IS DATA RETENTION STILL NECESSARY IN THE AGE OF SMARTPHONES?

MICHAEL SPREITZENBARTH AND SVEN SCHMITT

It is well known that smartphone operating systems persistently store location information in their local storage for various reasons. However, less well known is probably the fact that also various applications do this, too. In this article we will give you some hints where you can find this data on Android smartphones as well as we will present a system with which all this information can be extracted and visualized at the same time. We will also provide you with a comparison of the quality and quantity of location data gathered through data retention in contrast to the data gathered by forensic acquisition.

Location data in mobile phones:

In recent years, new types of mobile phones, so–called smartphones, have permeated the market. Being small personal computers, they offer much more than the possibility to make phone calls and surf the Internet. Within the last two years the mobile phone market has been restructured and the operating system Android has become the market leader with more than 50% of market share and more than 75 million sold units in the fourth quarter of 2011 [Gartner Inc. – Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth – *http://www.gartner.com/it/page.jsp?id=1924314]*. Having such a smartphone in place, more and more users take advantage of the offered variety of applications of third party developers that are directly installed on the phone. So they are able to communicate with friends and relatives via social networks like twitter, Google+ or Facebook. To increase performance of the build–in navigation software and for several other reasons, mobile devices persistently store location data within their own local memory. In April 2011 it was reported that Android and iOS store sensitive geographical data [J. Angwin and J. Valentino–Devries. – Apple, Google Collect User Data. – *http://online.wsj.com/article/SB10001424052748703983704576277101723453610.html]* [J.R. Raphael. – Apple vs. Android location tracking: Time for some truth. – *http://blogs.computerworld.com/18190/*apple android location tracking, August 2011.]. This data is stored in cache files on the system. But not only the operating system generates geographical data. Many apps that provide location–based services create and store such data, too. A short overview of the files, we will analyze in the upcoming sections can be seen in Table 1, all the corresponding apps had the development state of November 2011. Smartphones with Android Gingerbread in version 2.3.4 were used for our experiments and the analysis that can be found in this article (Table 1).

Starting with something easy – the cache files:

Android is maintaining two cache files with location information. One is cache.wifi (a wifi router database with MAC and GPS data of the router) the other is cache.cell (a database with the id of mobile communication cells and their GPS data). These cache files are located at */data/data/com.google.android.location/files/*. Due to the fact that these files are in binary format, the Python code–snippet displayed in Listing 1 should help you to encode the actual data.

Under ideal circumstances you can find up to 200 wifi routers and up to 50 mobile communication cells with the corre-

Table 1. *Android applications and stored location information*

App Name	Stirage Location	Content
System	cache.cell	Last 50 mobile telecommunication cells
	cache.wifi	Last 200 wifi routers
Camera	/sdcard/DCIM/Camera/ /sdcard/external_sd/DCIM/Camera/	Latitude and longitude of picture location
Browser	CachedGeopositions.db	Latitude, longitude, accuracy and timestamp
Twitter	AUTHOR_ID.db Table: statuses	Latitude and longitude of status message
	AUTHOR_ID.db Table: search_queries	Latitude, longitude and radius of location search queries
Facebook	fb.de Table: user_statuses	Latitude and longitude of status message
	fb.de Table: user_values	Latitude, longitude and timestamp of last checkin
Google Maps	da_destination_history	Source and destination of navigation

Table 2. *Important GPS data inside the Exif area [4]*

Tag Name	Field Name	Tag ID
North or South Latitude	GPSLatitudeRef	1
Latitude	GPSLatitude	2
East or West Longitude	GPSLongitudeRef	3
Longitude	GPSLongitude	4
Altitude	GPSAltitude	6
GPS time (atomic clock)	GPSTimeStamp	7
GPS satellites used for measurement	GPSSatellites	8

sponding GPS data and approximate distance in these files. An example of the decoded data can be seen in Listing 2.

