Forensic Science International: Digital Investigation 45 (2023) 301559

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2023 USA - Proceedings of the Twenty Third Annual DFRWS Conference

Every step you take, I'll be tracking you: Forensic analysis of the tile tracker application

Lauren R. Pace [a, b, 1, *], LaSean A. Salmon [a, b, 1], Christopher J. Bowen [a, b, 1], Ibrahim Baggili [a, b], Golden G. Richard III [a, b]

[a] School of Electrical Engineering & Computer Science, Louisiana State University, USA
[b] Center for Computation and Technology, Louisiana State University, USA

ARTICLE INFO

Article history:

Keywords:
Bluetooth tracker
IoT
Mobile forensics
Memory forensics
Artifacts
Android forensics
iOS forensics

ABSTRACT

The rise in popularity of personal Bluetooth trackers has incited a need for forensic analysis tools that aid law enforcement in artifact recovery. With 40 million Tile devices reportedly sold at the time of writing, Tile trackers are one of the most popular personal Bluetooth trackers. This growth has not been without consequence, as reports of Bluetooth trackers being used for malicious activities have also escalated. Our work presents a forensic analysis of the Tile ecosystem and the Tile application on iOS, Android, and Windows. This analysis revealed valuable forensic artifacts that contained a diverse set of sensitive user data, including SQLite databases, XML files, cache files, and event logs. This data included information such as geolocation coordinates from the previous 30 days. As part of our analysis process, we developed an open-source tool capable of parsing these forensic artifacts from the Tile application: Tile Artifact Parser (TAP). TAP parses SQLite databases and virtual memory files, mapping geolocation coordinates and linking them according to timestamps. The ability to quickly and efficiently parse and map these location points provides valuable information in an investigation. TAP also aids investigators by detecting potentially spoofed data and flagging it. The robustness of TAP was tested to ensure its effectiveness and behavior in cases of incomplete or missing data.

1. Introduction

Bluetooth trackers are personal Internet of Things (IoT) devices that allow users to keep track of belongings ranging from backpacks to keys. These devices work by transmitting data to their companion devices using the Bluetooth Low Energy (BLE) protocol (Briggs and Geeng, 2022). One of the most popular Bluetooth tracker manufacturers is Tile. Tile trackers use a BLE connection established with the companion device hosting the Tile application. Once attached to a personal belonging, location data corresponding to the object is updated using the Global Positioning System (GPS) services of the companion device. In this sense, accurate location data for Tile devices is restricted by their ability to connect with a central device. Tile overcomes this restriction by establishing a network of user devices from which location data can be obtained, making it easier to find possessions if they are lost (Tile, n.d.). This Tile device communication ecosystem is detailed in Fig. 1.

The personal Bluetooth tracker industry has grown rapidly. Acumen Research and Co. project the global market value of IoT trackers to reach $1.65 billion by 2030, with Tile having sold more than 40 million devices to date (Acumen, 2022; Perez, 2021). As Bluetooth trackers integrate into the lives of a large pool of users, privacy concerns arise surrounding the substantial amounts of pervasive data they will accumulate. There have recently been lawsuits filed regarding Bluetooth trackers being used for stalking (Archie, 2022).

Life360, a technology company specializing in location-based services, announced in November 2021 that they would be acquiring Tile for $205 million. The statement claims that by combining Life360's network of 33 million smartphones with Tile their finding network will expand roughly ten times (Life360, 2021). To the best of our knowledge, the impact that Life360's acquisition has had on the security of Tile has yet to be researched.

* Corresponding author. School of Electrical Engineering & Computer Science, Louisiana State University, USA.

E-mail addresses: lpace9@lsu.edu (L.R. Pace), lsalmo1@lsu.edu (L.A. Salmon), cbowe13@lsu.edu (C.J. Bowen), ibaggili@lsu.edu (I. Baggili), golden@cct.lsu.edu (G.G. Richard).

[1] Authors contributed equally.

https://doi.org/10.1016/j.fsidi.2023.301559

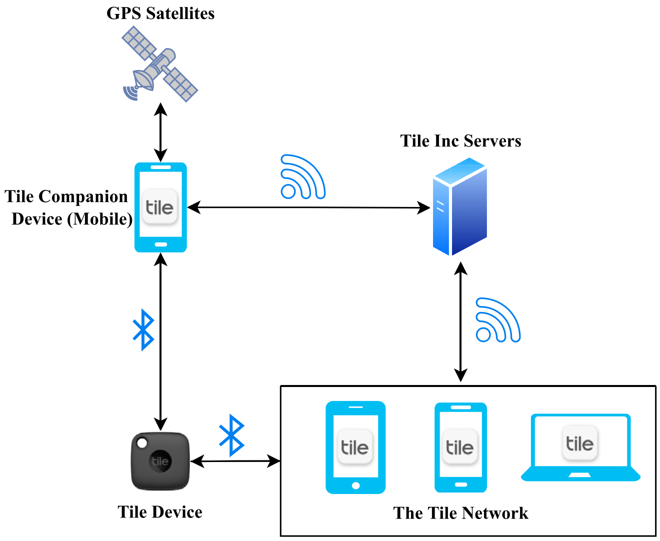


Fig. 1. Tile IoT ecosystem.

As Tile devices pervade the market, concerns regarding their use in criminal activities have risen. While there has yet to be any federal legislation outlining the protection of geolocation data for private citizens in the United States, many states have defined their own laws to criminalize unwanted tracking (Geolocation Privacy Legislation, 2021). Multiple incidents report unknown Tile devices being stashed in cars to enable stalkers to keep tabs on their victim's location (Welsch, 2022; Willey, 2018). Tile recently confirmed the serious nature and prevalence of these incidents with the release of their new anti-stalking feature, Scan and Secure, in March of 2022 (Tile, 2022). This feature enables users to detect any rogue Tile trackers planted on their person or belongings.

While the development of rogue device detection is a beneficial step in ensuring user safety, it is not enough to eliminate other malicious activity utilizing Tile devices. While Tile devices are used by valid users to track valuables, such as luggage, during transport, malicious users may use these devices to track criminal evidence or illegal goods (Tile, 2015). In these cases, the ability to recover forensically sound digital evidence from Tile can be invaluable to investigators.

With our work, we aim to reveal how data produced by Tile devices and their companion applications can be used to combat these criminal efforts. Our contributions are as follows:

- We present a peer-reviewed analysis of the Tile application across all companion device types.
- We created an open-source Python 3 tool called Tile Artifact Parser (TAP)[2] that can parse forensic artifacts from the Tile application on iOS, Android, and Windows.
- We incorporated memory forensics as a tool to identify Tile artifacts stored in volatile memory.
- We provided an analysis of the potential for anti-forensics techniques to be used on Tile application data and presented mitigation strategies.

