

Samsung Tracking Tag Application Forensics in Criminal Investigations

Sungkyunkwan University

Hongseok Yang, Sanghyug Han, Mindong Kim, Gibum Kim

CONTENTS

- 01.** Introduction
- 02.** Literature Review
- 03.** Experiment and Result
- 04.** Implementation and Verification
- 05.** Conclusion



1

**Samsung Tracking Tag Application Forensics
in Criminal Investigations**

Introduction

Background
Objective



Background



Tracking tags are being utilized in various fields

- From 2021, Samsung and Apple venture into the OFN market
- Pros : utilized to locate elderly individuals with dementia or lost items
- Cons : can also be misused for stalking and vishing

Lack of forensic research on Samsung tracking tag

- Apple's location-based services : not available in South Korea
- Tile : low market share
- Insufficient research on Samsung tracking tags in general

Objective

- ✓ Analyze artifact structure in Samsung tracking tag applications
- ✓ Identify artifact changes from anti-forensic actions
- ✓ Develop analysis tool

The study aims to analyze the Samsung tracking tag applications to identify artifacts and propose digital forensic techniques to substantiate user actions.

2

Samsung Tracking Tag Application Forensics
in Criminal Investigations

Literature Review

Bluetooth Low Energy (BLE) and offline Finding Network
Vulnerability Analysis
Artifact Analysis



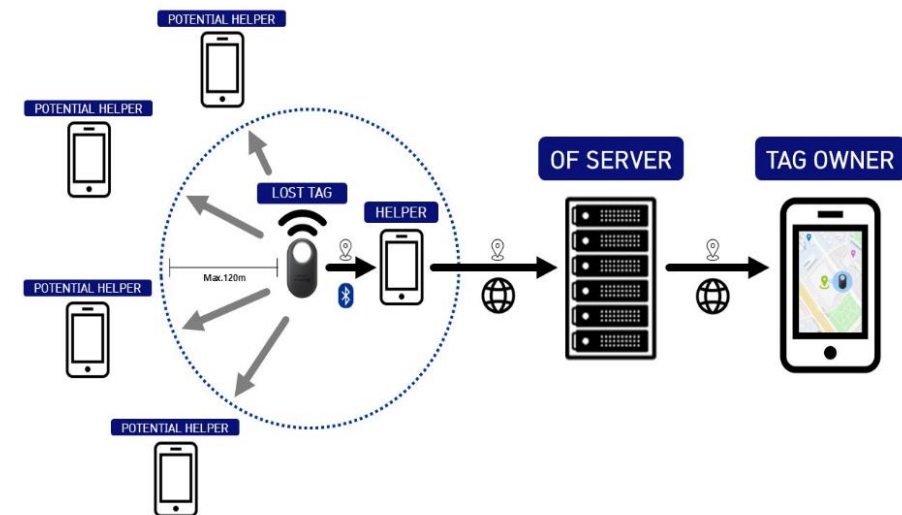
BLE (Bluetooth Low Energy) and OF network (offline Finding Network)

✓ Bluetooth Low Energy

- BLE has a shorter communication range, slower processing speed, and lower power consumption than Bluetooth Classic

✓ Offline Finding Network

- Enables the tracking of devices without an internet connection
- A lost tag Periodically emits a BLE signal to broadcast to helper devices
- Helper devices within the OF network receive the tag's signal and relay it and location to the OF network server
- Tag owners can check the uploaded location






Samsung's OF network and Applications

✓ Galaxy (SmartThings) Find Network

- Samsung's Offline Finding network : introduced in 2021
- Tags utilizing Samsung's OF network include both Samsung and third-party products

✓ Applications (run only on Galaxy devices)

- Users can manage SmartThings Find devices via the SmartThings (ST) App
- SmartThings Find (STF) : a plugin application after installing the ST App
- Samsung Find (SF) : another App for retrieving location of SmartThings Find Devices

Name		First Release Date (Number of Downloads)	Description
SmartThings (ST)		Apr 17, 2017 (1B+)	Tag management (Essential)
SmartThings Find (STF)		N/A	Tag's location retrieval (ST's plugin application)
Samsung Find (SF)		Aug 29, 2024 (10M+)	Tag's location retrieval (Optional)

Vulnerability Analysis Research

Who Tracks the Trackers? Circumventing Apple's Anti-Tracking Alerts in the Find My Network (Mayberry, T. et al., 2021)

The authors demonstrated a method to bypass AirTag's 'Item Safety Alert' feature by creating a custom device that can participate in Apple's Find My network without triggering safety alerts.

