Security and Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System

Authors: Tingfeng Yu, James Henderson, Alwen Tiu, Thomas Haines

Australian National University

Outline

- **Introduction**
- Background & Methodology
- the FMM Protocol
- Security analysis
- Summary

Bluetooth Location Tracking

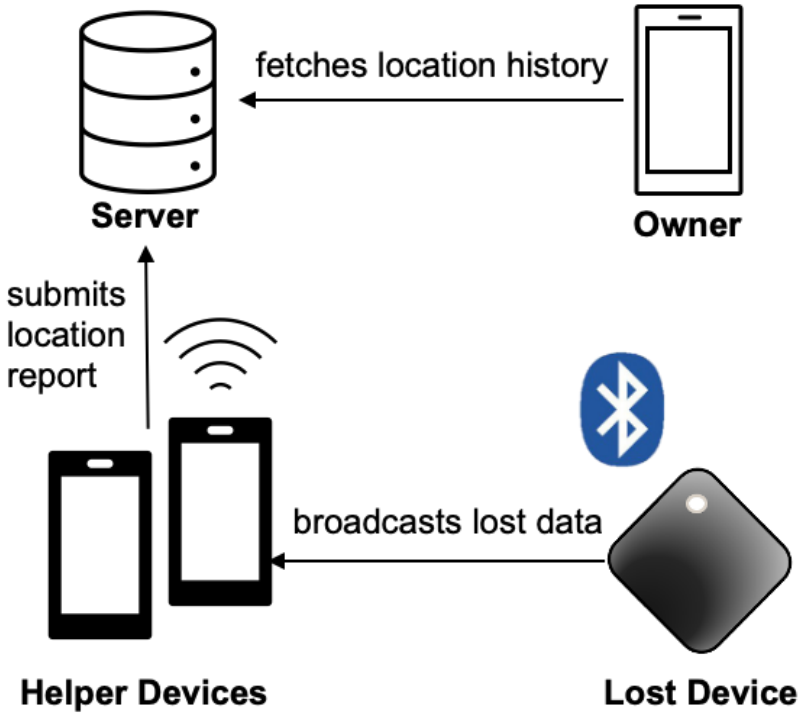
Find My Device Feature:

- Apple's FindMy
- Google's Find My Device
- Samsung's Find My Mobile (FMM)

Offline Finding (OF):

Allows a device without internet connection to be found using:

- Bluetooth Low Energy (BLE)
- Crowd-sourced tracking network



The Galaxy SmartTag

The Galaxy SmartTag

- A BLE tracker released in 2021, a new joiner of Samsung's FMM network
- Extends FMM by allowing owners to track not only their devices, but also personal belongings (attached to the tag).

Figure. SmartTag[1]

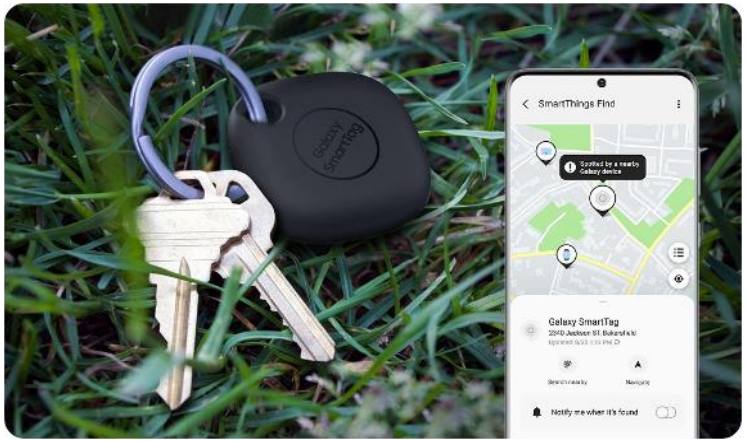


Figure. SmartThings[2] (the client app for FMM devices)

1. https://image-us.samsung.com/SamsungUS/home/mobile/mobile-accessories/phones/09032021/EI-T5300BBEGUS_1_Gallery-Image_1600x1200-jpg.jpg
2. https://images.samsung.com/is/image/samsung/assets/lucid/how-do-i-connect-my-smarttag-to-my-phone/how-do-i-connect-my-smarttag-to-my-phone-header.png

Motivation

Samsung's FMM is one of the largest OF networks in the world. Security or privacy flaws within the network may cause extensive impact...

Research Questions

- (RQ1) Identification of an OF (Offline Finding) device
  - Can an FMM device be identified over BLE?
- (RQ2) Unwanted tracking
  - Can the FMM network be abused for unwanted tracking?
- (RQ3) End-to-end location privacy
  - Can the FMM protocol protect the location privacy from the vendor?
- (RQ4) Location report integrity
  - Can an actor (a helper device or someone outside the FMM network) forge a location report for a lost device?

