



DFRWS EU 2025 - Selected Papers from the 12th Annual Digital Forensics Research Conference Europe

## Samsung tracking tag application forensics in criminal investigations



Hongseok Yang, Sanghyuk Han, Mindong Kim, Gibum Kim \*

Dept. of Forensic Sciences, Sungkyunkwan University, 25-2 Sungkyunkwan-Ro, Jongno-Gu, Seoul, 03063, South Korea

### ARTICLE INFO

#### Keywords:

Tracking tag  
IoT forensics  
Android forensic  
Anti-forensics  
Stalking

### ABSTRACT

With the advancement of offline Finding Network (OFN) technology, tracking tags are being utilized in various fields, including locating elderly individuals with dementia, caring for children, and managing lost items. Recently, however, tracking tags have been misused in stalking, surveillance, and debt collection, highlighting the growing importance of digital forensics in proving criminal acts. While there has been some research on Apple AirTag and Tile products, studies focusing on Samsung's tracking tag have been lacking. Therefore, this paper proposes digital forensic techniques for law enforcement agencies to analyze Samsung tracking tag applications to identify perpetrators and substantiate criminal activities. We analyzed six tags and three applications, recognizing tag identifiers, and confirmed that location data is stored in both plaintext and encrypted forms within SQLite databases and XML files. Additionally, we conducted experiments on five different anti-forensics scenarios: 1) deletion of a registered tracking tag, 2) deletion of location data, 3) account logout, 4) service withdrawal, and 5) application synchronization, finding meaningful results to substantiate criminal actions. Furthermore, we developed S.TASER (Smart Tag Parser) based on Python that allows for the identification of deleted tags, recovery of identification data, and visualization of collected location data per tag. S.TASER's code, experimental scenarios, and raw data are publicly available for further verification. This study aims to contribute to the global digital forensic industry by suggesting additional options for investigation and evidence gathering of crimes that make use of Network.

### 1. Introduction

A tracking tag is a coin-sized Bluetooth Low Energy (BLE) device that, when logically connected to a smartphone, allows real-time tracking of the tag holder's location. It is used to prevent the loss of belongings and to locate missing children or dementia patients. Tile first released its product in 2013, and Samsung and Apple entered the market in 2021 with the development of the Galaxy SmartTag and AirTag. As of May 2023, the number of registered Samsung SmartThings devices exceeded 300 million, while approximately 55 million AirTags and 40 million Tile devices had been sold by 2022, making these services widely popular. The tracking tag market, valued at \$1.78 billion in 2022, is projected to grow to \$16.7 billion by 2031 (IMIR, 2023).

However, tracking tags have been misused in various crimes such as stalking, illegal tracking, sexual offenses (People v. Molina, 2024), murder, and vehicle theft. In the UK, there has been research on AI-assisted stalking crimes, where artificial intelligence learns data from social media and devices to monitor an individual's location and activities (Caldwell et al., 2020). In the U.S., states like Pennsylvania

(Neely, 2024), Ohio (Owen, 2022), and Florida (Scheckner, 2024) have pending legislation against installing or tracking someone with a tracking tag without their consent. In South Korea, the anti-stalking law was amended in 2023 to include acts committed via information and communication networks, but it still falls short of covering tracking tags. As tracking tags are increasingly used in crimes and the scope of relevant criminal penalties expands, the importance of evidence analysis for tracking tags in criminal investigations is increasing whereas related research is falling behind. In particular, despite the global use of Samsung tracking tag, it has not been dealt with thoroughly compared to AirTag and Tile.

Therefore, this study aims to analyze the Samsung tracking tag applications to identify artifacts and propose digital forensic techniques to substantiate user actions. In Chapter 2, the operation principles of an offline finding (hereafter referred to as 'OF') network, as well as Samsung tracking tag are examined, along with literature review. In Chapter 3, eight experimental scenarios which were designed through interviews with three experienced forensic analysts are suggested, and functions of the applications are analyzed. In Chapter 4, the types and storage

\* Corresponding author.

E-mail addresses: [ininondumak@gmail.com](mailto:ininondumak@gmail.com) (H. Yang), [claratus32@gmail.com](mailto:claratus32@gmail.com) (S. Han), [kim.md0925@gmail.com](mailto:kim.md0925@gmail.com) (M. Kim), [freekgb02@gmail.com](mailto:freekgb02@gmail.com) (G. Kim).

<https://doi.org/10.1016/j.fsidi.2025.301875>

locations of artifacts, and the artifacts for each scenario are identified and verified. In Chapter 5, this paper suggests a newly devised tool for rapid analysis of Samsung tracking tags.

## 2. Related work

### 2.1. Operating principles

A tracking tag is designed to enable real-time location tracking of objects, maintaining functionality over an extended period while providing accurate and efficient tracking. The core technology of the tracking tag is BLE.

BLE enables communication between low-power devices, such as smartwatches and sensors. Compared to traditional Bluetooth Classic, it has a shorter communication range and slower processing speed. However, in theory, a coin cell battery can power BLE devices for up to 14 years (Gomez et al., 2012).

The GAP (Generic Access Profile) protocol operates within BLE and serves four key roles (Yu, T. et al., 2024). The ‘Advertiser’ broadcasts its data broadly, while the ‘Observer’ processes advertisements without establishing a connection. The ‘Central’ receives advertisements and attempts to connect with devices for 2-way communication, while the ‘Peripheral’ broadcasts data widely and accepts connection attempts from the Central.

While a tracking tag allows for precise location tracking, devices outside the BLE communication range cannot be tracked. To overcome the limitations of BLE communication, the concept of network was introduced, utilizing crowd-sourcing technology (Fig. 1.). When a tracking tag is lost, it periodically emits a BLE signal to broadcast its location. A ‘Helper’, which is the device closest to the tracking tag and part of the OF network, receives this signal and sends it to the OF network server. The tracking tag’s owner can then check the uploaded location data through the SmartThings Find app. The more devices participate in an OF network, the more accurate and faster the location tracking becomes (Yu et al., 2022).