Another good point to search for location data – the pictures: Nearly all smartphones have a build–in camera. This camera is able to add special meta data to the pictures the user is taking. This meta data contains the type of the camera, ISO, resolution of the picture, the timestamp when the picture has been taken and location data. If the picture was taken outside a building, the location data is quite accurate and so, this data is qualified for an exact movement profile. To find this data inside a JPEG picture you have to search for the Exif [Standard of Japan Electronics and Information Technology Industries Association – Exchangeable image file format for digital still cameras: Exif Version 2.2 – *http://www.exif.org/Exif2–2.PDF]*

Listing 1. *Python code-snippet to encode the location cache files of an android system*

```
outputFile = open("OUTPUT_FILENAME", 'a+')
cacheFile = open("CACHE_FILENAME", 'rb')
version, entries = struct.unpack('>hh', cacheFile.read(4))
i = 0
while i < entries:
   key = cacheFile.read(struct.unpack('>h', cacheFile.read(2))[0])
   (accuracy, confidence, latitude, longitude, readtime) = struct.unpack('>iiddQ', cacheFile.read(32))
   outputFile.write('%25s %7d %5d %10f %10f %s \n' % (key,accuracy,confidence,latitude,longitude,time.strftime("%x %X %z",
                    time.localtime(readtime/1000))))
   i=i+1
cacheFile.close()
outputFile.close()
```

Listing 2. *Decoded cache.wifi and cache.cell*

```
  key    accuracy  confidence    latitude    longitude          timestamp

                    00:1e:58:82:79:31     55         92         49.368610    8.587524    09/05/11 04:26:12 +0200
                    00:23:08:ae:29:90    104         87         49.368626    8.588344    09/05/11 04:26:12 +0200

                    228:1:606:430744    1623         75         47.257888    7.695389    08/13/11 12:04:21 +0200
                    228:1:606:430742    1433         75         47.266354    7.711417    08/13/11 12:06:33 +0200
```