Outline

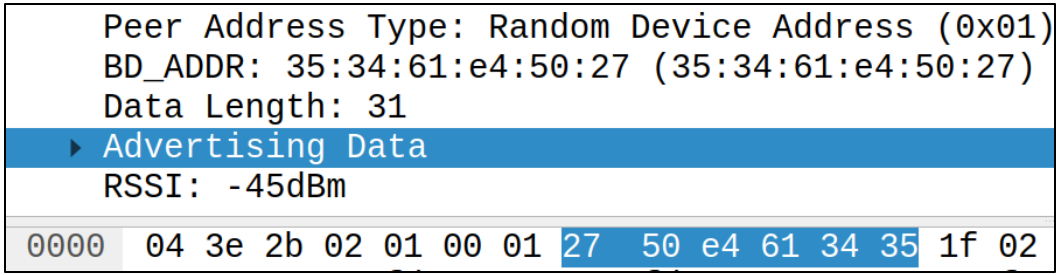
- Introduction
- ➢ **Background & Methodology**
- the FMM Protocol
- Security analysis
- Summary

BLE Protocol Overview

Bluetooth Low Energy (BLE): a wireless communication technology

BLE communication:

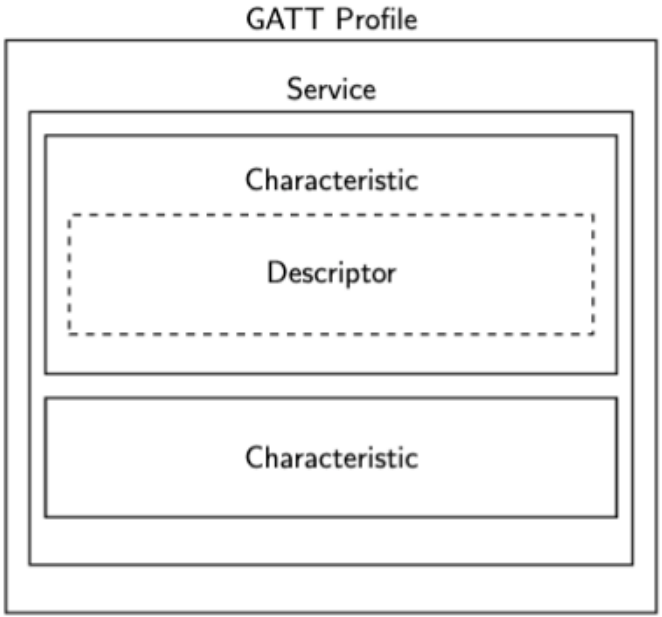
- BLE advertisement



- Data exchange over a connection via GATT (Generic Attribute Profile)

Generic Attribute Profile (GATT)

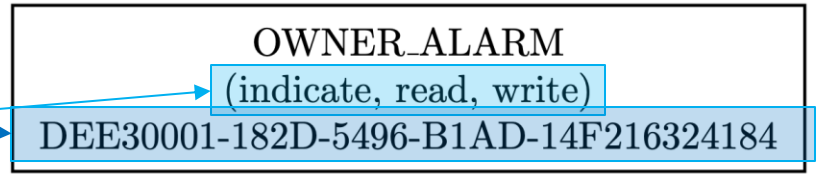
Defines how data is organized and exchanged over a BLE connection.



Generic Attribute Profile (GATT)

A GATT characteristic:

- UUID
- Properties



Exchanging data over a characteristic:

- **Read**: client reads the value of a characteristic from the GATT server
- **Write**: client writes data to a characteristic on the server
- **Indication, Notification**: server pushes data to the client

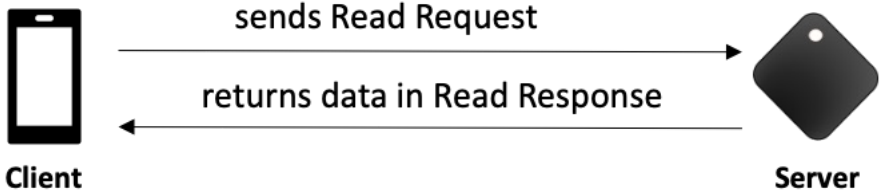


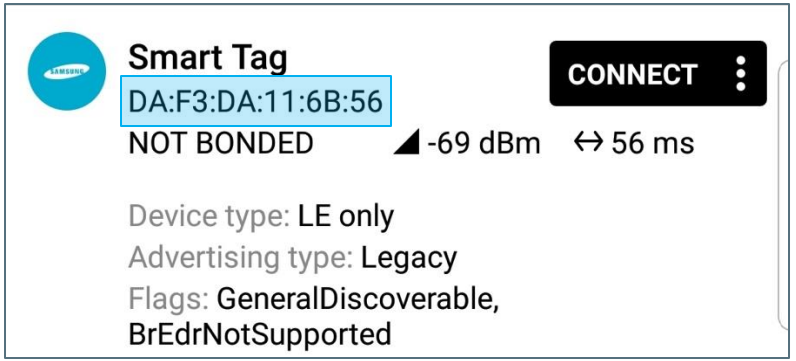
Figure. GATT read example

BLE MAC Address

MAC address: a 6-byte value that uniquely identifies a device.

Types of MAC address:

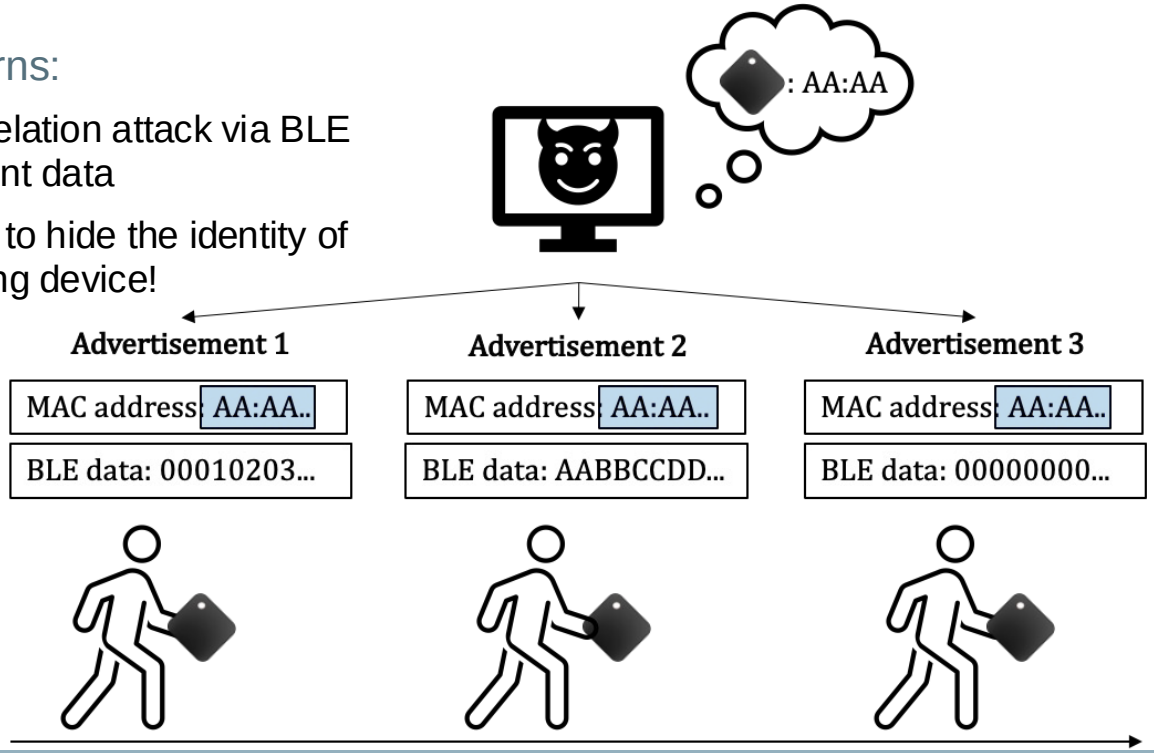
- Dynamic:
  - Resolvable RPA (Random Private Address)
  - Non-Resolvable RPA
- Static:
  - Public Address
  - Random Static Address



BLE MAC Address

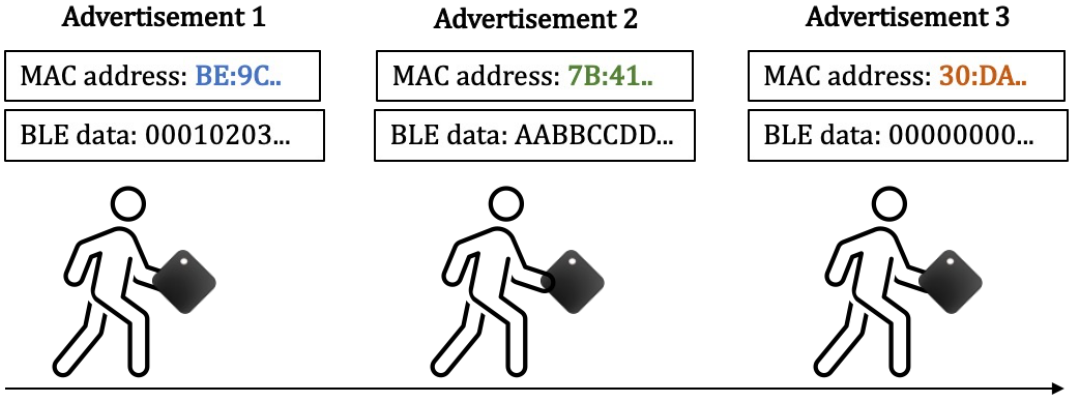
Privacy concerns:

- Identity correlation attack via BLE advertisement data
- Need a way to hide the identity of an advertising device!



LE Privacy Feature

Uses an RPA that re-randomizes at specific timing interval instead of a static address:



A registered SmartTag advertises on an RPA that randomizes every 15 minutes:

”2022-08-31 23:21:01.600”	”1b:8f:d7:d1:be:c6”
”2022-08-31 23:36:31.715”	”3c:58:10:a6:3a:2b”
”2022-08-31 23:52:01.828”	”1d:c2:5e:29:1b:c6”

Methodology

1. BLE passive observation

- passive scanning of BLE data
- extract and analysis HCI snoop log from a Galaxy phone

2. BLE active interaction

- connect to a SmartTag's GATT server
- enumerate and test interesting characteristics/services identified from the HCI snoop log analysis

3. Web traffic analysis

- set up a proxy server
- Intercept/modify the traffic exchanged between the online devices and Samsung's servers

4. APK reverse engineering

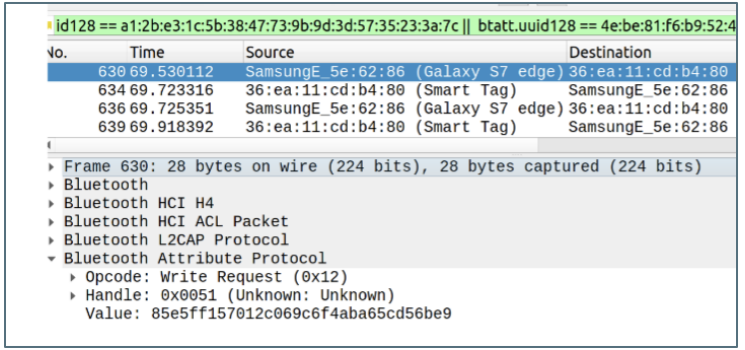
- Use the JADX-GUI decompiler to decompile and analyse SmartThings and FMM (the client apps)