Various tracking tags have been released, including Apple’s AirTag, Tile’s Tile Pro, and Samsung’s SmartTag2, each with different performance characteristics. The AirTag has a relatively short maximum detection range for BLE, but it benefits from the large number of users worldwide, providing a favorable operational environment (Singh, 2024). Tile’s Tile Pro can be used on both Android and iOS devices, and although it does not support Ultra Wide Band (UWB), such shortcoming is compensated by integrating its own network with the Amazon Sidewalk network, enhancing indoor location tracking and the performance of its OF network. Samsung’s SmartTag2 has a comparatively wider BLE

detection range and extended battery life.

Additionally, Samsung smartphones have the largest number of active users among Android OS devices (Onyango, F., 2024), which contributes to the superior performance of its OF network (Table 1).

### 2.2. Prior research

#### 2.2.1. Vulnerability analysis

Mayberry, T. et al. (2021) proposed an attack method that bypasses the ‘Item Safety Alert’ feature introduced to prevent stalking with AirTag and argued for a redesign of the protocol. They demonstrated how to modify a tag using the same chipset as the AirTag and manipulate the transmitted message using the bit-flipping technique to make it recognizable as an iPhone. They also outlined methods such as rotating a single predefined public key to multiple keys over time to enable prolonged tracking, or generating keys in real-time to avoid triggering the ‘Item Safety Alert’ notification and track the victim.

Roth, T. et al. (2022) successfully modified the firmware of an AirTag and cloned the tag using a ‘voltage glitching’ attack, which involves rapid fluctuation of the chip’s power supply. AirTag has a stalking prevention feature that provides smartphone notifications or AirTag alerts when someone else’s AirTag is nearby. However, by removing the built-in speaker and altering the firmware, the researchers were able to bypass this feature.

Yu, T. et al. (2022) analyzed Samsung’s Find My Mobile (FMM), which is a part of SmartThings Find, to determine whether information related to the device and tag owner could be identified and if location data could be tampered with. They conducted BLE and web traffic analysis, device log analysis, firmware and APK reverse engineering, and successfully identified information that could infer the device and tag owner. They also managed to track the location of the tag and the

**Table 1**  
Specifications of major tracking tags.

Product Name	AirTag	Tile Pro	SmartTag2
Release Year	2021	2024	2023
Compatible Device OS	iOS	Android and iOS	Android
OF Network	Find My Network	Tile Network	Galaxy Find Network
Max. Detection Range (BLE)	10 m	122 m	120 m
UWB Support	Yes	No	Yes
Max. Continuous Usage Time	1 year	1 year	1.4 year

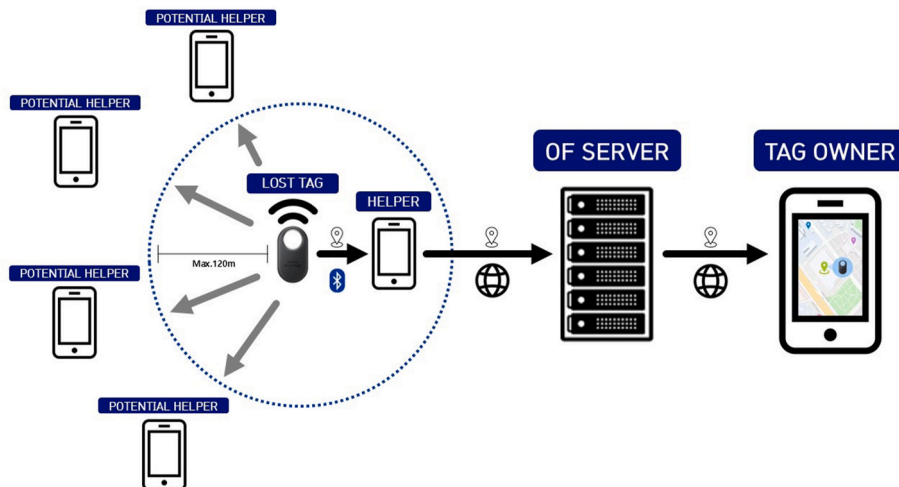


Fig. 1. Overview of the operation of an OF Networks.

'Helper' using the same OF network server and manipulate the location data. In a later revised paper (Yu, T., 2024), it was noted that Samsung addressed most of these issues. However, they confirmed that vulnerabilities could still be exploited on smartphones and tracking tags using outdated software and firmware versions, since they can still participate in an OF network.

Alamleh, H. et al. (2024) conducted various attacks, including physical manipulation and firmware exploitation, to analyze the vulnerabilities of AirTag and SmartTag. AirTag was found to be secure because the location data is decrypted on the user's device, but it was vulnerable to spoofing attacks. SmartTag, on the other hand, decrypts location data on the server, allowing server operators to access the decrypted information, making it more vulnerable from a data leakage perspective. Both tags were designed with a focus on battery life, making system patches difficult, and neither had a secure boot process, which further weakened their security.

### 2.2.2. Artifact analysis

Appalachian4n6 (2022) identified AirTag information, owner details, and the last known location's address and latitude/longitude coordinates left in the Items.data file of the Find My app on macOS and iOS devices. Chris Vance (D20 Forensics, 2022; Magnet Forensics, 2022) discovered data related to AirTags in iOS's Unified Logs, including serial numbers, UUIDs, key storage locations, pairing dates, and user-assigned names. Additionally, several data files within the Find My app path contained information such as other Apple IDs associated with the owner, details of AirTags and other tags, and information about other iOS and macOS devices linked to the Apple ID.

Binary Hick (2022) analyzed the AirTag detection applications 'Tracker Detect' and 'AirGuard' on Android OS. Tracker Detect deleted AirTag detection logs upon system reboot, while AirGuard stored data such as the AirTag's MAC address, last known location, the first and last detection timestamps, signal strength at detection, and latitude/longitude data in a database. Using Google's "Unknown Tracker Alerts" feature (Android Help, n.d.), they identified AirTag device information, location, and owner details in specific databases (Binary Hick, 2023). They also confirmed the presence of artifacts and data related to tags registered in Google's Find My Device app on iOS (Binary Hick, 2024-a). Finally, Binary Hick (2024-b) organized the information that could be accessed by both the owner and the victim in a simulated stalking scenario using a tag registered in Google's Find My Device app. This included details available from the perspective of the tag owner as well as the information visible to the victim.