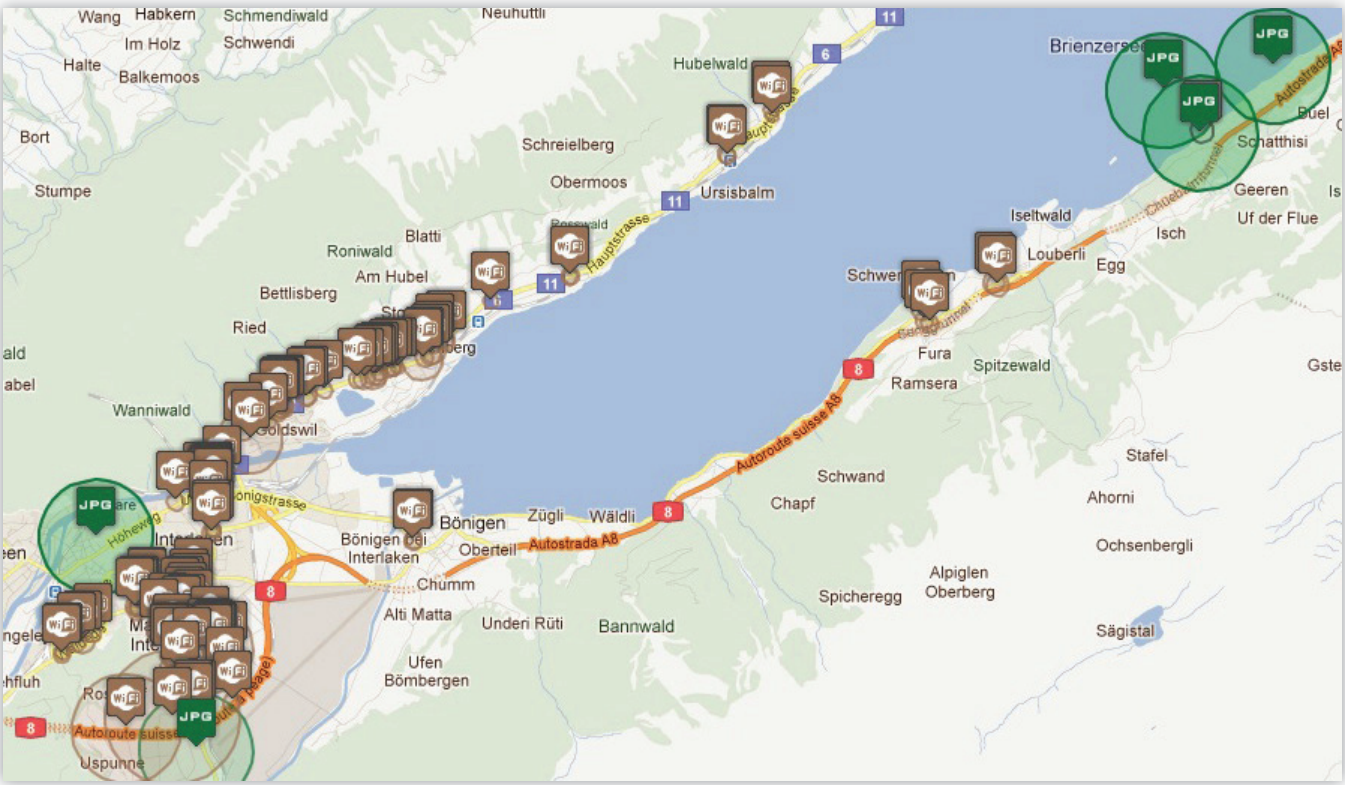

Figure 1. *Movement profile generated from data stored on one of our smartphones*

area in the byte code of the picture file. The interesting parts of the Exif area are listed in Table 2. On most of the smartphones you can find the pictures either in */sdcard/DCIM/Camera/"* or in */sdcard/external_sd/DCIM/Camera/* (Table 2).

Looking for the needle in a haystack – the applications:
As mentioned before, there are thousands of applications in the Android–Market, which request the permission to access location data while installing the app. If you try to generate a movement profile of a smartphone user, the databases of these apps are a good point to search for further location data. In this article we will focus on some of the most commonly installed apps: Twitter, Google Maps, the Android Browser and Facebook.

Each Android application has its own directory, either on the internal memory or on the external sd–card. The structure of the application directories is */data/data/PACKAGE_NAME/*. Inside this directory you normally find a subdirectory with the SQLite databases of the application that we will explain in the upcoming sections.

To get the interesting information from Twitter, you have to analyze the database that can be found in */data/data/com.twitter.android/databases/USERID.db*. In this database the table *statuses* is located. This table holds all status updates that have been twittered by the user. Each status is stored with the corresponding status content, a timestamp, the user id, latitude and longitude. Another interesting table within this database is *search_queries*. This table holds meta data to every search the user did through the Twitter app with the actual position of the user (latitude and longitude), time and the query.

Google Maps has one database of interest for gathering location information: */data/data/com.google.android.apps.maps/databases/da_destination_histor* Here, the application stores all navigations the user has requested. For a forensic acquisition only the start point of a navigation is of interest, because there is no evidence that the user really travelled to the destination.

Another application where you can find traces of location data is the build–in Android Browser. In the database directory of this app you can find a file called *CachedGeopositions.db*, which contains latitude, longitude and a timestamp of the last position the smartphone was active and has used the browser. This data is used for location–based results of Google search queries.

The last application we will analyze in this article is the Facebook app. Within the main database file *fb.db* are two tables of interest for our investigation: *user_statuses* and *user_values*. In the first table (*user_statuses*) you are able to find latitude and longitude of each status message the user posted on his wall (assuming that the user didn't switch off the positioning service of Facebook). In the second file you can find the last position the user did a so–called check–in with corresponding latitude, longitude and timestamp.

Building the big picture:
After we got all the data from cache files, pictures and application databases, we now want to merge these data to generate a movement profile of the smartphone user. In our approach we use the Google Maps JavaScript API [Google Inc. – Google Maps JavaScript API v3 – *http://code.google.com/intl/de–DE/apis/maps/documentation/javascript/]* and create an interactive map, with every data point and the corresponding accuracy displayed as a circle with an icon representing the kind of data. When moving the mouse to one of the icons, some more information like *name of the picture* and *time the picture was taken* will be displayed. An example of such an interactive map can be seen in Figure 1.