This paper provides an investigation into the artifacts that can be discovered from the Tile application. This is followed by an analysis of the artifacts and a discussion of our tool created to analyze these artifacts.

2. Related work

In recent years, there has been a rise in Bluetooth personal trackers (Acumen, 2022). This section follows the forensic analysis completed with similar applications to Tile, along with prior research on application analysis.

2.1. Application analysis

Application forensics research has advanced rapidly over the years.

Previous research focuses on the advancement of forensics techniques and their relevant applications (Chernyshev et al., 2017).

Other works focused on obtaining digital evidence or decrypting network traffic originating from social media applications (Walnycky et al., 2015; Karpisek et al., 2015). Similar to Tile, social media and messaging applications involve the exchange of sensitive data over the internet, and valuable information originating from these applications is commonly recovered from suspects' devices during forensic investigations.

As an example, forensic analysis of the Discord application for Windows, a popular messaging platform, revealed the ability to retrieve sensitive data and recover deleted data from the local application files (Moffitt et al., 2021). The artifacts identified through this forensic analysis can be used during a forensic investigation to collect information about a suspect and their activity.

Applications that collect users' locations are common, but they come with safety concerns.

Knox et al. (2020) researched Happn, a dating application that tracks users' locations and matches with other users whose locations cross. The iTunes backups and the physical image artifacts they obtained also showed the user's location (Knox et al., 2020). Life360, a family tracking application, was researched by (Bays and Karabiyik, 2019). They investigated what artifacts could be found on iPhone Operating System (iOS) and Android devices. Initial investigations revealed little from either device, with only user GPS coordinates found on the iOS device. Rooting an Android device allowed for GPS location data to be found (Bays and Karabiyik, 2019). Rooting and jailbreaking devices is not ideal, and the goal is to avoid this step in our research. An analysis of 13 health and fitness applications on Android was conducted by (Hassenfeldt et al., 2019). Utilizing Android Debug Bridge and XRY, they recovered personal information and geolocation data consisting of latitude, longitude, and altitude (Hassenfeldt et al., 2019).

2.2. Memory forensics

Efforts to recover artifacts from Random Access Memory (RAM) have gained traction in digital forensics over the last decade. Memory images allow insight into the actively running processes on a device. Case and Richard III (2017) provided a critical analysis of the current state and goals of memory forensics.

There are now many open-source tools that allow for extraction of forensic artifacts from volatile memory. Researchers and investigators use memory forensics techniques and tools to identify valuable forensic artifacts (Case and Richard, 2016; Casey et al., 2019; Satrya and Kurniawan, 2020).

We now discuss previous research involving two prominent BLE trackers: AirTag, a major competitor of Tile, and Tile, the focus of this paper.

2.3. Bluetooth trackers [AirTag and Tile]

AirTag is Apple's personal tracking solution. Similar to the Tile Network, Apple's Find My network is an expansive web of Apple

[2] https://github.com/BiTLab-BaggiliTruthLab/TAP.

devices that allows AirTags to communicate over both the internet and BLE advertisements. Roth et al. (2022) identified a voltage glitching attack that could be performed on the Nordic nRF 52832 chip used for BLE in AirTags, allowing for firmware extraction. Modification of this firmware led to the discovery of hidden device functionality, AirTag identity spoofing, and BLE communication hijacking. Their work proved the impact that firmware reverse engineering could have on the safety and privacy of BLE tracker users.

Gazeau and Liu (2020) addressed the rising controversy surrounding Tile, referencing an increase in crimes that relied on maliciously planted Tile devices. With the mitigation of criminal activity as the goal of their research, they focused on recovering geolocation artifacts related to the use of Tile devices. Their scope was limited to the forensic analysis of application data on an iPhone X. Log files were extracted from an unencrypted backup of the companion device. They were able to retrieve coordinates and corresponding timestamps from these files, which they plotted on an interactive map using Python 3 scripts.

On his blog in 2019, Vance (2019) analyzed the Tile Android application, looking for location traces contained in the application data. He found Extensible Markup Language (XML) files and SQLite databases linking Media Access Control (MAC) addresses and Unique User Identification (UUID)s to crucial data such as location, timestamps, and device names. This data included many Tile devices belonging to other users. Vance went on to address the Tile iOS application with the same motivations. In this work, he created an artifact plugin for the open-source iOS Logs, Events, And Plists Parser (iLEAPP) that identifies forensic artifacts in iOS logs and artifacts for parsing (Vance, 2020).

Weller et al. (2020) investigated the security of Tile and other personal trackers that utilize BLE. They found that the firmware was stored on the server in plaintext. If connected to the vulnerable Message Queuing Telemetry Transport (MQTT), users' phones could be pinged with only the easily leaked Tile UUID. Weller et al. (2020) reported these vulnerabilities to Tile, and Tile responded and reportedly applied fixes.

Currently, there is little peer-reviewed work completed on Tile. Our work provides a comparison between the forensic acquisition process across the Tile Android, iOS, and desktop applications.

To the best of our knowledge, none of the previous investigations were conducted on the latest version of Tile since the Life360 acquisition. Previous literature also does not address the implications of anti-forensic techniques on Tile application data. Additionally, we created a more comprehensive parsing tool that pulls forensic artifacts from both databases and memory dumps.

3. Methodology

Our work focused on forensically reconstructing and extracting user data from Tile companion devices. To address multiple types of devices investigators may discover, we utilized an iPhone SE, an Android Huawei tablet, an Android Samsung Phone, and a Windows Virtual Machine (VM). The phases of our study consisted of Setup and Scenario Creation, Data Acquisition, Data Analysis and Results, and Tool Development: TAP (Tile Artifact Parser). A detailed apparatus of devices and software is shown in Table 1.

3.1. Setup and Scenario Creation

To begin the experimental setup, the iPhone SE, Huawei Android tablet, and Samsung Galaxy S10+ were factory reset, and a fresh Windows VM was created. Then, the Tile application was installed on all companion devices. The same account was logged into all devices, except for the Galaxy S10+, prior to data generation. Various Tile trackers were paired to the accounts, and scenarios were created to mimic different real-world events. Location spoofing applications were installed for Windows and Android to test the efficacy of anti-forensics techniques.

Two forensic workstations, an iMac and a Windows desktop computer, were set up with the software listed in Table 1. The data generation phase of our experiment was planned by designing real-world investigation scenarios that exemplify the relevance of Tile data acquisition. These scenarios included the following:

- A mobile smart device is either discovered at the scene of a crime or suspected to have been used in a crime.
- A suspect is identified, and their home is searched. The desktop computer in their home is left on, and the Tile desktop application is running.
- A malicious actor utilizes a location spoofing application to modify their location.