Pace, L. et al. (2023) discovered that the Tile application stores tag-related information across various operating systems, including iOS, Android, and Windows. They identified artifacts such as the tag information, latitude/longitude coordinates, and timestamps are stored in iTunes backups, Android backups, and Windows memory images.

Additionally, they developed an open-source tool called TAP (Tile Artifact Parser), which assesses the likelihood of location tampering by analyzing the plausibility of the speed derived from the tag's locations and timestamps before and after movement.

To date, research on tracking tags has predominantly focused on vulnerability detection, with forensic investigations remain notably underexplored. Although substantial work has been conducted on Apple and Tile products, there remains a significant gap in the forensic examination of Samsung SmartTag, particularly within the Android OS ecosystem.

## 3. Experimental design

### 3.1. Purpose and scenarios

This study aims to analyze the Samsung tracking tag application to identify artifacts of user-registered tag information and collected location data, offering clues for crime investigation. To develop

experimental scenarios, 17 legal cases (Gwangju District Court Decision, 2021godan4187 et al.) from Korean courts involving the use of tags were collected and analyzed. The analysis focused on identifying the models of tags used and the functionalities of the associated applications. The findings revealed a common pattern; criminals registered tags via the application, attached them to the target's belongings and tracked their location. This pattern was incorporated into experimental scenarios. Additionally, the scenarios also considered the application features that could potentially be leveraged for anti-forensic purposes, such as tag deletion, location data deletion, account logout, and service withdrawal. To refine the scenarios, feedbacks were obtained from three analysts with experience in tracking tag analysis within the Korean Police via a questionnaire, which resulted in the final selection of three general scenarios for default ordinary user and five anti-forensics scenarios. Three general scenarios include analysis of (1) basic artifacts and their structures (2) artifacts of tracking tag registration, (3) artifacts of location data retrieval. Five anti-forensic scenarios analyze artifacts of (4) registered tracking tag deletion, (5) location data deletion, (6) account logout, (7) service withdrawal, and (8) application synchronization. Smartphone factory reset and application deletion, with no possibility of data recovery, were excluded from the scenario.

### 3.2. Experimental environment

The experiment was conducted using Samsung Galaxy smartphones and focused on the following applications: SmartThings (hereafter referred to as ST) for tracking tag management, SmartThings Find (a plugin of ST, hereafter referred to as STF) for location tracking, and Samsung Find (hereafter referred to as SF). To use the Samsung tracking tag, it is mandatory to install ST and STF, while SF is optional. However, to compare the artifacts generated by each application, all three were installed for the experiment.

As the experiment focused on the application, the smartphone models and operating systems did not affect the results. However, for reliability, and the robustness of the experiment, the latest model (SM-S901N) was included.

To examine whether artifacts change based on the type of tag, various tags that can be registered in ST were included. To determine the existence of artifacts that may be discovered during the process of registering multiple tags of the same model, two units of the EI-T5600 (0AFD, 452) were prepared. Detailed information on devices and software used are shown in Table 2, while the experimental scenario and target applications are listed in Table 3.

The experiment utilized a total of eight tools, including Magisk, Frida, and Jadx, for smartphone rooting, dynamic analysis, and static analysis of the applications, see Table 4.

Based on the analysis of South Korean court rulings and expert interviews, the web versions of SmartThings apps were excluded from the scope of analysis, as they have not yet been used in any of the previous criminal cases.

### 3.3. Experimental procedure

The experiment commenced with a factory reset of smartphones. Smartphones for scenario (2) and (3) were rooted to analyze encrypted data and acquire network data between the smartphone and the server.

The tags were in motion while conducting actions related to the user's general and anti-forensic activities. To confirm the accuracy of the location information artifacts from the application, actual location data was collected through a separate GPS application at the exact same location of the tags.

To monitor changes of the collected artifacts, data from the target applications was collected via the Android Debug Bridge (ADB) shell and the commercial software (MD-NEXT) both before and after the user's actions.

In scenarios (2) and (3), network data between the smartphone and

**Table 2**  
Experimental devices and software.

Category	Manufacturer	Name	Version	Unit	Purpose
Smartphone	Samsung	SM-A600N	Android 10 (Rooted)	2	Scenario 2, 3
"	"	"	Android 10	3	Scenario 1, 8
"	"	SM-S901N	Android 13	1	Scenario 1, 4, 5, 6, 7
Application	"	SmartThings (ST)	1.8.18.21, 1.8.21.28	N/A	Tag management
"	"	SmartThings Find (STF)	1.8.25-3, 1.8.27-10	N/A	Location retrieval
"	"	Samsung Find (SF)	1.3.12, 1.4.00.10	N/A	"
Tag	SOLUM	SOLUM SMART TAG	CS06BHB01D (0A6W,009)	1	Location data collection
"	Samsung	Galaxy SmartTag	EI-T5300 (0AFD,435)	1	"
"	"	"	EI-T5300 (0AFD,430)	1	"
"	"	Galaxy SmartTag2	EI-T5600 (0AFD,451)	1	"
"	"	"	EI-T5600 (0AFD,452)	2	"

**Table 3**  
Experimental scenario and target applications.

No.	Experiment type	Target application			Experiment summary
		ST	STF	SF	
1	Basic artifact structure	○	○	○	Tag registration, location data retrieval
2	Tracking tag registration	○	○	N/A	Tag registration, deletion, re-registration and network packet collection
3	Location data retrieval	○	○	○	Location data retrieval through STF and SF, network packet collection
4	Registered tracking tag deletion	○	○	○	Registered tag deletion through ST
5	Location data deletion	○	○	○	Location data deletion through STF and SF
6	Account logout	○	○	○	Account logout through ST
7	Service withdrawal	○	○	○	Withdrawing from the SmartThings service through ST
8	Application synchronization	○	○	○	Comparison of results after location data deletion and STF and SF synchronization in multi-device environment

**Table 4**  
Analysis software.

No.	Manufacturer	Name	Version	Purpose
1	John Wu	Magisk	27.0	Rooting
2	Google LLC	Android Debug Bridge	1.0.41	Data extraction
3	GMDSOFT	MD-NEXT	2.1.13.2492	"
4	Frida Core LLC	Frida	16.4.7	Dynamic analysis
5	Skylot	Jadx	1.5.0	Static analysis
6	Toolshed Labs SLU.	HTTP Toolkit	1.19	Network analysis
7	DB Browser for SQLite team	DB Browser for SQLite	3.13.0	Database analysis
8	BasicAirData	GPS Logger	3.2.2	Collecting GPS info

the server was collected using HTTP Toolkit to examine the tag registration process and the location data retrieval process.