Generating movement profiles fully automated – ADEL: ADEL (Android Data Extractor Lite) [M. Spreitzenbarth, S. Schmitt and F. Freiling – Forensic Analysis of Smartphones: The Android Data Extractor Lite (ADEL) – The 2011 ADFSL

Conference on Digital Forensics, Security and Law, Richmond, Virginia, 2011)] is a forensic data extraction and analysis tool for the Android platform. The tool consists of multiple scripts (modules) written in Python and can be extended rather easily. It is able to automatically dump predefined SQLite database files from Android devices as well as it can extract the content stored within the dumped databases. A flow chart showing the structure of ADEL is depicted in Figure 2. In the first step, ADEL establishes a connection to an Android device via the Android Debugging Bridge (adb), dumps predefined SQLite databases off the phone and stores them on the investigator's machine (dump module). All of the following steps are performed on the created database copies in read–only mode, thus ensuring the integrity of underlying data (Figure 2).

In the second step contents within the dumped database copies are analyzed and extracted (analysis module). Therefore we developed a specialized parser module for the SQLite database file format [SQLite. – The SQLite Database File Format. – *http://www.sqlite.org/leformat2.html]*. It extracts the contents by directly parsing the database file and does not issue SQL statements to a running SQLite instance. After having extracted the contents, an XML–based report is generated in order to ease further use and depiction of data (report module). The report can, e.g., be viewed in an ordinary web browser and be refurbished with the help of an XSL file.

In the current development state, the following information can be dumped and analyzed with ADEL:

- telephone and SIM–card information,
- address book and call lists,
- calendar entries,
- browser history and bookmarks,
- SMS messages and
- location data of the most popular apps and the system.

One disadvantage of ADEL is the fact that it can only be used with mobile phones that provide root access and an insecure kernel flag.

Some background information on data retention:
In 2006 the European Union issued a directive [European Parliament and the Council of the European Union. – Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. – Official Journal of the European Union, L 105:0054–0063, 2006.] to harmonize the regulations within the EU member states regarding the retention of data generated by publicly available electronic communications services. One main goal of this directive was to allow law enforcement to access traffic data of suspects, e.g., to find out with whom the suspect had communicated or which digital services he had used. In addition to data about individual communications, the directive also demanded that certain location data are retained. More specifically, the directive requires retaining the following data for at least six months:

- Identity and exact GPS position of the radio cell from which the user started a phone call.