Data generation consisted of walking a predetermined path with both the Tile devices and the companion devices, see Fig. 7.

Table 1
Apparatus table depicting the hardware and software utilized throughout the experiment.

Hardware/Software	Use	Company	Software/Model Version
MediaPad M5 Tablet	Tile Companion Device	Huawei Technologies Co., Ltd	Android 8.0.0
iPhone SE	Tile Companion Device	Apple Inc	iOS 15.5
Galaxy s10+	Tile Companion Device	Samsung	Android 11.0.0
Windows Virtual Machine	Tile Companion Device	Microsoft Corporation	Windows 10.0.19042
VMWare Workstation Pro	Host VMs	VMWare	16.0.0 build-16894299
Tile (Android)	Tile Data Generation and Acquisition	Tile Inc.	2.75.0
Tile (iOS)	Tile Data Generation and Acquisition	Tile Inc.	2.105.0
Tile (Windows Desktop)	Tile Data Generation and Acquisition	Tile Inc.	3.3.27.0
Tile Mate	Tested Tile Device	Tile Inc	T1401S
Tile Sticker	Tested Tile Device	Tile Inc	T1501S
Tile Slim	Tested Tile Device	Tile Inc	T1601S
iPhone Backup Extractor	Extract iPhone Backup Data	Reincubate Ltd.	7.3.5.0
UFED	Phyical Acquisition for Android	Cellebrite	7.53.0.24
iMac	Data Acquisition/Analysis	Apple	Monterey 12.0.1
Alienware Aurora R12	Data Acquisition/Analysis, VM Host	Dell	Windows 10.0.19044
DB Browser for SQLite	View SQLite Databases	DigitalOcean, LLC	3.12.2
Fake GPS Location	Location Spoofing	Lexa	2.1.2
AnyGo	Location Spoofing	Shenzhen LuckyDog Tech. Co., Ltd.	5.9.2

Devices were deposited and retrieved from locations when applicable to an associated scenario. The spoofing applications were used to accomplish falsified location injection. The next phase was to perform data acquisition.

3.2. *Data acquisition*

Artifacts from the Android device were captured using a Cellebrite Universal Forensics Extraction Device (UFED). Because we used a Huawei tablet, the UFED Decrypting Kirin Bootloader had to be used to obtain an exhaustive backup. For retrieving iPhone artifacts, the iPhone was backed up to the iMac workstation. Files were extracted from the backup using iPhone Backup Extractor. For the Tile desktop application, we used VMware to create a memory image of the VM and then inspected the memory for forensic artifacts.

4. Data Analysis and Results

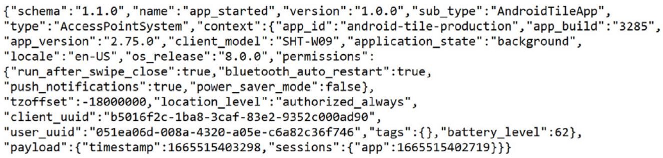
Different data analysis techniques were used for each companion device. The database format differed between Android and iOS. Both devices contained detailed log files showing a history of Bluetooth payloads. These payloads include MAC addresses, user UUIDs, battery level of the device, and client model. The iOS payloads provide more detailed information, including location coordinates, location accuracy, and Tile UUIDs. These logs are depicted in Fig. 2.

4.1. *iOS backup*

Numerous SQLite database files were found in the iTunes backup. A database titled `com.thetileapp.tile-TileNetworkDB.sqlite` held particular significance. One table in the database, `ZTILENTITY_PLACEMARK`, stored latitude, longitude, timestamp, postal code, city, and street address information. Part of this table is depicted in Fig. 3. All timestamps were stored in the Apple Cocoa Core Data format. This data combined effectively shows the general path of a person or their belongings. Table 2 shows the artifacts found.

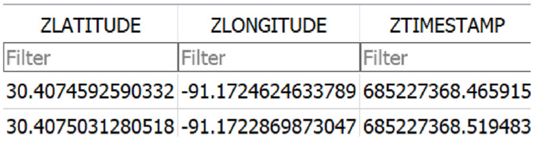
4.2. *Android backup*

The Tile data contained in the full Huawei Android tablet backup included numerous cache, XML, and SQLite database files. Some XML files detailed the email linked to the account and a variety of UUIDs, tokens, and cookies. Within the Tile data, we also discovered links that opened HyperText Markup Language (HTML) webpages picturing mapped locations. These images correlated to places traveled during the Setup and Scenario Creation phase of the experiment.



```
{"schema":"1.1.0","name":"app_started","version":"1.0.0","sub_type":"AndroidTileApp",
"type":"AccessPointSystem","context":{"app_id":"android-tile-production","app_build":"3285",
"app_version":"2.75.0","client_model":"SHT-W09","application_state":"background",
"locale":"en-US","os_release":"8.0.0","permissions":
{"run_after_swipe_close":true,"bluetooth_auto_restart":true,
"push_notifications":true,"power_saver_mode":false},
"tzoffset":-18000000,"location_level":"authorized_always",
"client_uuid":"b5016f2c-1ba8-3caf-83e2-9352c000ad90",
"user_uuid":"051ea06d-008a-4320-a05e-c6a82c36f746","tags":{},"battery_level":62},
"payload":{"timestamp":1665515403298,"sessions":{"app":1665515402719}}}
```

Fig. 2. Tile application log entry from an Android device.



ZLATITUDE	ZLONGITUDE	ZTIMESTAMP
Filter	Filter	Filter
30.4074592590332	-91.1724624633789	685227368.465915
30.4075031280518	-91.1722869873047	685227368.519483

Fig. 3. `ZTILENTITY_PLACEMARK` table entries containing geolocation coordinates.

Table 2
iOS Application Artifacts.