AirTag of the Clones: Shenanigans with Liberated Item Finders (Roth, T. et al., 2022)

The study modified the firmware of an AirTag and cloned the tag using a 'voltage glitching attack'. By removing the built-in speaker and altering the firmware, the authors were able to bypass stalking prevention feature.

Privacy Analysis of Samsung's Crowd-Sourced Bluetooth Location Tracking System (Yu, T. et al., 2022)

The authors analyzed Samsung's Find My Mobile to determine whether information related to the device and tag owner could be identified and location data could be tampered with. They conducted firmware and APK reverse engineering, and successfully identified information of the device and tag owner. They also tracked and manipulated the location of the tag.

Securing the Invisible Thread: A Comprehensive Analysis of BLE Tracker Security in Apple AirTags and Samsung SmartTags (Alamleh, H. et al., 2024)

The study conducted various attacks, including physical manipulation and firmware exploitation, to analyze the vulnerabilities of AirTag and SmartTag.

Artifact Analysis Research

AirTags within iOS File Systems (Appalachian4n6., 2022)

The author 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.

An Android and AirTags Study Series (Binary Hick., 2022, 2024)

The studies analyzed the AirTag detection applications 'Tracker Detect' and 'AirGuard' on Android OS, as well as Google's 'Unknown Tracker Alerts' feature. They identified AirTag information in databases, including the MAC address and geolocation data.

Every step you take, I'll be tracking you: Forensic analysis of the tile tracker application (Pace, L. et al., 2023)

The authors discovered that the Tile application stores tag-related artifacts across various operating systems, including iOS, Android, and Windows. Additionally, they developed an open-source tool called TAP (Tile Artifact Parser).

**While substantial work has been conducted on Apple and Tile products,
a significant gap remains in the forensic examination of Samsung tracking tags**