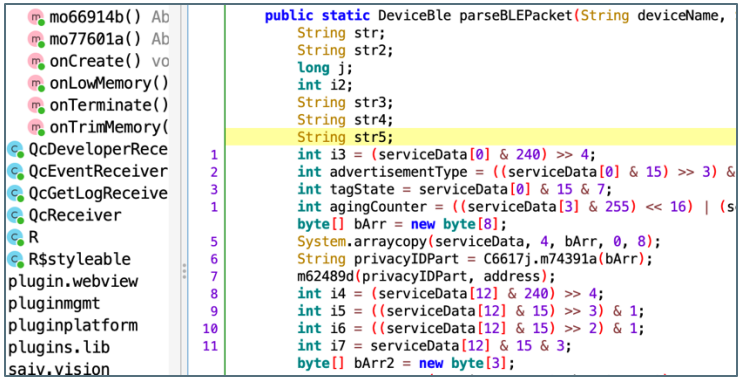
5. Firmware reverse engineering

- Use Ghidra to analyse the dumped SmartTag firmware

6. Dynamic analysis

- Use Frida to analyse the runtime behaviour of the client apps
- dumped the value of the key-pair used for location reports



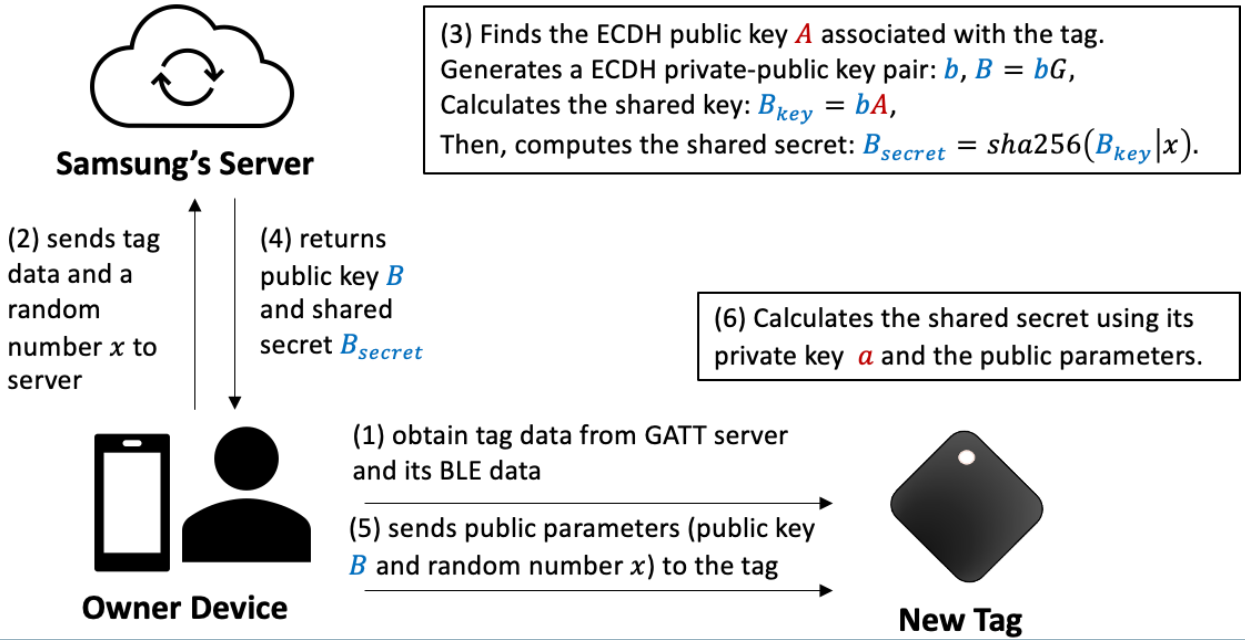


Outline

- Introduction
- Background & Methodology
- **the FMM Protocol**
- Security analysis
- Summary

Device Registration

Establishes a shared secret via ECDH



Registered Device

The shared secret established during the registration process is used to derive 4 AES subkeys via SHA-256:

- $authKey$
- $gattKey$

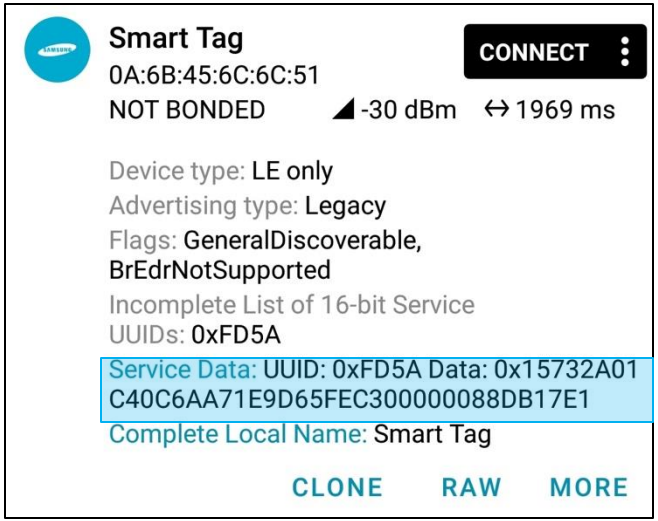
secure communication between owner and tag