Database Name	**Tables Found**
com.thetileapp.tile-TileNetworkDB.sqlite	ZTILENTITY_PLACEMARK ZTILENTITY_TILEFIRMWARE ZTILENTITY_USER
Table Name	**Information Found**
ZTILENTITY_PLACEMARK	location accuracy, latitude, longitude, timestamp, country, locality, sublocality, building/street name
ZTILENTITY_TILEFIRMWARE	Tile firmware image (binary)
ZTILENTITY_USER	email addresses of shared accounts
ZTILENTITY_NODE	activated tile devices, ID attached to each device, last modified timestamp
Files	
Events_b.log Events_c.log	client, user, and Tile UUIDs, location accuracy, latitude, longitude, timestamps, battery level, client model

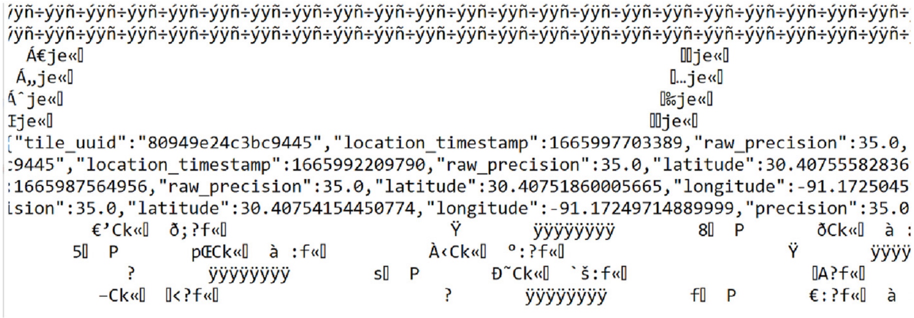
The data acquired from the SQLite databases was more limited in statistical significance compared to our iOS findings. One database table, the `notification_content_data_table`, contained a few entries that included geolocation coordinates. A table titled `regions` was also found in the Tile data. This table appeared to be similar to the `ZTILENTITY_PLACEMARK` table found in iOS, but it was not populated. A summary of the Android artifacts we discovered can be found in Table 3.

4.3. *Desktop application*

Forensic investigation of the Tile desktop application was conducted by analyzing system data files. Within the WindowsApps folder were three Tile data directories that contained Dynamic Link Libraries (DLL) and other Tile application data. We also captured a memory image of the VM while the Tile process was running. Virtual memory (VMEM) files retrieved from the Windows VM memory dumps revealed geolocation coordinates, timestamps in Epoch timestamp format, Tile UUIDs, and Tile device names stored in plaintext (depicted in Fig. 4). The ability to efficiently retrieve

Table 3
Android application artifacts.

Database Name	**Tables Found**
tileAndroidDb.db	notification_v2_table
Table Name	**Information Found**
notification_v2_table	notification data
notification_content_data_table	location accuracy, longitude, latitude
Files and Cache	
fw_files	Tile firmware image (binary)
TilePrefs.xml	client UUID, cookies, linked accounts tile-image-cache Google API static location images
events_b.log events_c.log	client and user UUIDs, timestamps, battery level, client model



```
ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷
ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷ýÿñ÷
  Á€je«▯                                                                  ▯▯je«▯
 Á„je«▯                                                                  ▯…je«▯
Áˆje«▯                                                                  ▯‰je«▯
Ije«▯                                                                  ▯▯je«▯
["tile_uuid":"80949e24c3bc9445","location_timestamp":1665997703389,"raw_precision":35.0,
:9445","location_timestamp":1665992209790,"raw_precision":35.0,"latitude":30.40755582836
:1665987564956,"raw_precision":35.0,"latitude":30.40751860005665,"longitude":-91.1725045
ision":35.0,"latitude":30.40754154450774,"longitude":-91.17249714889999,"precision":35.0
         €’Ck«▯   ð;?f«▯               Ÿ          ÿÿÿÿÿÿÿÿ        8▯  P        ðCk«▯  à :
      5▯  P        pŒCk«▯  à :f«▯          À‹Ck«▯  º:?f«▯                  Ÿ          ÿÿÿÿ
           ?         ÿÿÿÿÿÿÿÿ        s▯  P        Đ˜Ck«▯  `š:f«▯                ▯A?f«▯
        –Ck«▯  ▯<?f«▯                ?         ÿÿÿÿÿÿÿÿ        f▯  P        €:?f«▯  à
```

Fig. 4. Geolocation coordinates found in the Windows VM Memory Dump.

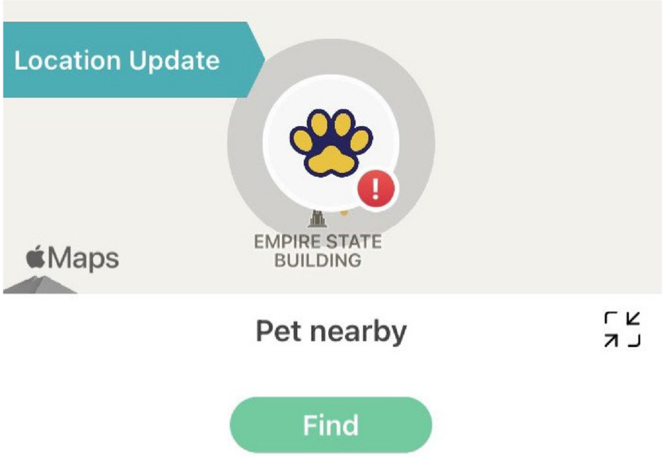


Fig. 5. Appearance of a spoofed location in the Tile iOS application.

these coordinates from a running machine provides valuable insight into a person's, or their belongings', whereabouts without recovery of a mobile device. We describe the tool created to accomplish this in Tool Development: TAP (Tile Artifact Parser).

4.4. Firmware

Leakage of Tile firmware binaries occurred through three links to an Amazon Web Service (AWS) bucket where the files were stored server-side. These links were stored inside a database found locally in the Tile iOS application data and in the Windows virtual memory (VMEM) file. No authentication was required to download the firmware through these links. The firmware binaries were also found in the iPhone and Android backups.

4.5. Anti-forensics

Spoofing applications are used to falsify the location of a user's device. To test how other users' data is affected by a spoofed device in the Tile Network, we conducted the following experiment:

- The iPhone and Samsung phones were set up with two different Tile accounts.
- A Tile Sticker paired to the iPhone was placed out of BLE range (approximately 100 m) of test companion devices and marked as lost.
- The location of the Samsung phone was modified using the GPS spoofing application.
- We walked near the “lost” Tile Sticker with the GPS spoofed Samsung phone.

When within range, the location received by the iPhone through the Tile Network matched the modified location of the spoofed device.

We also tested how effective location spoofing is on the iPhone. Applications that allow users to spoof their location are barred from the Apple App Store but can be downloaded to a computer. Therefore, spoofing an iPhone's location can only occur while the phone is physically connected to a computer with a spoofing application downloaded. While connected to a Windows location spoofing application, the Tile user interface (UI) displayed the falsified data. See Fig. 5.

5. Tool development: TAP (Tile Artifact Parser)

Acquiring data in a timely and reliable fashion is crucial to the success of an investigation. For this reason, the primary purpose of our tool development was to accelerate the process of locating and parsing out any forensically relevant Tile data. Tile Artifact Parser (TAP) is a command line forensics tool developed in Python 3. The tool can parse artifacts from memory images from Windows and SQLite database files that are collected from iOS.

5.1. Tool usage

TAP provides two main functions:

- Parsing data from Windows computer memory images.
- Parsing data from SQLite databases recovered from mobile device logical backups.