3

Samsung Tracking Tag Application Forensics
in Criminal Investigations

Experiment and Result

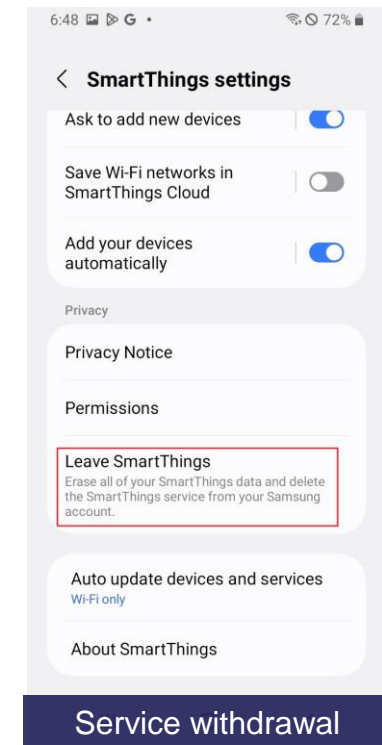
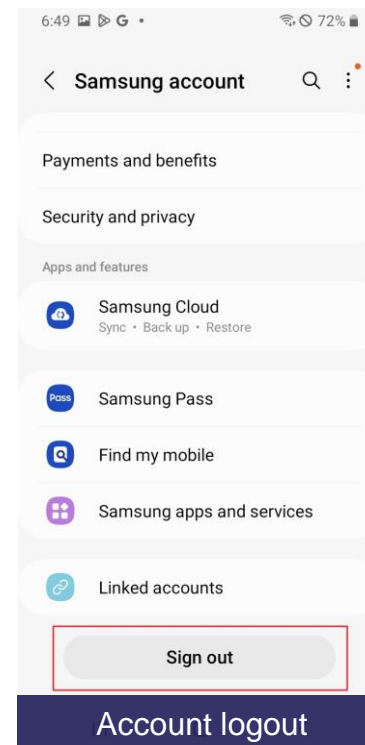
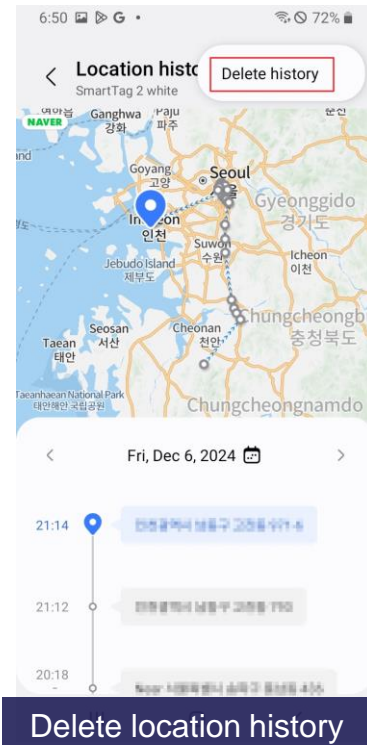
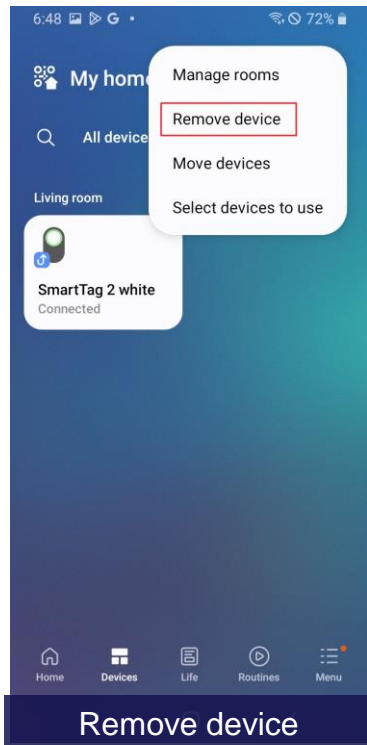
Experiment Scenarios
Environment
Methodology
Results



Experiment Scenarios

✓ Scenario Design

- Analysis of 17 criminal cases in South Korea to identify application usage patterns
- Examined the functionalities of applications available to users
- Refined the scenario through a survey of analysts with experience in tracking tag analysis



Experiment Scenarios



Scenarios Explanation

- Designed three general scenarios and five anti-forensics scenarios
- Scenarios 2 and 3 involved collecting network data between smartphones and servers
- Excluded smartphone factory reset and application deletion (data recovery impossible)

No	Experimental type	Experiment summary
1	Basic artifact structure	Tag registration, location data retrieval
2	Tracking tag registration	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

Experiment Environment

✓ Devices

- Multiple smartphones and tracking tags were tested for reliability and robustness
- Some devices were rooted to collect device and network data

✓ Applications

- Artifact structures of ST changed after the paper submission
- The artifact structures of the old version are posted on GitHub

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	

Methodology

✓ Experiment Methodology

Step 1. Perform a factory reset on smartphones

Step 2. Register and move tracking tags while also collecting separate GPS data

Step 3. Perform and record actions on registered tags

Step 4. Acquire App data before and after key actions (e.g., deletion) in each scenario

Step 5. Compare the acquired data and derive conclusions

* All experiment data is available on GitHub



Results

✓ Artifact Types and Storage Paths

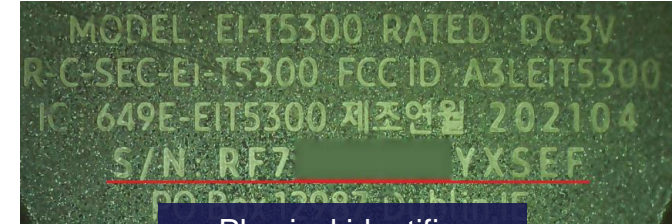
- Artifacts consist of tag identification data and location data
- Identification data includes the tag's unique information linked to the user account
- Location data is stored in string format, containing timestamp, device ID, latitude, longitude, accuracy, and address
- No significant differences were observed between the target applications
- Artifacts are stored in SQLite databases, XML files, and cache files

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

Results

✓ Identification data

- various tag-related information was used as identification data during the tag registration process
- specific details (e.g., color) can also be identified in the tag model information
- The serial numbers on the surface were not used for registration