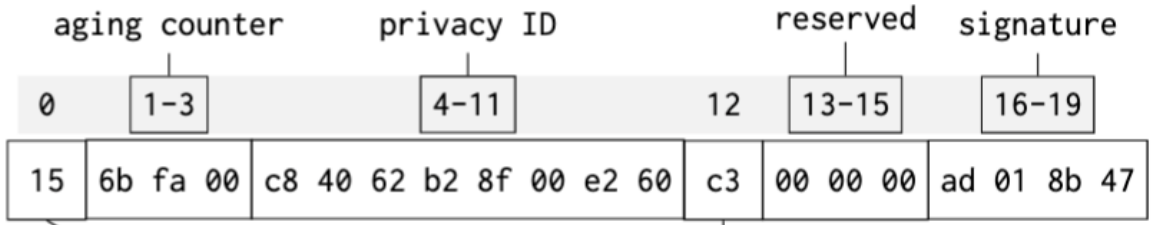
- $pidKey$
- $signKey$

used to generate the BLE data for Offline Finding (OF)

Registered Device

The registered tag broadcasts BLE data in a fixed structure for OF (Offline Finding):





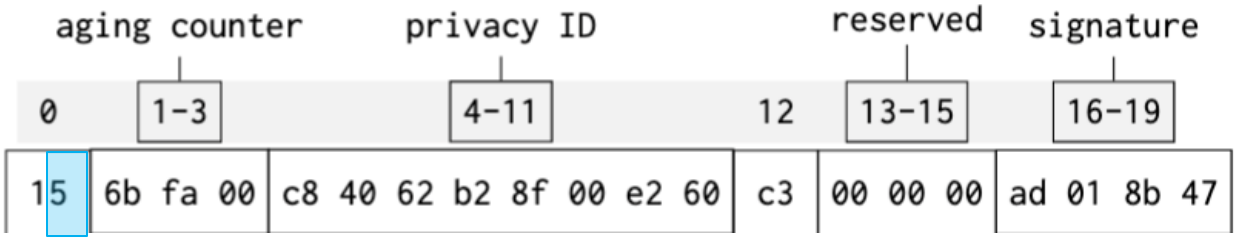
Byte 0: 0x15

Bit	0-3	4	5-7
Data	version	adv type	tag state
Value	0x01	0x00	0x05

Byte 12: 0xC3

Bit	0-3	4	5	6-7
Data	region ID	encryption flag	UWB flag	battery level
Value	0x0c	0x00	0x00	0x03

Registered Device - BLE data

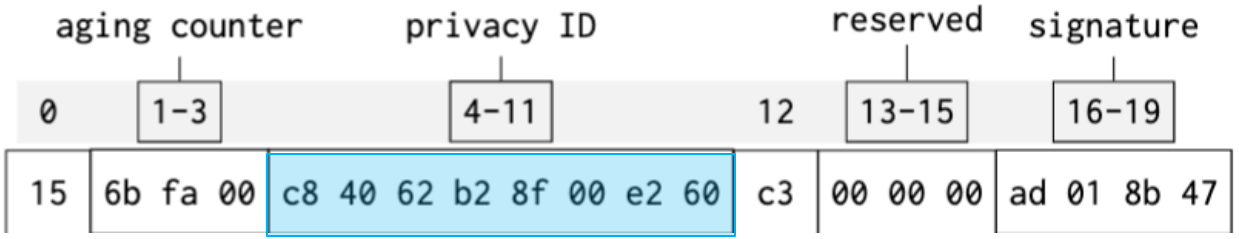


Operating state (1-6)

- state 1: premature offline mode
- state 2: offline mode
- state 3: overmature offline mode
- state 4-6: connected to owner device(s)

Helper devices only reports devices advertising under offline modes (state 2 or 3)

Registered Device - BLE data



Privacy ID

- Uniquely identifies a registered FMM device
- Each FMM device has a Privacy ID pool (a set of unique privacy IDs)

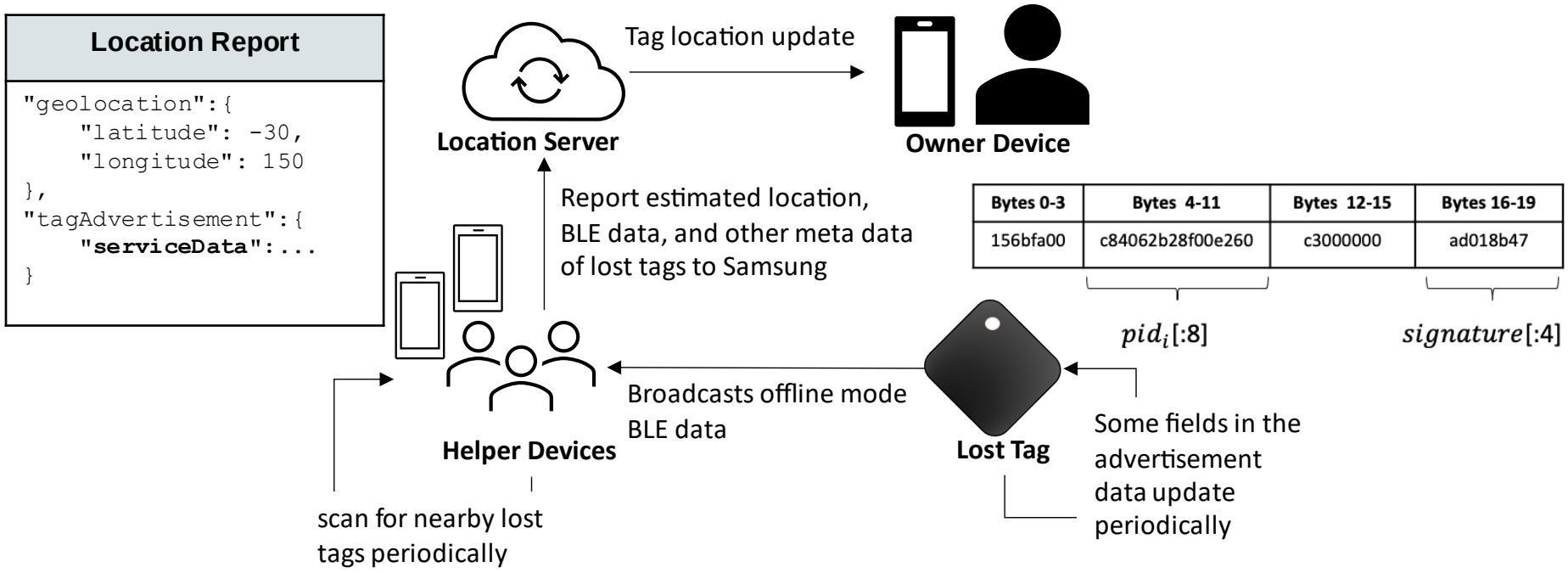
$$PID\ Pool \begin{cases} PID_0 \\ \dots\dots \\ PID_i = E_{pidKey}(privacyIV, \\ \dots\dots \\ PID_n \end{cases}$$

