finding the device if lost. For some devices, it might not be possible to advertise when the device is turned off due to the battery life implications. In such cases, it is strongly recommended to periodically wake up the device and advertise for a short period of time (e.g. wake up every 10 minutes and advertise for 10 seconds).</p> <p>The FHN frame carries a public key used to encrypt location reports by any supporting client that contributes to the crowdsourcing network. Two types of elliptic curve keys are available: a 160-bit key that fits legacy BLE 4 frames, or 256-bit key that requires BLE 5 with extended advertising capabilities. The Provider's implementation determines which curve is used.</p> <p>★ <b>Note:</b> Using 256-bit keys means that older phones that don't support BLE 5 can't scan and report for advertising devices, thus reducing the size of the network.</p>

### Ephemeral identifier (EID) computation

A random is generated by AES-ECB-256 encrypting the following data structure with the ephemeral identity key:

Octet	Field	Description
0 - 10	Padding	Value = 0xFF
11	K	Rotation period exponent
12 - 15	TS[0]...TS[3]	Beacon time counter, in 32-bit big-endian format. The K lowest bits are cleared.
16 - 26	Padding	Value = 0x00
27	K	Rotation period exponent
28 - 31	TS[0]...TS[3]	Beacon time counter, in 32-bit big-endian format. The K lowest bits are cleared.

**Table 10:** Construction of a pseudorandom number.

★ **Note:** The device is assumed to have a 32-bit time counter in seconds.

★ **Note:** Rotation period exponent is fixed and set to 10, corresponding to 1024 seconds.

### Encryption with EID

★ **Note:** The following details for encryption and decryption with EIDs are only included for completeness of the specification. Only the Seeker implements these, not the Provider.

To encrypt a message  $m$ , a sighter (having read  $R_x$  from the beacon) would do the following:

1. Choose a random number  $s$  in  $F_p$ , as defined in the [EID computation](#) section.
2. Compute  $S = s * G$ .
3. Compute  $R = (R_x, R_y)$  by substitution in the curve equation and picking an arbitrary  $R_y$  value out of the possible results.
4. Compute the 256-bit AES key  $k = \text{HKDF-SHA256}((s * R)_x)$  where  $(s * R)_x$  is the  $x$  coordinate of the curve multiplication result. Salt isn't specified.
5. Let  $UR_x$  and  $LR_x$  be the upper and lower 80-bits of  $R_x$ , respectively, in big-endian format. In a similar way, define  $US_x$  and  $LS_x$  for  $S$ .
6. Compute  $\text{nonce} = LR_x || LS_x$ .
7. Compute  $(m', \text{tag}) = \text{AES-EAX-256-ENC}(k, \text{nonce}, m)$ .
8. Send  $(UR_x, S_x, m', \text{tag})$  to the owner, possibly through an untrusted remote service.

### Decryption of values encrypted with EID ⇄

The owner's client, which is in possession of the EIK and the rotation period exponent, decrypts the message as follows:

1. Given  $UR_x$ , obtain the beacon time counter value on which  $UR_x$  is based. This can be done by the owner's client computing  $R_x$  values for beacon time counter values for the recent past and near future.
2. Given the beacon time counter value on which  $UR_x$  is based, compute the anticipated value of  $r$  as defined in the [EID computation](#) section.
3. Compute  $R = r * G$ , and verify a match to the value of  $UR_x$  provided by the sighter.
4. Compute  $S = (S_x, S_y)$  by substitution in the curve equation and picking an arbitrary  $S_y$  value out of the possible results.
5. Compute  $k = \text{HKDF-SHA256}((r * S)_x)$  where  $(r * S)_x$  is the  $x$  coordinate of the curve multiplication result.
6. Compute  $\text{nonce} = LR_x \parallel LS_x$ .
7. Compute  $m = \text{AES-EAX-256-DEC}(k, \text{nonce}, m', \text{tag})$ .

### ID rotation

A resolvable (RPA) or non-resolvable (NRPA) BLE address must be used for advertising FHN frames. RPA is required for LE Audio (LEA) devices and is recommended for other devices, with the exception of locator tags that don't use bonding.

Fast Pair advertisement, FHN advertisement and the corresponding BLE address(es) should rotate at the same time. Rotation should happen every 1024 seconds on average. The precise point at which the beacon starts advertising the new identifier must be randomized within the window.

The recommended approach to randomize the rotation time is to set it to the next anticipated rotation time (if no randomization was applied) plus a positive randomized time factor in the range of 1 to 204 seconds.

When the device is in unwanted tracking protection mode, the BLE address of the FHN advertisement should be fixed, but the RPA for FP non-discoverable advertisement (such as Fast Pair) must keep rotating. It's acceptable to use different addresses for the different protocols.

Upon receiving a capability update request (0x0601), if the Provider has enabled support for FHN tracking, it should respond as shown in table 12.

Octet	Data Type	Description	Value
0	uint8	Device capability sync event	0x06
1	uint8	FHN tracking	0x03
2 - 3	uint16	Additional data length	0x0007
4	uint8	FHN provisioning state	0x00 if unprovisioned; 0x01 if provisioned by any account
5 - 10	byte array	The current BLE MAC address of the device	varies

**Table 12:** Device capability sync event: added tracking capability.

<https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn#eid-computation>

## Unique Features of Google Find My Device

### 1. Global Positioning: Leveraging a Vast Network of Devices

Unlike other Bluetooth tracker apps, Google's Find My Device capitalizes on its extensive network of over a billion Android devices worldwide. This vast ecosystem allows the service to utilize these devices as "signal nodes," aiding users in locating lost items.

- **How It Works:**

- When your tracker (such as those compatible with Google's network) disconnects from your phone, it enters a "lost mode."
- Nearby Android devices detect the tracker's Bluetooth signal and securely upload the location information to Google's servers.
- You can then view the item's last known location in real-time via the Find My Device app.

This global positioning capability ensures that even if your item is far from your phone, other Android devices can assist in locating it, offering coverage that surpasses traditional Bluetooth trackers.

<https://www.seinxon.com/blogs/blog-posts/google-find-my-device-vs-other-bluetooth-tracker-apps>

### How does crowdsourcing work?

Android devices participating in the Find Hub network use Bluetooth to scan for nearby items. If they detect your items, they securely send the location where they detected the items to Find Hub. Your Android device does the same to help others find their lost items when it detects them nearby.

### End-to-end encryption

The Find Hub network encrypts the locations of your items using a unique key that only you can access by entering your Android device's PIN, pattern, or password.

This end-to-end encryption, which is backed by the same technology used by Google Password Manager to secure your passwords, ensures that the locations of your items are private from Google. They're only visible to you and those you share your items with in Find Hub.

**Important:** If you haven't set a PIN, pattern, or password on your Android device, you must set one to take advantage of Find Hub network. Until then, by default, the network uses your Android device to help others find their items and your Android device stores encrypted recent locations for itself and connected accessories with Google to help you find your devices. To take advantage of the Find Hub network and have the best offline finding experience, set a PIN, pattern, or password on your Android device.

<https://support.google.com/product-documentation/answer/14796936?hl=en>