Physical identifier

Name	Example	Description
Model name (modelName)	EI-T5300	Model Code
Manufacturer ID (mnId)	0AFD	Manufacturer Information
Model ID (setupId)	435	Detailed Model Code
User-defined name (label)	SmartTag 1 white 2	User-Defined Tag Name
Logical identifier (logId)	C****B769464	Unique Identifier (Immutable)
UUID (deviceId)	2a6a413d-e33b-48e3-a955-58afa0ecb332	Registered Tag Unique Identifier (Changes upon Each Registration, Even for the Same Tag)
Serial Number (physical identifier)	RF7*****YXSEF	Surface-Printed Unique Identifier (Immutable)

Results

✓ Artifact Generation during Tag Registration

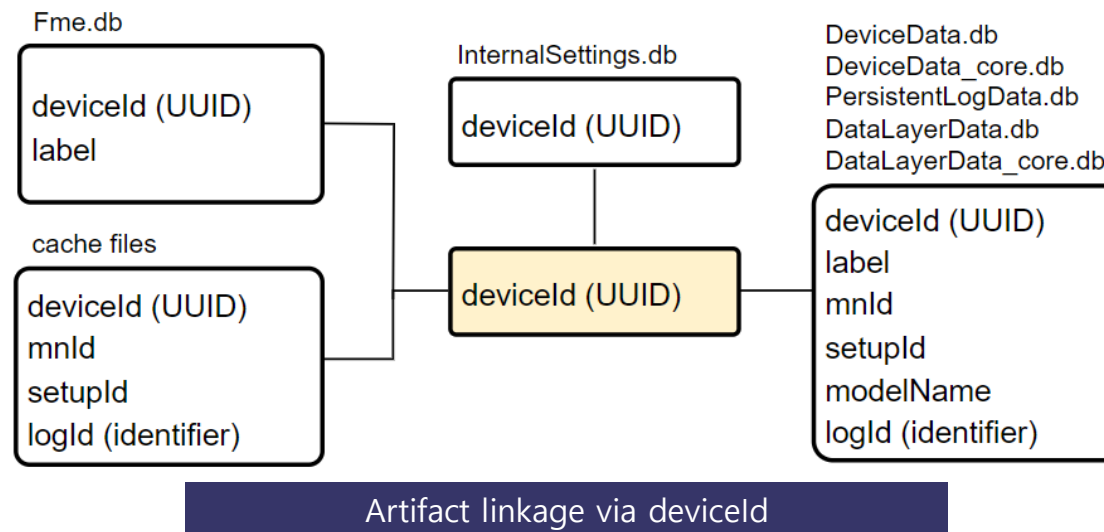
- A tracking tag is registered in the following order: ① search for nearby tag, ② obtain registration information based on model, ③ check for duplicate identifiers, ④ complete the registration, and ⑤ retrieve information for the registered tag
- In addition to database records, the registration process triggers API calls to the server, generating cache data
- Step ⑤ is executed automatically at the completion stage, without user intervention, serving as a reference point for determining the exact tag registration time
- Detailed information is available on GitHub

Order	Action	Identification data	Artifact
1	Search tag	mnId, setupId	EasySetupIconNameDb.db
2	Obtain registration information	mnId, setupId	Cache
3	Check for duplicate logId	mnId, setupId, logId, modelName	Cache
4	Registration completion	deviceId, modelName, label, mnId, setupId, logId	DataLayerData.db
5	Information retrieval	deviceId	Cache

Results

✓ Identification Data (deviceId)

- The deviceId of the tag can be found in various database files
- Certain database information that contains only the deviceId was de-duplicated
- InternalSettings.db is important as it stores the deviceIds of all registered tags after app installation. Even if a tag is deleted, the data remains



Results

✓ Location Data

- The applications differentiate between data collected over a maximum period of one week and the last known location data
- STF's long-term data and SF's last known data are encrypted
- Even without access to the encryption key, the approximate number of times location data was collected per tag can still be checked daily