IV

Byte 0	Byte 1	Bytes 2-9	Byte 10	Byte 11
$i \gg 8 \wedge 256$	$i \wedge 256$	Privacy ID seed	$(i \gg 8) \wedge 256$	$i \wedge 256$

Plaintext

Lost-and-Found in FMM



Outline

- Introduction
- Background & Methodology
- the FMM Protocol
- ➢ **Security analysis**
- Summary

Security and Privacy Analysis

Attack surface defined for each RQ

Model	RQ	Assumptions	Capabilities	Attack Scenario
Passive Proximity-based (A1.1)	RQ1	(1) Within BLE communication distance with a tag. (2) Controls a Bluetooth capable device	(1) Record and replay BLE advertisements	Attackers can track neighbours' FMM devices by eavesdropping on BLE advertisements (A1.1) or interacting with the SmartTag's GATT server (A1.2), to infer the presence of a device, thereby revealing their routines.
Active Proximity-based (A1.2)			(1) Interact with tag's GATT server	
Network-based (A2)	RQ4	(1) MitM position between Samsung server and a tag.	(1) Intercept, redirect, or modify network traffic	Thefts can hide their locations by forging location reports using the device's/tag's lost mode advertisement, leading victims on a false trail.
Service Operator (A3)	RQ3	(1) Access to backend systems.	(1) Access to all location reports and secret keys for each registered SmartTag.	Service operators can infringe user privacy by inferring social connections through location history analysis.
Tag Owner (A4)	RQ2	(1) Owns a SmartTag. (2) Access to a Bluetooth capable device. (3) Direct contact with a victim	(1) Hide the tag/customized tracking device in victim's belongings	A tag owner can covertly track a colleague by hiding the tag in their belongings, or create a hard-to-detect customized tracker using Samsung's OF protocol.

(RQ1) Identification of an FMM device

Flaws allowing an FMM device to be identified over BLE

1. various readable GATT characteristics leaking identifiable data
2. the DFU (Device Firmware Update) characteristic for SmartTags allows any connected device to reboot the tag, revealing its static address
3. small privacy pool (size 50) for FMM mobile devices. Allows a proximity-based attacker to collect all the privacy IDs within a short period of time, then perform correlation attacks

Impact: defeating the purpose of the LE privacy feature, which aims to protect a tag's long term identity using RPAs (Random Private Addresses)

(RQ2) Unwanted tracking

Existing anti-tracking algorithms for SmartTags:

- Samsung's in-built feature:
  - requires the user to perform manual BLE scanning
  - displays any Overmature mode (state 3) tags detected from the scan
- AirGuard[1] by SEEMOO Lab:
  - runs BLE scanning in the background
  - detects Overmature mode tags

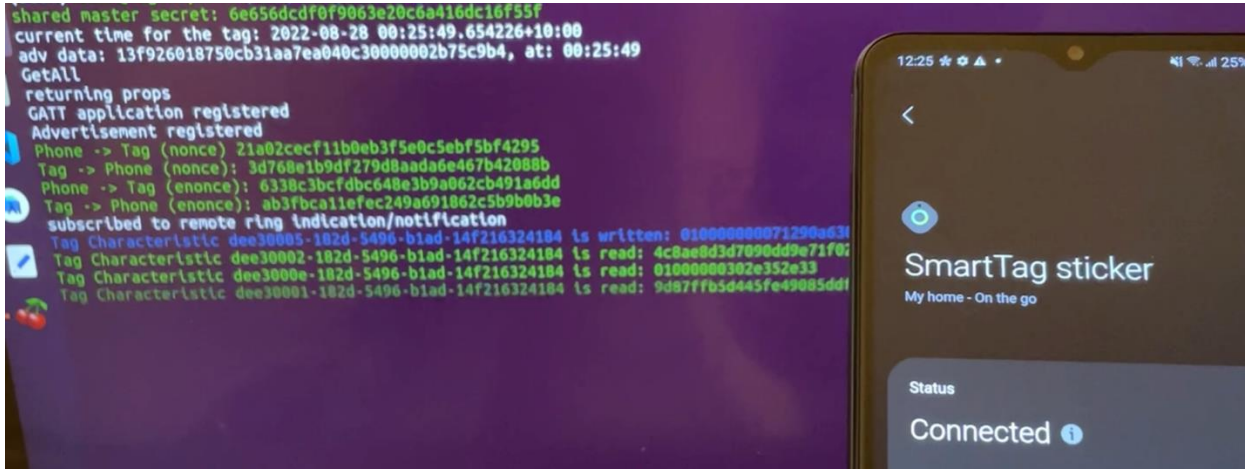
A SmartTag operates under Offline mode (state 2) for the first 24h after it is lost, then transition to Overmature Offline mode.