TAP is used with the following commands:

```
python3 TAP.py [−h] [−o OUTPUT] [−s STARTDATE]
[−e ENDDATE] [−f FALSE DATA] input
```

The input file must be a VMEM file or an SQLite file. The `-h` flag is for printing a menu to show available commands. The `-o` flag allows specification of where to output the data. The `-f` flag toggles the identification of data with a high chance of being falsified. There is also the ability to select a date range to map coordinates within with the `-s` and `-e` flags.

Algorithm 1. Mapping Coordinates

```
Algorithm 1 Mapping Coordinates
Requirements: Database files from image or VMEM file
Input: (optional) OutputPath, (optional) dateRange, (optional) falseData
Output: Map of Location Data and Text-based Report

for $file \in directory$ do
    if $vmemfile$ then
        $backups.add(vmemfile)$
    end if
    if $SQLitefile$ then
        $backups.add(SQLitefile)$
    end if
end for
$locationData \leftarrow getLocationData(backups)$
$sortedLocations \leftarrow sortLocation(Timestamp)$
if $dateRange$ then    ▷ Remove coordinates not within selected range
    $sortedLocations.remove(!dateRange)$
end if
for $location \in locationData$ do
    $Map.plot(location)$
end for
for $location \in map$ do
    $Location.drawLine$    ▷ Connect coordinates to show path based on timestamp
end for
if $falseData$ then    ▷ Spoofing detection
    $MaxMPH = 650$
    for $l \in locationData$ do
        if $calculateSpeed(l, l+1) > MaxMPH$ then
            $spoofedPoints.add(l+1)$
        end if
    end for
end if
if $OutputPath$ then $Output\ report$
else
    $Output\ report$    ▷
end if
procedure CALCULATESPEED($coord1, coord2$)
    $timegap \leftarrow coord2.time - coord1.time$
    $distancegap \leftarrow coord2.location - coord1.location$
    $speed \leftarrow distancegap/(timegap/1000/60/60)$
end procedure
```

5.2. TAP parsing function

TAP searches through logical extraction images from a companion device or a VMEM file to (1) locate forensic artifacts from the Tile companion devices and (2) display the data in an easily readable format.

The tool searches through a directory and locates any memory dumps or SQL files. After finding all relevant files, the tool then parses for geolocation data.

For parsing a memory dump, TAP linearly searches through a VMEM file for a signature to identify geolocation data originating from the Tile Windows application. To parse Tile iOS database files, TAP searches for the `ZTILENTITY_PLACEMARK` table and parses out the geolocation data.

The data found is exported to a text file report. The report details the input parameters supplied and information about the files that were identified and parsed by the tool. Also included is an overall count and list of all data points parsed by TAP.

TAP outputs the geolocation coordinates from either the memory dump or the database file to an HTML file. The HTML file contains an interactive map and generates paths between coordinates based on the correlated timestamps. The high-level algorithm our tool implements is detailed in Algorithm 1, and the example output is shown in Fig. 7a and b.

5.3. TAP spoofing detection

We have shown that location data can be easily spoofed. Thus, methods for detecting falsified data should be implemented.

In TAP, we chose to categorize data with a potential risk of falsification based on a reasonable measure of speed. TAP identifies the distance between two consecutive data points and uses timestamps to calculate speed. We considered the top speed of a commercial plane (~650 miles per hour) to be the upper limit for reasonable travel. Data points categorized as being a spoofing risk are output and labeled at the bottom of the TAP output report to alert investigators. The potentially spoofed locations are also marked on the map with a black location marker, as seen in Fig. 6. Any coordinates directly associated with an unreasonably high-speed change in location are indicated on the mapped output with a black marker. Details of the spoofing detection procedure are presented in Algorithm 1.

5.4. Tool evaluation

To test the validity of the parsing method TAP uses, we completed an evaluation of the accuracy and exhaustiveness of the output. We aimed to establish whether the data extracted by TAP was a statistically significant match to the original data input. To do this, we performed two sets of correlation tests addressing each of the two data formats TAP parses: SQLite database files and VMEM files. To evaluate the robustness of our tool, we generated a diverse set of data files, including files that were empty, contained partial data, or contained a reasonably large number of data points. In the subsections that follow, we detail the data generation process and the exact test procedure carried out to address each data format.

5.5. SQLite database and TAP correlation tests

Preliminary testing was completed using a set of SQLite data extracted directly from the Tile iOS application database. The data extracted by our parser proved to be a one-to-one match to the input SQLite file. The results of this test are depicted in Fig. 8a and b. For more robust testing, we randomly generated ten SQLite data sets of varying sizes using a Python 3 script. Both the number of geolocation coordinates input and recovered by TAP are depicted in Table 4. For all tested data sets, TAP was able to recover the complete set of data points. We also compared the coordinates and determined that the tool maintained a 100% accuracy rate for identifying valid data points.

5.6. VMEM and TAP correlation tests

Preliminary testing of the memory image parsing functionality was completed on an 8 GB memory image collected while the Tile application was running on a Windows VM. To verify the accuracy