Data obtained before and after the user's actions were compared to analyze the artifacts and their changes. The experiment assumed a scenario in which the analyst had access to the encryption key in the Android Keystore to discover all artifacts associated with each application.

The identified encrypted materials, such as location data, were decrypted through dynamic analysis of the application using Frida. The detailed activities performed by the user for each scenario were documented in GitHub repository. (S.TASER).

## 4. Experimental results

### 4.1. General scenarios

#### 4.1.1. Basic artifact structure

The result artifacts consists of tag identification data and location data, and no significant differences between target applications are observed.

Artifacts are stored in SQLite databases, XML files, and cache files, and the storage paths for each application are found as in Table 5. Specifically, the identification data are categorized into deviceId, model name, and tag name, while location data are categorized into latitude, longitude, and address. The application artifact structure is found as in Table 6.

Differences are found in the artifact structure depending on the version of the target application. The structure of the previous versions (ST:1.8.18.21, STF:1.8.25-3, SF: 1.4.00.10) are available on S.TASER's GitHub repository.

#### 4.1.2. Tracking tag registration

An analysis of the network data collected during the tag registration process shows that a tracking tag is registered in the following order: ① search for nearby tag, ② obtain registration information per model, ③ check for duplicate identifiers, ④ complete registration, and ⑤ retrieve information for the registered tag, see Table 7.

During the tag registration process, the model name (modelName), manufacturer ID (mnlD), model ID (setupId), the user-defined name (label), the logical identifier (logId), and the UUID (deviceId) assigned to the registered tags were used to identify the tag.

The Serial Number (physical identifier) on the surface of the tag was not found during the registration process, see Fig. 2.

The identification data, including the logId (identifier), and deviceId, serves as unique values that can specify tag users. The logId is a fixed value for the tag, while the deviceId is a dynamic value that changes with each registration, even for the same tag. The setupId enables the identification of specific device details, such as tag color, even when the model (modelName, marketingName, etc.) is the same. This information can be an additional hint in criminal investigations, as it provides crucial data that can help differentiate between devices in various contexts. For example, in Fig. 3, although two devices have identical model name or marketing name, the product color can be identified through the setupId

**Table 5**  
Application artifact path.

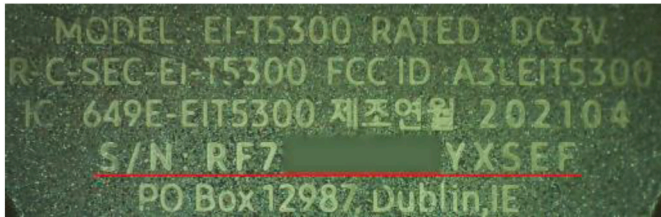
Application	Path	Sub path	Type
ST/STF	data/data/com.	databases	SQLite
	samsung.android.oneconnect	shared_prefs cache	XML cache
SF	data/data/com.samsung.android.app.find	databases	SQLite

**Table 6**  
Application artifact structure.

Database (Table)	deviceId	label	mnId	setupId	modelName	logId	timestamp	GeoInfo	App
DataLayerData.db (DeviceDomain)	○	○	○	○	○	○	○	–	ST/ STF
DataLayerData_core.db (DeviceDomain)	○	○	○	○	○	○	○	–	"
DataLayerData.db (BleDeviceCapabilityStatusDomain)	○	–	–	–	–	–	–	–	"
DataLayerData_core.db (BleDeviceCapabilityStatusDomain)	○	–	–	–	–	–	–	–	"
PersistentLogData.db (PersistentLogDomain)	○	○	○	○	○	○	○	–	"
Fme.db (FmeAppData)	○	○	–	–	–	–	–	○	"
InternalSettings.db (insettings)	○	–	–	–	–	–	–	–	"
EasySetupIconNameDb.db (EasySetupIconDb)	–	–	○	○	–	–	○	–	"
FME_SELECTED_DEVICE.xml	○	○	–	–	–	–	–	○	"
cache Files	○	–	○	○	–	○	–	–	"
com.samsung.android.pluginplatform.pluginbase.sdk. PluginSQLiteOpenHelper. [AppId].location_history * Encrypted	○	–	–	–	–	–	–	○	"
app-database.db (DeviceDomain)	○	–	–	–	–	–	–	○	SF
find-sdk (FmeAppData) * Encrypted	○	–	–	–	–	–	–	○	"

**Table 7**  
Artifact during the tag registration process.

Order	Action	Identification data
1	Search tag	mnId, setupId
2	Obtain registration info	mnId, setupId
3	Check for duplicate	mnId, setupId, logId, modelName
4	Registration completion	deviceId, modelName, label, mnId, setupId, logId
5	Information retrieval	deviceId



**Fig. 2.** Serial Number on the surface of the tag.

```

1 SELECT modelCode, marketingName, mnId, setupId \
2 FROM CatalogDevices where marketingName = 'Galaxy SmartTag2'

```

	modelCode	marketingName	mnId	setupId
1	EI-T5600BBEGKR	Galaxy SmartTag2	0AFD	452
2	EI-T5600BWEGKR	Galaxy SmartTag2	0AFD	451

**Fig. 3.** An example of the setupId's meaning.

(452 = black, 451 = white).