Existing algorithms can only detect trackers after being tracked for 24h.

1. https://github.com/seemoo-lab/AirGuard

(RQ2) Unwanted tracking

An impersonated tag operates the same as a legitimate tag (video link). The user can customise its BLE behaviour by specifying its tag state and MAC-payload rotation interval.

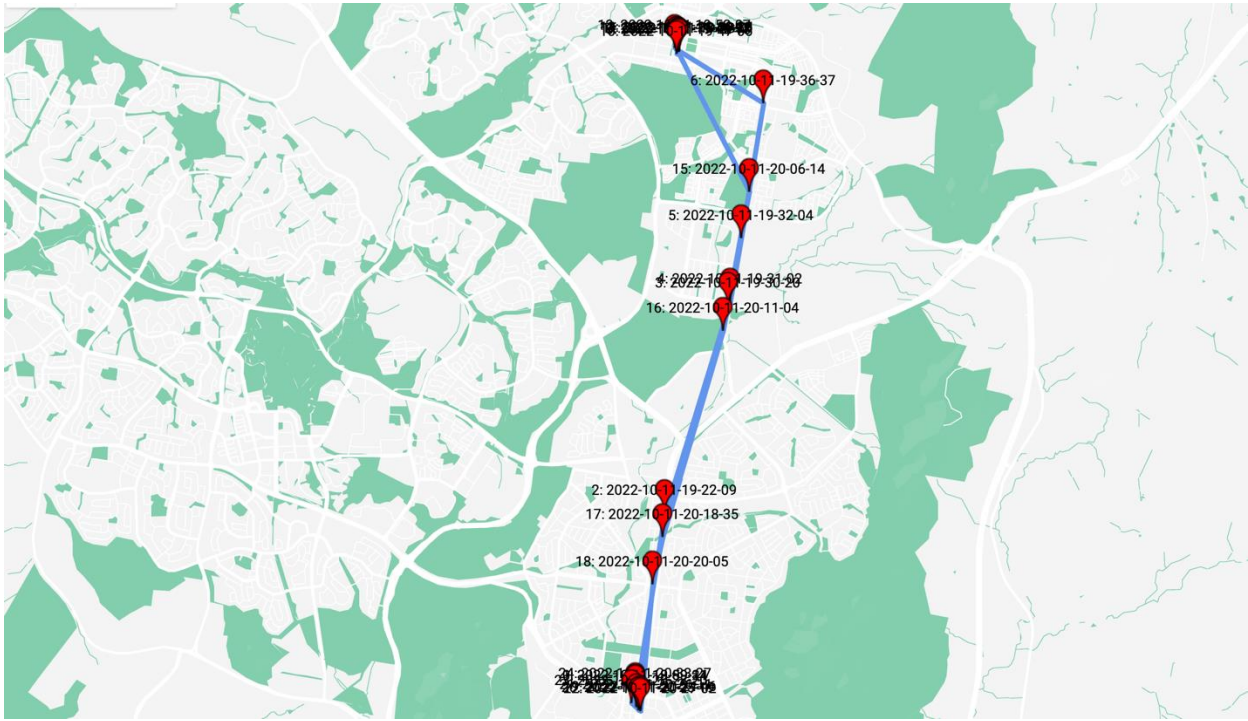


Impact: an attacker can bypass both anti-tracking algorithms by simply configuring the impersonated tag to always advertise on Offline mode

(RQ2) Unwanted tracking

Tracking experiment

Estimated path plotted from the location history returned by the Samsung's server



(RQ3) End-to-end location privacy

Vendor knows necessary key materials for computing the privacy IDs of any registered device

- vendor has the ECDH public key (A) for every device
- vendor generates the ECDH key pair (b, B) on owner's behave

Samsung's Server

(3) Finds the ECDH public key A associated with the tag.
Generates a ECDH private-public key pair: b, $B = bG$,
Calculates the shared key: $B_{key} = bA$,
Then, computes the shared secret: $B_{secret} = sha256(B_{key}|x)$.

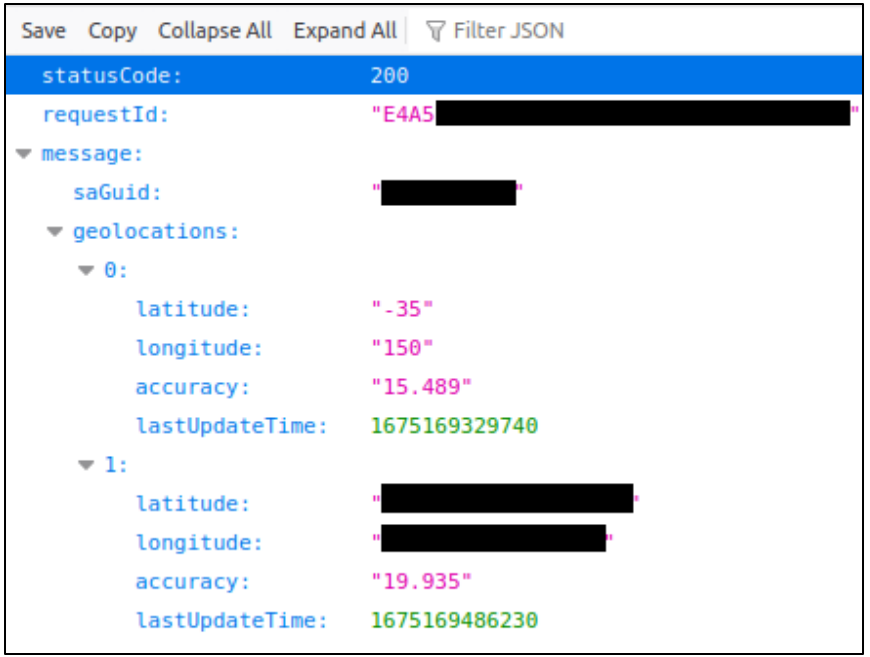
Any key material that an owner device or a FMM device can derive can also be derived by the vendor