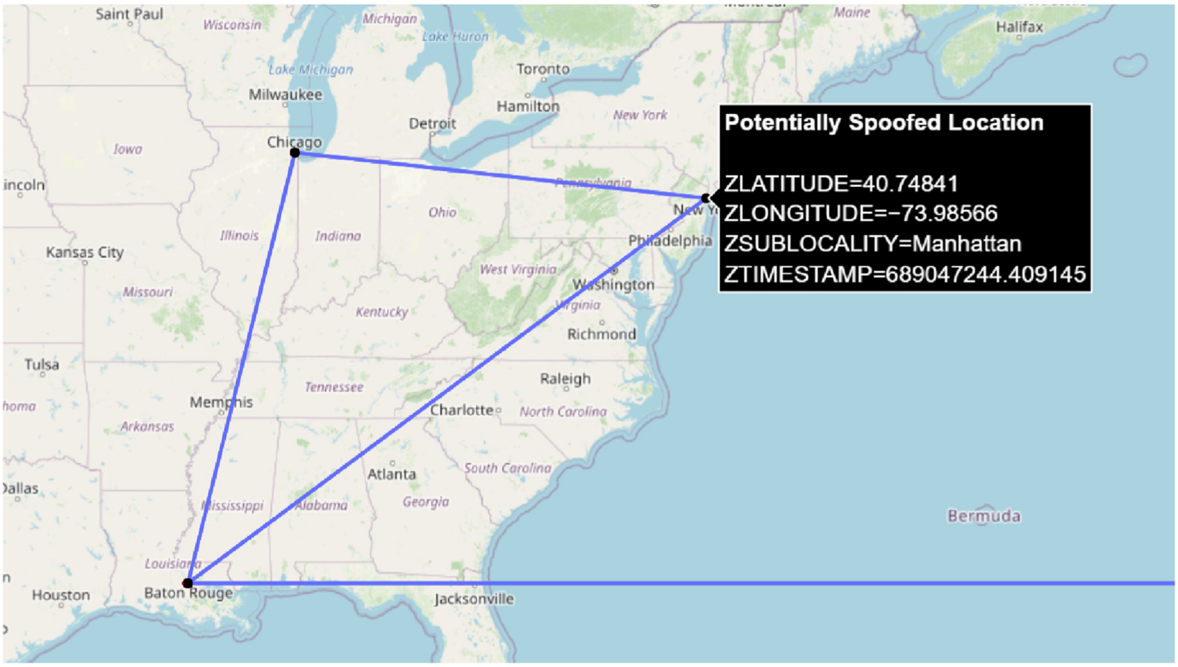
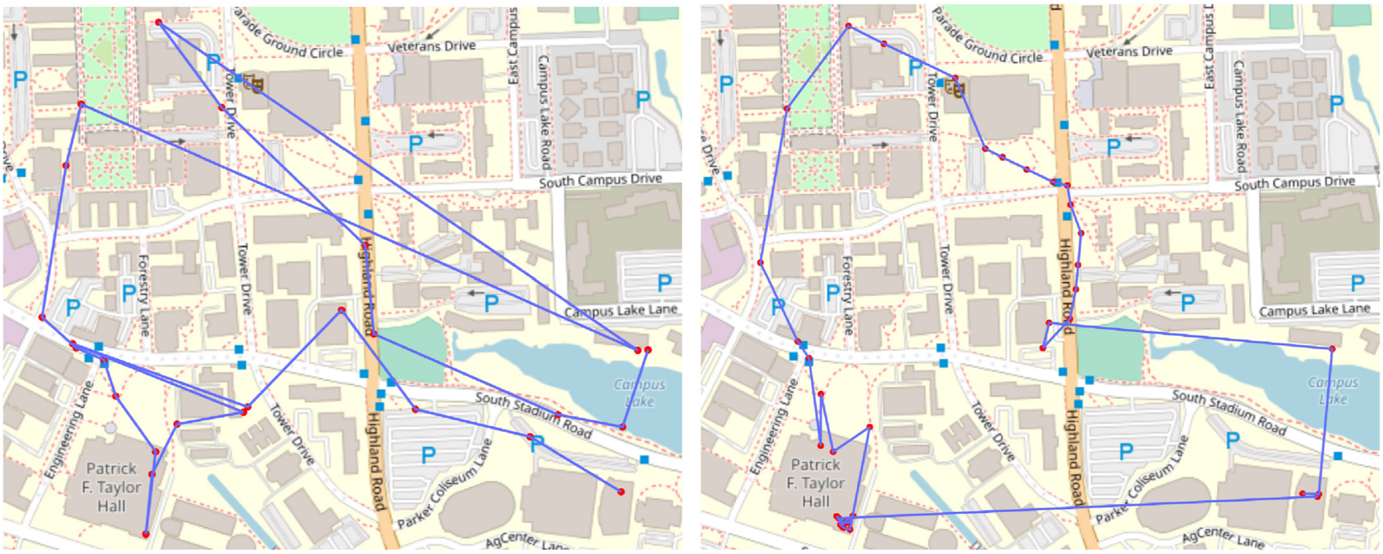


Fig. 6. TAP output of potentially spoofed locations.



(a) Mapping iOS location data.

(b) Mapping VMEM location data.

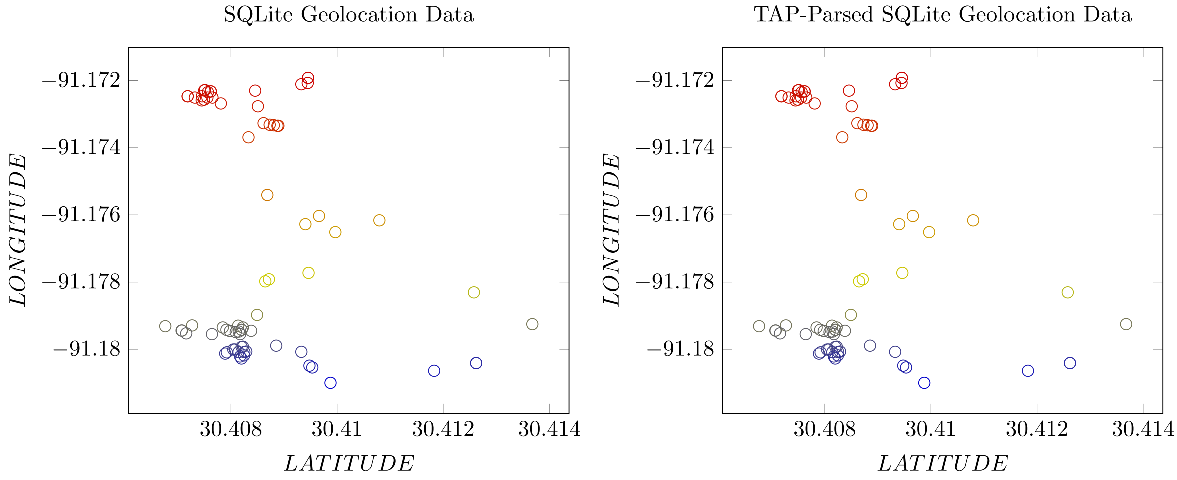
Fig. 7. Result of TAP output with same date range.

and exhaustiveness of TAP's collection method, we created ten 2 GB memory images from a Windows VM. In each image, we embedded varying numbers of geolocation coordinates randomly generated using a Python 3 script. Two of the ten images also contained a number of partial or missing data objects that represented potential smearing during memory collection. TAP attempts to detect and document this corrupt data. The results of a test completed on a 50-coordinate image are depicted in Fig. 9a and b. The data points recovered during memory testing are depicted in Table 5. Cross-analysis of the VMEM input and TAP output yielded a 100% accuracy rate for the identification of valid data points. On average, invalid data points were correctly identified and output to the report 50% of the time.

6. Discussion

Our work expands upon previous research done by (Vance, 2019, 2020; Gazeau and Liu, 2020). We presented a complete analysis of the 2022 Tile devices spanning multiple platforms, and we developed a tool to parse the data sets found. Comparing our findings with earlier work, it appears that the Life360 acquisition has not changed how the Tile application stores data. The appearance of plaintext firmware has been patched; however, the firmware is still accessible as a binary file.