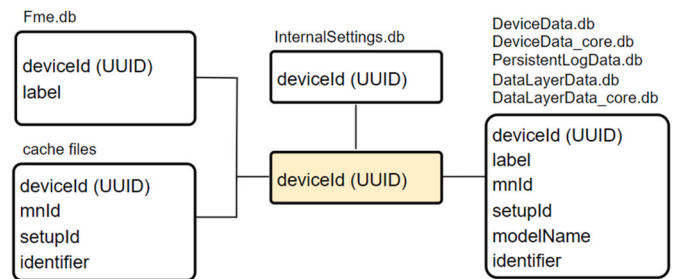
The key value that links the tag's artifacts is the deviceId, which connects individual elements such as the DB tables, as shown in Fig. 4.

It also serves as the reference point for recovering information from deleted tags.

#### 4.1.3. Location data retrieval

Location data is stored in string format, including timestamp, deviceId, latitude, longitude, accuracy, and address. The applications differentiate between data collected over a maximum period of one week and the last known location data.

The data (com.samsung.android.pluginplatform.pluginbase.sdk.PluginSQLiteOpenHelper. [AppId].location\_history hereafter referred to



**Fig. 4.** The artifact structure centered around the deviceId

as location\_history) collected by STF over a week is encrypted, so if an analyst cannot access the key, they can only retrieve the last known location from STF. In contrast, SF records location data in plain text for up to 7 days (Table 8.).

The encrypted STF location data and the plaintext SF location data are identical. The detailed DB and network data are available on S. TASER's GitHub repository.

In the encrypted location database table (EncLocationHistory) of STF, the encDeviceid (encrypted deviceId) value combines the timestamp with the last four digits of the plaintext deviceId (UUID) of the tag (as shown in Fig. 5). Without the decryption key, this table does not reveal specific location data. However, it allows analysts to roughly track how many times a user collects the target's location data on a daily basis.

## 4.2. Anti-forensics scenarios

### 4.2.1. Registered tracking tag deletion

Once ST is installed, the deviceId information of all registered tags are retained in the InternalSettings.db, regardless of the deletion of the registered tags. The deviceId of active tags are additionally stored in DataLayerData.db. By comparing the results from these two databases, it is possible to identify the deviceId of deleted tags. To identify a real-world tag registered through the application, the logId (identifier) is required. The logId can be confirmed through the following three methods: analyzing application logs, tag registration databases, and

**Table 8**  
Location data artifacts.

Application	Long-term	Last known
ST/STF	location_history *Encrypted	FME_SELECTED_DEVICE.xml Fme.db
SF	app-database.db	find-sdk *Encrypted

	encDeviceid	deviceid
1	1733058838564366c	db348cf2-ddae-4eb8-be3f-62cfeccd366c
2	1733058838564b332	2a6a413d-e33b-48e3-a955-58afa0e3b332

Fig. 5. UUID and encDeviceid’s last 4 characters comparison.

cache files.

First, search the PersistentLogData.db, which stores the logs from ST, for records related to a tag’s activity (i.e., creation, deletion). Analyzing these records will provide analysts with both the deviceId and the logId (Fig. 6).

Next, ST synchronizes the information of registered tags with DataLayerData.db, DataLayerData\_core.db. However, when a tag is deleted, the information in DataLayerData.db is removed, whereas data may still remain in DataLayerData\_core.db allowing for the recovery of the logId.

Finally, by utilizing the patterns of artifacts generated during the tag registration process, information such as the logId can be obtained. The steps include:

1. Searching the ST cache files using the API URL accessed for tag registration and the deviceId information.
2. Infer the tag registration time, using the server access time within the cache files.
3. Discover the logId of the deleted tag, by searching artifacts related to tag searches, model-specific registration information, and duplicate logId checks that occur adjacent to the inferred registration time.

#### 4.2.2. Location data deletion

The deletion of location data can be achieved by deleting individual tag information in ST and by removing the location data from individual tags in STF and SF.

If only the tag information is deleted (2024. 12. 03. 02:03 deviceId: 192adc88-b1cb-4cf3-b9a1-04f67ae7da45), the long-term location data is not removed and remains available even after seven days have passed (as of 2024. 12. 12), see Fig. 7.

Regarding deleting a tag’s location data on a smartphone with both STF and SF installed, the result depends on the application used. Deleting via SF removes all data, while deleting via STF does not delete the SF data (Table 9).

#### 4.2.3. Account logout

When logged out of ST and SF, the tag information in the cache, as well as the location files (location\_history), and DataLayerData.db are deleted.

However, by analyzing the remaining FME\_SELECTED\_DEVICE.xml and DataLayer-Data\_core.db files after logout, it is possible to retrieve the identification data such as the logId and the last known location of the tag.

#### 4.2.4. Service withdrawal

When withdrawing from the service, ST deletes the artifacts, similar to the logout scenario. However, by analyzing the remaining artifacts, specifically the DataLayerData\_core.db and Fme.db, it is possible to retrieve information about the tag and the last known location.

```

1 SELECT datetime(timestamp/1000, 'unixepoch') as time, title, description
2 from PersistentLogDomain where title like 'getupdateddata'
3 and (tag = 'DeviceResource' or tag = 'DataLayerDataBaseContentProviderOnCore')
4 and description like 'removed%';

```

	time	title	description
1	2024-12-04 05:35:16	getUpdatedData	removed : 1, ...

Fig. 6. Tag deletion log in PersistentLogData.db.

	time	deviceId	latitude
1	2024-12-02 13:43:35	192adc88-b1cb-4cf3-b9a1-04f67ae7da45	36.742254081818174
2	2024-12-02 18:22:19	192adc88-b1cb-4cf3-b9a1-04f67ae7da45	36.74239072579355
3	2024-12-03 01:24:02	192adc88-b1cb-4cf3-b9a1-04f67ae7da45	36.76131184
4	2024-12-03 01:26:29	192adc88-b1cb-4cf3-b9a1-04f67ae7da45	36.776522635

Fig. 7. Results of location data after deleting registered tag.

Table 9

Alterations in artifact after deletion.

Tag deletion	Deletion with STF	Deletion with SF
Long-term location data is not deleted	Location data in SF is not deleted	All location data is deleted

#### 4.2.5. Application synchronization

It is possible to install the tag applications on multiple smartphones and log in using the same Samsung account. Therefore, it is necessary to verify the impact of the location data deletion on other applications.