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

	encDeviceid	deviceid
1	1733058838564366c	db348cf2-ddae-4eb8-be3f-62cfecdc366c
2	1733058838564b332	2a6a413d-e33b-48e3-a955-58afa0ecb332

encDeviceid = timestamp(UNIX) + deviceid last 4 bytes



No	Datetime	EncUUID	Count
1	2024-11-30 15:00:00	1733058838564b332	28

At least 28 location data were collected on 2024-12-01 (GMT+9)

Results

✔ Artifact Summary (Paper contains errors)

App	Database(Table)	deviceId	label	mnId	setupId	modelName	logId	timestamp	GeoInfo
ST/STF	DataLayerData.db (DeviceDomain)	○	○	○	○	○	○	○	-
	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	○	-	○	○	-	○	-	-
SF	com.samsung.android.pluginplatform. pluginbase.sdk.PluginSQLiteQpenHelper. [Appld].location_history * Encrypted	○	-	-	-	-	-	○	○
	app-database.db (item_history)	○	-	-	-	-	-	○	○
	find-sdk (TagGeolocation) * Encrypted	○	-	-	-	-	-	○	○

Anti-forensics scenarios

✓ Registered Tracking Tag Deletion

- The deletion of a registered tag can be confirmed by comparing deviceId information from InternalSettings.db and DataLayerData.db
- To accurately identify a real-world tag, logId is also required
- To retrieve the logId of a deleted tag, the following sources are analyzed:
1) Application logs, 2) Tag registration database (DataLayerData_core.db), 3) cache files

```

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, ...

Application log in persistentLog.db

Anti-forensics scenarios

✓ Recover Deleted Tag Data

Step 1. Identify the deleted deviceId by comparing InternalSettings.db and DataLayerData.db

Step 2. Search for information retrieval cache data based on the deviceId

Step 3. Analyze the discovered cache to determine the registration time of the deleted tag

Step 4. Search for other cache data near the registration time

Step 5. Analyze the cache files and extract logId

Order	Action	Identification data	Artifact
1	Search tag	mnId, setupId	EasySetupIconNameDb.db
2	Obtain registration information	mnId, setupId	Cache
3	Check for duplicate logId	mnId, setupId, logId, modelName	Cache
4	Registration completion	deviceId, modelName, label, mnId, setupId, logId	DataLayerData.db
5	Information retrieval	deviceId	Cache

Step 4 ~ 5



Step 2 ~ 3

Anti-forensics scenarios



Location Data Deletion

- Users can delete a tag's location data by either removing the registered tag or deleting location data for an individual tag
- Even if users delete the registered tag, STF/SF long-term location data remains stored
- This long-term data persists beyond seven days
- The impact of location data deletion varies depending on the specific application used