(RQ3) End-to-end location privacy

Location history response

- Location history response from Samsung consists of a list of geolocations in plaintext.
- Vendor links each location report to the owner's account based on the privacy ID contained in the report.

Impact: no end-to-end location privacy

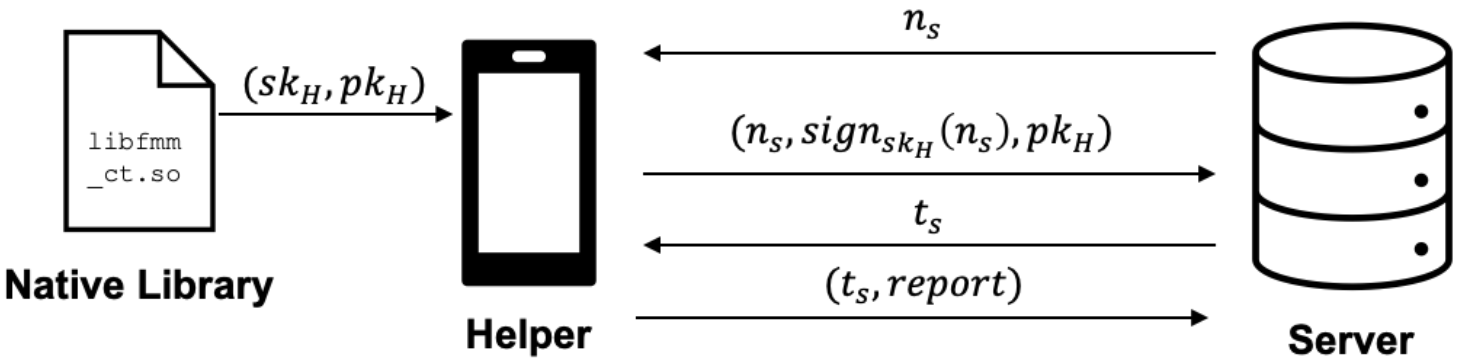


(RQ4) Location report integrity

Location report protocol

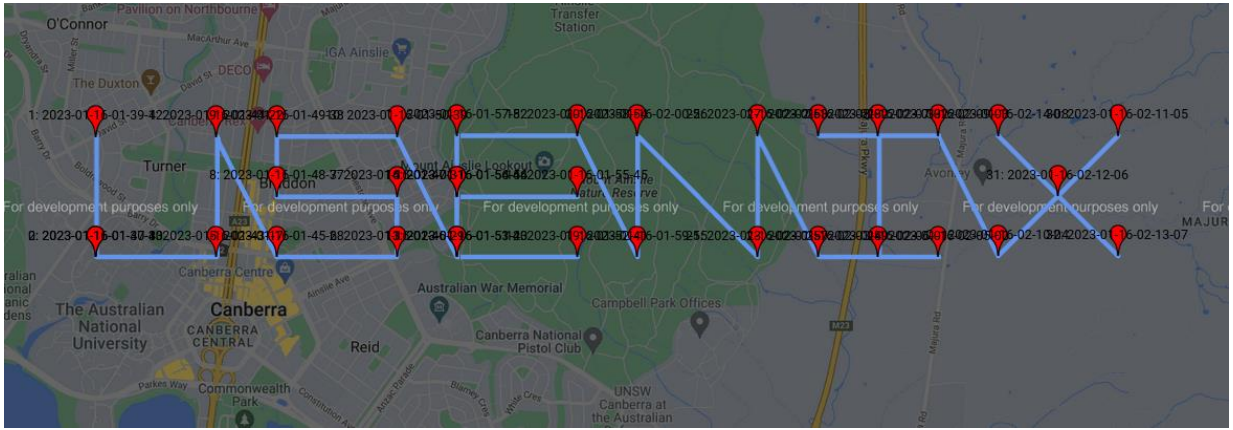
Uses a signing key to authenticate a helper device to the server.

`libfmm_ct.so`, a native library file contained in the SmartThings APK, contains a default signing key pair (sk_H, pk_H) that can be extracted through runtime memory analysis.



(RQ4) Location report integrity

Location report forgery



Impact: An attacker **without** a galaxy device can obtain the access token and submit forged location reports

Outline

- Introduction
- Background & Methodology
- the FMM Protocol
- Security analysis
- **Summary**

Summary

- (RQ1) Identification of an FMM device
  - proximity-based attacks: GATT leaking identifiable data, DFU reboot, ...
- (RQ2) Unwanted tracking
  - unwanted tracking via FMM device emulation
- (RQ3) End-to-end location privacy
  - lack of end-to-end privacy: lost device locations are not protected from the vendor
- (RQ4) Location report integrity
  - leaked signing key pair allows actors outside the network to report locations

Read our paper for protocol details and more attacks on FMM!