Once location data is deleted using STF on smartphone A, it syncs with smartphone B. However, in this case, SF location data on both A and B remains intact. On the other hand, once location data is deleted using SF on smartphone A, all other data except SF location data on B is deleted.

The changes of each artifact throughout anti-forensics scenarios are shown in Table 10.

### 5. Tool development and validation

#### 5.1. Tool development

Based on the aforementioned research findings, we developed an automated Samsung tracking tag analysis tool called S.TASER (Smart Tag Parser). S.TASER is the Python-based tool that analyzes tag applications’ artifacts and stores the results in an SQLite database. The tool then provides functionalities such as recovering deleted tag information and visualizing location data collected from individual tags based on the above log and database (Fig. 8.).

#### 5.2. Tool validation

The results of S.TASER are tested from two perspectives: (1) the identification of deleted tags and recovery of information (tag information analysis), (2) accuracy of location data extracted for each tag. The testing methods are as follows:

For tag information analysis, forensic images collected during the scenario were analyzed using S.TASER, and the results were compared with the user behaviors from the scenario (Table 11).

For tag location data, the external GPS data collected during tag movement in each scenario were compared with the location data extracted by the tool.

The experimental results indicated that the tool successfully retrieved deviceId (UUID) of all tags. However, there were instances where the identification data (e.g., logId) for deleted tags could not be recovered. Main reason behind the issue is because when the identical tag is re-registered, the cache files are updated with the information from the re-registration point, leading to potential loss of previous identification data (Fig. 9.).

To confirm the accuracy of the analyzed location data artifact, we compared the GPS values collected during tag movement with the analyzed location data.

Specifically, GPS data was collected every 2 s, and the GPS values within 5 s before and after the time of the location data analyzed from the tag were examined. These GPS and tag location values were visualized on a map, confirming that the tag’s artifact accurately reflects its

**Table 10**  
Artifact changes after deletion.

Application	Data source	Tracking tag deletion	Location data deletion	Account logout	Service withdrawal
ST/STF	DataLayerData.db	x	N/A	x	x
	DataLayerData_core.db	x	N/A	o	o
	cache	o	N/A	x	x
	PersistentLogData.db	o	N/A	Δ	Δ
	InternalSettings.db	o	N/A	o	o
	EasySetupIconNameDb.db	o	N/A	o	x
	Fme.db	Δ	x	Δ	o
	FME_SELECTED_DEVICE.xml	Δ	Δ	o	x
	location_history	o	x	x	x
	SF	app-database.db	o	Δ	o
	find-sdk	o	Δ	o	o

\* Not deleted (o), Conditionally deleted (Δ), Deleted (x).

**Tag's Information**

	deviceid	Status	RecoveredMethod	label	Model	mnid	Setupid	logid	RegistrationTime(UTC)
1	00a77e36-693d-4b34-b627-5b18bc9d3301	Recovered	log data	SmartTag 2 black re	EI-T5600	0AFD	452	Y****1056402	2024-12-04 05:44:20
2	1f8eb4c3-657b-4afd-89b0-302caf4a0bf6	Recovered	pattern	unknown		0A6W	009		2024-12-03 05:19:15
3	3b6ac4ea-ba38-4b9e-a4c0-ecfd53567b2b	live	-	SST	SOLUM SMART TAG	0A6W	009	C****666661C	2024-12-03 05:20:18
4	f486b86e-81b2-4c64-9e83-5d0291136bca	live	-	SmartTag 2 black2	EI-T5600	0AFD	452	Y****1198805	2024-12-04 06:05:28
5	ffc32683-a361-41f6-b48d-7499cf7f1118	Recovered	log data	SmartTag 2 black	EI-T5600	0AFD	452	Y****1056402	2024-12-03 05:16:02

**Tag's Location History**

	deviceid	StartTime(UTC)	EndTime(UTC)	Count	Latitude	Longitude	Accuracy	Source
1	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 05:16:10	2024-12-03 05:21:02	5	37.5536	126.9706		app-database.db
2	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 05:43:48	2024-12-03 05:44:20	2	37.4558	126.8939		app-database.db
3	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 05:48:43	2024-12-03 05:48:43	1	37.4162	126.8849		app-database.db
4	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 06:12:42	2024-12-03 06:16:04	5	36.7935	127.1048		app-database.db
5	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 07:17:18	2024-12-03 14:50:24	5	36.7425	126.9843		app-database.db
6	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-03 15:51:46	2024-12-04 02:28:17	11	36.7422	126.9843		app-database.db
7	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-04 02:42:35	2024-12-04 03:07:22	3	36.7699	126.9799		app-database.db
8	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-04 03:09:33	2024-12-04 03:46:51	9	36.7759	126.9796		app-database.db
9	ffc32683-a361-41f6-b48d-7499cf7f1118	2024-12-04 04:20:57	2024-12-04 05:14:28	6	36.7423	126.9842		app-database.db

**Fig. 8.** Output of S.TASER (Tag's information and location data).

**Table 11**  
S.TASER results (Tag information analysis).

Experiment name	Raw data Total (Deletion)	S.TASER Total (Recover)
Basic artifact structure	4 (0)	4 (0)
Tracking tag registration	3 (1)	3 (1)
Location data retrieval	N/A	N/A
Registered tracking tag deletion	4 (2)	4 (2)
Location data deletion	N/A	N/A
Account logout	4 (4)	4 (2)
Service withdrawal	1 (1)	1 (1)
Application synchronization	2 (1)	2 (1)

timestamp	uuid	infotype	info
11	2024-12-03 05:15:37	Register from db	Galaxy SmartTag2, 0AFD, 450
12	2024-12-03 05:16:01	Register from webcache	0AFD, 452, EI-T5600, ...
13	2024-12-03 05:16:03	webcache	client.smartthings.com/...
14	2024-12-03 05:16:09	webcache	client.smartthings.com/...
15	2024-12-03 05:16:10	location	("start": "2024-12-03 ...