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

Anti-forensics scenarios

✓ Account Logout & Service Withdrawal

- Many data sources are deleted (e.g., location_history, DataLayerData.db etc.)
- However, remaining data may still allow identification and location data retrieval

✓ Application Synchronization

- Smartphones A and B are synchronized using the same account
- If location data is deleted via STF on smartphone A, it syncs with smartphone B; however, in this case, SF location data on both A and B remains intact
- If location data is deleted via SF on smartphone A, all other data except SF location data on B is deleted

	Deletion STF on smartphone A	Deletion SF on smartphone A
STF A	Deleted	Deleted
SF A	Remain	Deleted
STF B	Deleted	Deleted
SF B	Remain	Remain

Anti-forensics scenarios

✓ Summary

- In various anti-forensic scenarios, identification and location data remained undeleted

App	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	○	○
	cache	○	N/A	x	x
	PersistentLogData.db	○	N/A	Δ	Δ
	InternalSettings.db	○	N/A	○	○
	EasySetupIconNameDb.db	○	N/A	○	x
	Fme.db	Δ	x	Δ	○
	FME_SELECTED_DEVICE.xml	Δ	Δ	○	x
	location_history	○	x	x	x
SF	app-database.db	○	Δ	○	○
	find-sdk	○	Δ	○	○

○: Not deleted, x: Deleted, Δ: Conditionally deleted

4

Samsung Tracking Tag Application Forensics
in Criminal Investigations

Implementation and Verification

Implementation
Verification



Implementation

✓ S.TASER (Smart Tag Parser)

Menu

Parsing

Import

SmartTAG parSER

Samsung SmartThings App

Current Folder: None

Select SmartThings folder.

Select Folder

Result SQLite DB

Current DB File: C:\Users\A\Desktop\For DFRWS EU\program\S.TASER 1.0\resultdb\2025-03-28_22-07-57_result_db.sqlite

Change Folder

Samsung Find App (Optional)

Current Folder: None

Select Samsung Find folder.

Select Folder

Result Log

Current Log File: C:\Users\A\Desktop\For DFRWS EU\program\S.TASER 1.0\log\2025-03-28_22-07-57_result_log.txt

Change Folder

Start Parsing

Google Translate

Deploy

Implementation

✔ S.TASER (Smart Tag Parser)

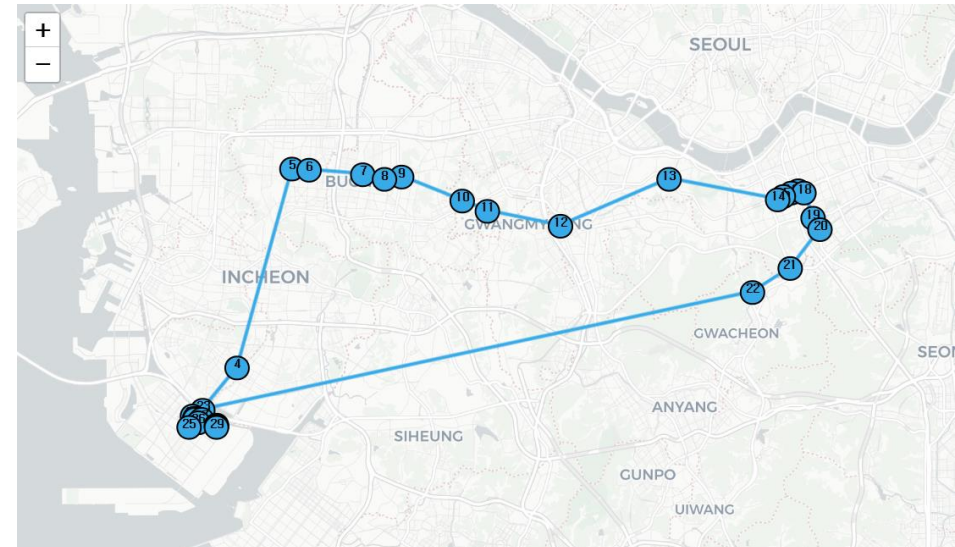
- Tool to parse application artifacts
- Provides functionalities such as recovering deleted tag information and visualizing data collected from individual tags
- * Parsing functionalities for encrypted location data have not been implemented

Tag's Information

	UUID	Status	RecoveredMethod	Name	Model	MNid	SetupId	Identifier
1	00a77e36-693d-4b34-b627-5b18bc9d3301	recovered	log data	SmartTag 2 black re	EI-T5600	0AFD	452	Y48081056402
2	1f8eb4c3-657b-4afd-89b0-302ca4a0bf6	recovered	pattern	unknown		0AGW	009	
3	3b6ac4ea-ba38-4b9e-a4c0-ecfd53567b2b	live	-	SST	SOLUM SMART TAG	0AGW	009	C40D6666661C