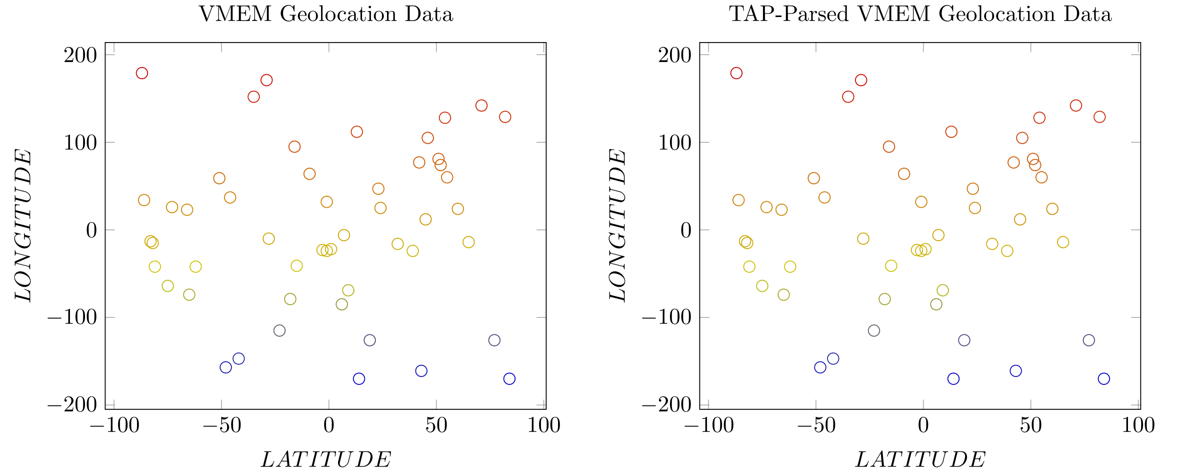
As discussed in Section 4, location data was found on all of the devices investigated. We also discovered log files with information such as MAC addresses, Tile UUIDs, phone battery level, and app



(a) Geolocation coordinates extracted from the iPhone Tile SQLite database.

(b) Geolocation coordinates extracted from the iPhone Tile SQLite database by TAP.

Fig. 8. SQLite Database vs. TAP Output.



(a) Geolocation coordinates extracted from the VMEM image containing 50 data points.

(b) Geolocation coordinates extracted from the VMEM image containing 50 data points by TAP.

Fig. 9. VMEM File vs. TAP Output.

activity. The geographic data from the iPhone and Windows artifacts are timestamped and can be used in cases where mapping out a user's movements are important. With the growth of the personal Bluetooth trackers market, more of these devices will be found during investigations.

It is imperative to know how to investigate Tile trackers as we have already seen cases of their use in crimes (Tile, 2015; Welsch, 2022; Willey, 2018). We provide a forensic technique for investigators to implement when recovering Tile data. It is important that investigators can acquire this data efficiently and from any applicable devices they find. To do this, they must have a well-defined collection technique and accurate data. When done successfully, investigators could recover correlative data between a suspect or victim and a specific place and time.

This technique can estimate a target's location because the Tile application uses the location services of companion devices. These companion devices will often be the user's personal mobile device or computer. The Tile Android application uses reliable utilities from the android.location package. The accuracy of data found in local databases is determined by the companion device's location settings.

Tile premium, the paid version of a Tile account, allows users to view location data from the previous 30 days. Our observations reveal that the local iOS database goes back about 30 days. This data is recoverable regardless of premium account status.

We also present a novel examination of the Tile desktop application. We have proved it is possible to collect valuable information for forensic investigations. After a manual analysis of the memory dump, we were able to identify signatures to look for in memory to collect valuable artifacts.

We found that there are more data points stored in memory than there are in the SQLite databases. We can only speculate that this is due to the Tile servers storing more data than the local iOS database saves.

Lastly, we investigated the capability of a bad-actor to spoof their location. There is potential for spoofing to be used as an anti-forensics technique if someone wants to hide their location. More alarmingly, location spoofing affects the location data of other

Table 4
Comparison between the number of geolocation data points in testing of SQLite databases and TAP output.

Number of Data Points	
SQLite Input	**TAP Output**
0	0
10	10
20	20
30	30
50	50
74	74
100	100
500	500
1000	1000
2000	2000
5000	5000

Table 5
Comparison between the number of geolocation data points in testing of VMEM files and TAP output.

Number of Data Points			
VMEM Input		**TAP Output**	
Valid	**Partial/Missing**	**Valid**	**Partial/Missing**
0	0	0	0
10	5	10	3
10	0	10	0
20	10	20	4
20	0	20	0
30	0	30	0
50	0	50	0
100	0	100	0
500	0	500	0
1000	0	1000	0

users' devices as well. Despite this, Tile currently provides no location spoofing detection in their application. To propose an interim solution, TAP provides important functionality to identify and flag potential location spoofing. TAP is limited in its spoofing detection to extreme spoofing. If a location is only slightly spoofed, or the location is slowly changed, TAP will not be able to detect spoofing. We proposed this spoofing detection as a proof of concept and one of many methods that Tile could implement in its application. There has been past research completed on spoofing mitigation in mobile applications (Koh et al., 2016). Implementing spoofing detection should be a priority for Tile to provide a safer experience to users.

TAP is limited by the in-text signature-based approach used to parse data from memory images. Consequently, anti-forensics techniques that inject false data into memory could be deployed due to a lack of origin process checking.

0The Android devices yielded disappointing artifacts. Databases were populated with very few coordinates and they were not connected to timestamps. We did not root our devices since the goal of this research was to aid investigators with a practical way to recover valuable information. Rooting Android or iOS phones is a tedious process, and rooted devices are unlikely to be commonly found.

In contrast, the iPhone and Windows artifacts included both geolocation and timestamp data. In conducting this research, we provide conclusive evidence that personal Bluetooth trackers are important to forensically investigate in criminal cases.

7. Conclusion and future work

Our work focused on collecting forensically relevant artifacts from Tile companion devices. We then analyzed our findings to reveal correlative data that could provide practical insight into an investigation. This practicality is further emphasized by our development of a novel tool that automates data extraction and analysis. Furthermore, the implications involved with storing personal geolocation data is not exclusive to Tile. The methodology described in this paper could be applied to other Bluetooth trackers to evaluate their importance in forensic investigations.