**After re-registration**

timestamp	uuid	infotype	info
11	2024-12-03 05:15:37	Register from db	Galaxy SmartTag2, 0AFD, 450
12	2024-12-03 05:16:03	webcache	client.smartthings.com/...
13	2024-12-03 05:16:09	webcache	client.smartthings.com/...
14	2024-12-03 05:16:10	location	("start": "2024-12-03 ...

**Fig. 9.** Recover failure case (Re-registration).

real-world location data (Fig. 10.).

**6. Conclusion**

We have summarized the artifacts containing identification and location data from Samsung's tracking tag applications and examined whether the investigative objectives—such as identifying the user (e.g., confirming whether a user who used a tag is the actual suspect) and determining the time frame of criminal activity—can be achieved even

in anti-forensics scenarios. These scenarios include: 1) registered tracking tag deletion, 2) location data deletion, 3) account logout, 4) service withdrawal, and 5) application synchronization.

As a result, we found that the applications store tag identification and location data in SQLite databases, XML files, and cache. In all analysis scenarios, this data could be successfully collected.

This study analyzed relevant artifacts to examine two key points through forensic experiments: 1) whether the tag used in the crime was registered (application user identification), and 2) confirmation of

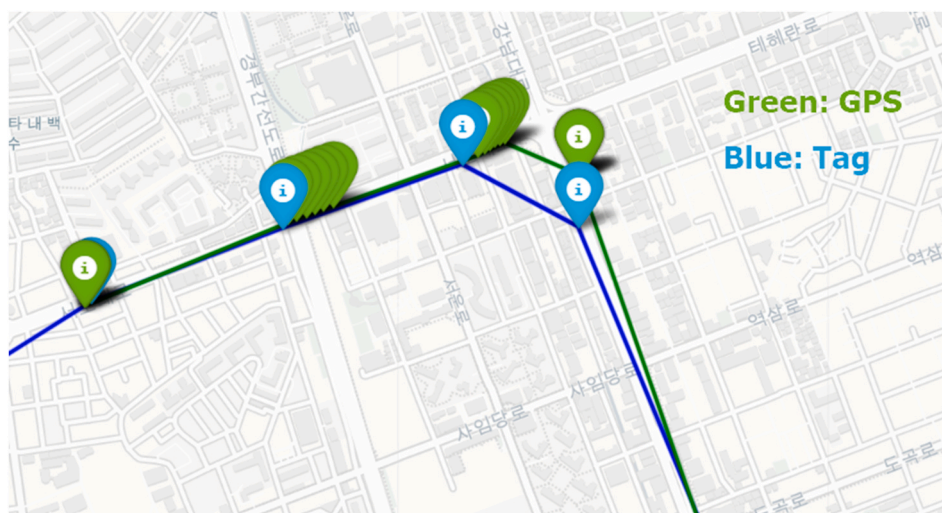


Fig. 10. Compare S.TASER's location data with GPS data.

criminal activities (location data, data collection period). In addition, the S.TASER code, experimental scenarios, and raw data are publicly available for further verification. The main significance of this research lies in being the first research to focus on Samsung tracking tag applications.

However, this study is applicable only in cases where the victim can identify the application user and a forensic analysis can be conducted on the user's smartphone. In situations where the application user cannot be identified at all, data from the tag itself needs to be obtained, which requires further research. Bongiorno (2021) have proposed methods for acquiring data from Samsung's SmartTag (EI-T5300), but these methods cannot be applied to the SmartTag2 (EI-T5600) due to a change in the chipset.

Additionally, the 7-day location data from SmartThings Find is encrypted with Android Keystore (Son et al., 2022), thus decrypting the entire location data was not possible. We could only deduce the time frame of criminal activity with available information such as initial registered tag information, daily location data collection statistics, and the last known location. For future research, we provide the encryption method of applications as well as decryption script in Frida.

Further research is needed on decryption to fully retrieve location data. Also, this study does not include web-based versions of Samsung tracking applications. Future follow-up research in this area is anticipated.

## Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00398745, Proof of evidence tampering and deployment of response technology in digital environment).