4	f486b86e-81b2-4c64-9e83-5d0291136bca	live	-	SmartTag 2 black2	EI-T5600	0AFD	452	Y48081198805
✔ 5	ffc32683-a361-41f6-b48d-7499cf7f1118	recovered	log data	SmartTag 2 black	EI-T5600	0AFD	452	Y48081056402

Tag's Location History

	UUID	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



Verification

✓ Validation of Identification Data Result

- Results were validated from the perspectives of identification and location data
- A deleted tag was considered successfully recovered when its logId was identified
- If the same tag was registered multiple times, only the logId information of the last registered and deleted tag was retrievable, while the logId of previous registrations remained inaccessible without application logs

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	ffc32683-a361-41f6-...	webcache	client.smarththings.com/...
14	2024-12-03 05:16:09	ffc32683-a361-41f6-...	webcache	client.smarththings.com/...
15	2024-12-03 05:16:10	ffc32683-a361-41f6-...	location	{"start": "2024-12-03 ...

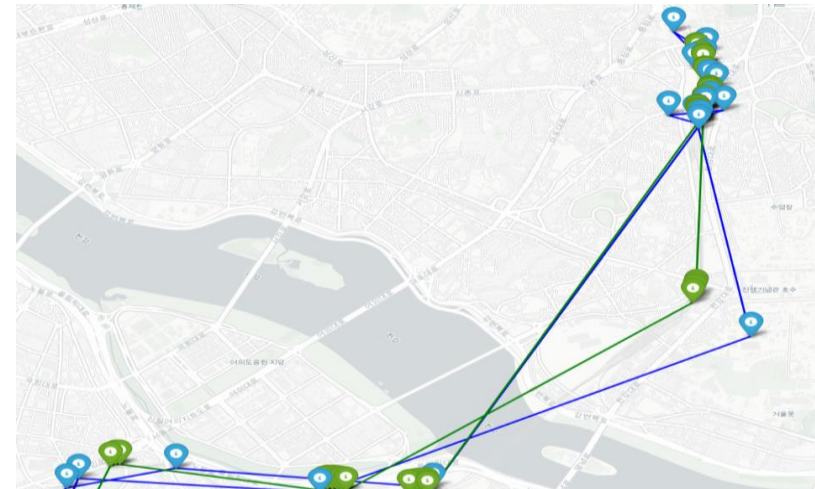
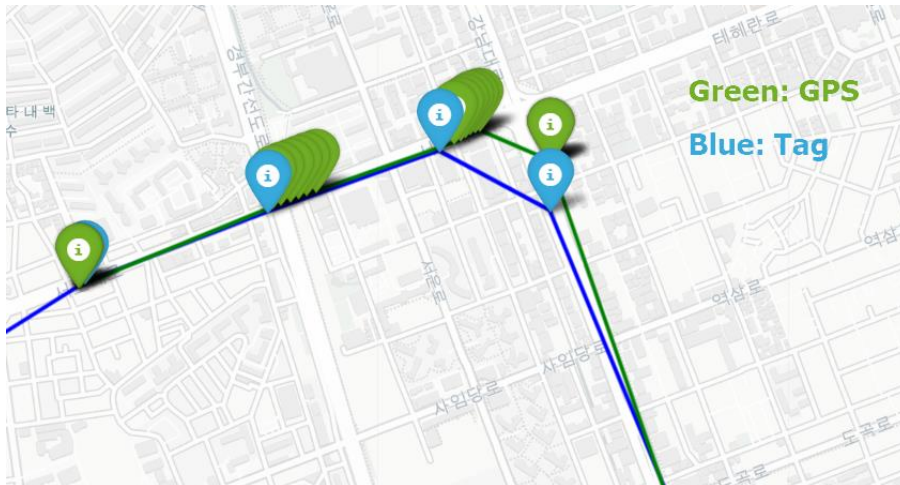
After re-registration

	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	ffc32683-a361-41f6-...	webcache	client.smarththings.com/...
13	2024-12-03 05:16:09	ffc32683-a361-41f6-...	webcache	client.smarththings.com/...
14	2024-12-03 05:16:10	ffc32683-a361-41f6-...	location	{"start": "2024-12-03 ...

Verification

✓ Validation of Location Data Result

- When a tag moves, GPS data is collected every 2 seconds using an additional smartphone application
- To verify the accuracy of the analyzed location data, it is compared with GPS data recorded 5 seconds before and after the application's location collection timestamp
- The tool's analysis results accurately reflect the tag's actual movement path



5

Samsung Tracking Tag Application Forensics
in Criminal Investigations

Conclusion



Conclusion

- We have summarized the artifacts containing identification and location data from Samsung's tracking tag application
- The study analyzed relevant artifacts to examine two key points through forensic experiments:
 - 1) whether a tag used in the crime was registered (application user identification),
 - 2) whether location information of the user was collected (timestamp, detailed data)
- Even after anti-forensic actions, a deleted tag's information can be identified using the remaining data
- Developed an automated Samsung tracking tag analysis tool called S.TASER (Smart Tag Parser)
- Some location data is encrypted, and S.TASER does not support decryption functionality

Q&A

Email : eininondumak@gmail.com