Future work should focus more on the anti-forensics methods that could be deployed against Tile investigators. Tile's Safe and Scan is a necessary feature, but it does not perform at the level of Apple's feature for scanning for foreign AirTags. It would be beneficial for a feature to be created to passively scan for persistent foreign Tile BLE advertisements. Furthermore, Tile should provide some form of location spoofing detection and mitigation. We introduced rudimentary spoofing detection in TAP. More work is required to implement a robust method for detecting spoofing. Our resources available during the spoofing evaluation were limited to the GPS coordinates, timestamps, and Tile UUIDs. Therefore, absolute spoofing detection was out of scope. However, we plan to explore more robust approaches for spoofing detection. One such method could be a positional evaluation of the user's travel path using consecutively collected location data points. Using these points, we could calculate some threshold for linearity beyond common human travel paths. It would also be beneficial to see if information can be obtained from the physical Tile hardware. Further investigation into the Tile Ultra as being the first-of-its-kind iOS and Android Ultra-Wideband (UWB) tracker would also be beneficial to the field (Gartenberg, 2021).

Acknowledgements

We would like to thank, Farkhund Iqbal, our shepherd and our reviewers for their helpful comments. We would also like to thank Raphaela Mettig and Andrew Case for the time they took to review our paper and offer suggestions.

This material is based upon work supported by the National Science Foundation under Grant Number 1946626. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also want to thank Louisiana State University for providing funding for equipment.

References

Acumen, 2022. Smart tracker market analysis - global industry size, share, trends and forecast 2022 - 2030. https://www.acumenresearchandconsulting.com/smart-tracker-market.

Archie, A., 2022. Two women who allege they were stalked and harassed using airtags are suing apple. https://www.npr.org/2022/12/07/1141176120/apple-airtag-harassment-stalker-lawsuit?utm_campaign=npr&utm_term=nprnews&utm_source=facebook.com&utm_medium=social&fbclid=IwAR1PcEl8_an2iWb3GBHs8ag3vndDvVzRlSOBVMIbk-m2O6C5dm9I122HPjs.

Bays, J., Karabiyik, U., 2019. Forensic analysis of third party location applications in android and ios. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6.

Briggs, J., Geeng, C., 2022. Ble-doubt: smartphone-based detection of malicious bluetooth trackers. In: 2022 IEEE Security and Privacy Workshops (SPW), pp. 208–214.

Case, A., Richard, G.G., 2016. Detecting objective-c malware through memory forensics. Digit. Invest. 18, S3–S10.

Case, A., Richard III, G.G., 2017. Memory forensics: the path forward. Digit. Invest. 20, 23–33.

Casey, P., Lindsay-Decusati, R., Baggili, I., Breitinger, F., 2019. Inception: virtual space in memory space in real space – memory forensics of immersive virtual reality with the htc vive. Digit. Invest. 29, S13–S21.

Chernyshev, M., Zeadally, S., Baig, Z., Woodward, A., 2017. Mobile forensics: advances, challenges, and research opportunities. IEEE Secur. Priv. 15 (6), 42–51.

Gartenberg, C., 2021. The tile ultra takes on airtags with uwb-powered ar tracking, coming early 2022. https://www.theverge.com/2021/10/12/22716870/tile-ultra-bluetooth-tracker-uwb-ar-early-2022.

Gazeau, V., Liu, Q., 2020. Catch me if you can: analyzing geolocation artifacts left by the tile application on iphones. Acta Sci. Comput. Sci. 2 (10), 38–43.

Geolocation Privacy Legislation, 2021. https://www.gps.gov/policy/legislation/gps-act/.

Hassenfeldt, C., Baig, S., Baggili, I., Zhang, X., 2019. Map my murder: a digital forensic study of mobile health and fitness applications. In: 'Proceedings of the 14th International Conference on Availability, Reliability and Security', ARES '19. Association for Computing Machinery, New York, NY, USA.

Karpisek, F., Baggili, I., Breitinger, F., 2015. Whatsapp network forensics: decrypting and understanding the whatsapp call signaling messages. Digit. Invest. 15, 110–118 (Special Issue: Big Data and Intelligent Data Analysis).

Knox, S., Moghadam, S., Patrick, K., Phan, A., Choo, K.-K., 2020. What's really 'happning'? a forensic analysis of android and ios happn dating apps. Comput. Secur. 94, 101833.

Koh, J.Y., Nevat, I., Leong, D., Wong, W.-C., 2016. Geo-spatial location spoofing detection for internet of things. IEEE Internet Things J. 3 (6), 971–978.

Life360, 2021. Life360 to acquire tile, creating the world leader in finding and location solutions. https://www.prnewswire.com/news-releases/life360-to-acquire-tile-creating-the-world-leader-in-finding-and-location-solutions-301430364.html.

Moffitt, K., Karabiyik, U., Hutchinson, S., Yoon, Y.H., 2021. Discord forensics: the logs keep growing. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 993–999.

Perez, S., 2021. Tile secures $40 million to take on apple airtag with new products. https://techcrunch.com/2021/09/16/tile-secures-another-40-million-to-take-on-apple-airtag-with-new-products/.

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.

Satrya, G., Kurniawan, F., 2020. A novel android memory forensics for discovering remnant data. Int. J. Adv. Sci. Eng. Inf. Technol. 10, 1008.

Tile, 2015. Where is my bag? tracking my luggage between connecting flights. https://www.tile.com/en-us/blog/travelhacks-tile-connecting-flights-luggage.

Tile, 2022. Tile's scan and secure feature addresses unwanted tracking. https://www.thetileapp.com/en-us/blog/tile-introduces-scan-and-secure-feature-unwanted-tracking-safety.

Tile. Tracking moving objects with tile. n.d. https://tileteam.zendesk.com/hc/en-us/articles/200591638-Tracking-Moving-Objects-with-Tile.

Vance, C., 2019. Android - locating location data: the tile app. https://blog.d204n6.com/2019/08/android-locating-location-data-tile-app.html.

Vance, C., 2020. ios - tile app part 2: custom artifact boogaloo. https://blog.d204n6.com/2020/09/ios-tile-app-part-2-custom-artifact.html.

Walnycky, D., Baggili, I., Marrington, A., Moore, J., Breitinger, F., 2015. 'Network and device forensic analysis of android social-messaging applications', *Digital Investigation*. In: The Proceedings of the Fifteenth Annual DFRWS Conference, vol. 14, pp. S77–S84.

Weller, M., Classen, J., Ullrich, F., Waßmann, D., Tews, E., 2020. Lost and Found: Stopping Bluetooth Finders from Leaking Private Information, WiSec '20. Association for Computing Machinery, New York, NY, USA, pp. 184–194.

Welsch, Q., 2022. Sheriff's office investigating 'particularly alarming' cyberstalking of teenage girl who found tracker on her car after hoopfest. https://www.spokesman.com/stories/2022/jul/21/sheriffs-office-investigating-cyberstalking-of-tee/.

Willey, J., 2018. Houston woman says ex used 'tile' device to stalk her repeatedly. https://abc13.com/houston-woman-harassment-high-tech-device-stalking/3719155/.