## References

- Android Help, n.d. Find unknown trackers. viewed 29 September 2024. <https://support.google.com/android/answer/13658562?hl=en&sjid=9344078109186014629-AP>.
- Alamleh, Gogarty, Ruddell, AlQahtani, 2024. Securing the invisible thread: a comprehensive analysis of BLE tracker security in Apple AirTags and Samsung SmartTags arXiv preprint arXiv:2401.13584 viewed 29 September 2024. <https://arxiv.org/abs/2401.13584>.
- Appalachian4n6, 2022. AirTags within iOS File Systems viewed 29 September 2024. <https://medium.com/@Appalachian4n6/airtags-within-ios-file-systems-279dc783b69f>.
- Binary Hick, 2022. Androids & AirTags. Oof. The Binary Hick viewed 29 September 2024. <https://thebinaryhick.blog/2022/01/08/androids-airtags-oof/>.
- Binary Hick, 2023. Android & AirTags (Part II). The Binary Hick viewed 29 September 2024. <https://thebinaryhick.blog/2023/08/13/android-airtags-part-ii/>.
- Binary Hick, 2024a. Not All Androids Who Wander Are Lost. A Look at Android's Find My Device Network. The Binary Hick viewed 29 September 2024. <https://thebinaryhick.blog/2024/08/23/not-all-androids-who-wonder-are-lost-a-look-at-android-s-find-my-device-network/>.
- Binary Hick, 2024b. Where the Wild Tags Are & Other AirTag Stories. The Binary Hick viewed 29 September 2024. <https://thebinaryhick.blog/2024/09/02/where-the-wild-d-tags-are-other-airtag-stories/>.
- Bongiorno, L., 2021. Samsung SmartTag Hack viewed 6 October 2024. <https://github.com/whidinjector/Samsung-SmartTag-Hack>.
- Caldwell, M., Andrews, J.T.A., Tanay, T., et al., 2020. AI-enabled future crime. *Crime Sci* 9 (14), 10–11. <https://doi.org/10.1186/s40163-020-00123-8> viewed 6 October 2024.
- D20 Forensics, 2022. [Air]Tag You're it! viewed 29 September 2024. <https://blog.d204n6.com/2022/04/airtag-youre-it.html>.
- Gomez, C., Oller, J., Paradells, J., 2012. Overview and evaluation of Bluetooth low Energy: an emerging low-power wireless technology. *Sensors* 12, 11734–11753.
- Gwangju District Court Decision 2021godan4187 Delivered on March 24, 2022., Mokpo Branch Court of Gwangju District Court Decision 2021gohap146 Delivered on April 21, 2022., Incheon District Court Decision 2021godan8411 Delivered on April 28, 2022., Hongseong Branch Court of Daejeon District Court Decision 2022godan844 Delivered on February 16, 2023., Seoul Eastern District Court Decision 2022godan3454 Delivered on February 22, 2023., Ulsan District Court Decision 2022gohap429 Delivered on March 24, 2023., Incheon District Court Decision 2022godan7833 Delivered on April 21, 2023., Seosan Branch Court of Daejeon District Court Decision 2022gohap115 Delivered on April 26, 2023., Gwangju District Court Decision 2023godan2262 Delivered on August 10, 2023., Goyang Branch Court of Uijeongbu District Court Decision 2023gohap27 Delivered on September 8, 2023., Yeongdeok Branch Court of Daegu District Court Decision 2023Godan211 Delivered on October 11, 2023., Daegu District Court Decision 2023gohap324 Delivered on October 13, 2023., Ansan Branch Court of Suwon District Court Decision 2023gohap291 Delivered on October 18, 2023., Ulsan District Court Decision 2023godan1688 Delivered on November 21, 2023., Sangju Branch Court of Daegu District Court Decision 2023godan319 Delivered on December 13, 2023., Pyeongtaek Branch Court of Suwon District Court Decision 2023godan2221 Delivered on January 3, 2024., Incheon District Court Decision 2023gohap644 Delivered on January 18, 2024.
- IMIR, 2023. Market Research Report viewed 6 October 2024. <https://www.intellectuallmarketinsights.com/report/airtag-market-research-report-and-current-trends/mi-005954>.
- Magnet Forensics, 2022. [Air]Tag You're it! – a Look through Location Artifacts Generated by Apple's AirTag, iOS, and macOS Devices within the FindMy Application viewed 29 September 2024. <https://www.magnetforensics.com/resources/airtag-youre-it-webinar-nov-9/>.
- Mayberry, T., Fenske, E., Brown, D., Martin, J., Fossaceca, C., Rye, E., Teplov, S., Foppe, L., 2021. Who tracks the trackers? Circumventing Apple's anti-tracking alerts in the Find my network'. In: *Proceedings of the 20th Workshop on Privacy in the Electronic Society (WPES '21)*, pp. 181–186.
- Neely, A., 2024. Pennsylvania Has Two Battling Bills that Could Make Tracking with AirTags Unlawful. *AppleInsider* viewed 6 October 2024. <https://appleinsider.com/articles/24/05/01/pennsylvania-has-two-battling-bills-that-could-make-tracking-with-airtags-unlawful>.
- Onyango, F., 2024. Samsung Sales: Market Share, Revenue & Statistics [Q2 2024]. *Tridens Technology* viewed 8 October 2024. <https://tridens technology.com/samsung-g-sales-statistics/>.

- Owen, M., 2022. Ohio House Introduces Bill to Criminalize AirTag Stalking. AppleInsider viewed 6 October 2024. <https://appleinsider.com/articles/22/05/15/ohio-house-introduces-bill-to-criminalize-airtag-stalking>.
- Pace, L., Salmon, L., Bowen, C., Baggili, I., Richard, G., 2023. Every step you take, I'll be tracking you: forensic analysis of the tile tracker application. *Forensic Sci. Int.: Digit. Invest.* 45 (Suppl. ment), 1–10.
- People v. Molina, 2024. Court of Appeal, Third District, California (Not Officially Published, Only the Westlaw Citation Is Currently Available. 2024 WL 1262906). Filed MAR 26. viewed 6 October 2024. [https://1.next.westlaw.com/Document/c2fcc9c0eba211ee9f95e0daeded7f4f/View/FullText.html?transitionType=UniqueDocItem&contextData=\(sc.Default\)&userEnteredCitation=2024wl1262906](https://1.next.westlaw.com/Document/c2fcc9c0eba211ee9f95e0daeded7f4f/View/FullText.html?transitionType=UniqueDocItem&contextData=(sc.Default)&userEnteredCitation=2024wl1262906).
- Roth, T., Freyer, F., Hollick, M., Classen, J., 2022. AirTag of the clones: shenanigans with liberated item finders. In: 2022 IEEE Security and Privacy Workshops (SPW), pp. 301–311.
- Scheckner, J., 2024. Bill Cracking Down on Tech-Assisted Stalking, AirTag Misuse Clears Second House Hurdle. Florida Politics viewed 6 October 2024. <https://floridapolitics.com/archives/652400-bill-cracking-down-on-tech-assisted-stalking-airtag-misuse-clears-second-house-hurdle/>.
- Singh, S., 2024. iPhone Users & Sales Statistics 2024 (New Data). Demandsage viewed 8 October 2024. <https://www.demandsage.com/iphone-user-statistics/>.
- Son, J., Kim, Y.W., Oh, D.B., Kim, K., 2022. Forensic analysis of instant messengers: dcrypt signal, wickr, and threema. *Forensic Sci. Int.: Digit. Invest.* (40), 301347 S.TASER. Available: <https://github.com/eininondumak/S.TASER>. viewed 15 December 2024.
- Yu, T., Henderson, J., Tiu, A., Haines, T., 2022. Privacy analysis of Samsung's crowd-sourced Bluetooth location tracking system. arXiv preprint arXiv:2210.14702 viewed 29 September 2024. <https://arxiv.org/abs/2210.14702>.
- Yu, T., Henderson, J., Tiu, A., Haines, T., 2024. Security and privacy analysis of Samsung's crowd-sourced Bluetooth location tracking system. In: Proceedings of the 33rd USENIX Security Symposium, pp. 5449–5466.