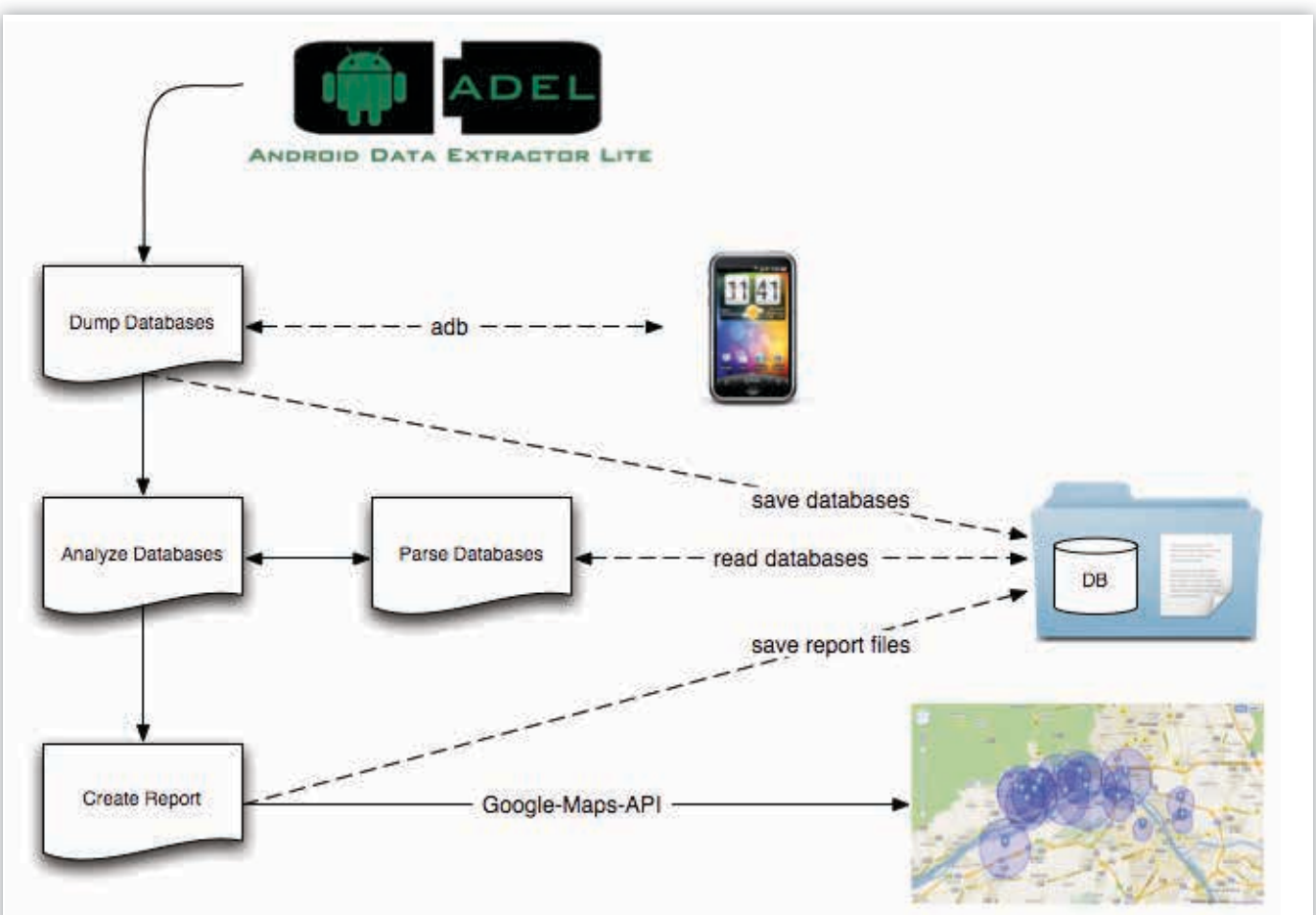

Figure 2. *System workflow of the Android data Extractor Lite*

Table 3. *Comparison of the data points gathered through ADEL and data retention*

Data Source	Smartphone	Data Retention
Cell ID	50	3223
Wifi	200	--
Twitter	9	--
Facebook	15	--
Pictures	20	--
Android Browser	2	--
Google Maps	4	--

- Identity and coordinates of the radio cell that has been active at the beginning of a GPRS data transmission.
- The time stamp belonging to this data.

Comparison between data retention and forensic acquisition: We used the data set provided by Spitz [ZEIT online. – Tell–all telephone. – *http://www.zeit.de/datenschutz/malte–spitz–data–retention]* as a comparison to our measurements with ADEL. This data set was collected within six months by a large German network operator according to the regulations of the EU data retention directive [European Parliament and the Council of the European Union. – Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. – Official Journal of the European Union, L 105:0054–0063, 2006.] Spitz' data set contains only GPS coordinates of the base station locations and a rough direction of the radio beam. So we had to make an assumption regarding accuracy of these measurements. Since cell site locations are smaller in densely populated areas than in the countryside and Spitz mainly had visited larger cities we assumed that the accuracy was in the range between 501 and 1000 meters most of the time. The rest of the time we assumed accuracy of at least 1000 meters (Table 3).

In Table 3 we provide an overview of the average data that has been restored from the smartphones we had used in two field experiments (one field experiment in late 2011 [M. Spreitzenbarth, S. Schmitt and F. Freiling – Forensic Analysis of Smartphones: The Android Data Extractor Lite (ADEL) – Advances in Digital Forensics VIII, G. Peterson and S. Shenoi, Ed., New York, NY: Springer Science+Business Media, 2012.] and another one in early 2012). We also add to the table some entries that refer to Spitz' data. We scaled down the number of data points in the data set to cover approximately the same time frame that was covered by the field experiments. As one may see clearly, the number of found data points from data retention is by far greater than the numbers found during the forensic analysis of smartphones. However, in this case we are dealing with mobile telephony cells only, while the data records of the smartphones show various other sources. The difference of the number of data records found is probably caused by the fact that the smartphones only save the last 50 mobile telephony cells (Figure 3).

Figure 3 compares the accuracy and number of location information of the smartphones with the accuracy of retained data. On the top of the figure the average smartphone data are shown. On the bottom you see the data of the data retention dataset. Here it is clearly noticeable that the number of data points of data retention are usually much greater as compared to a forensic analysis. If one considers the accuracy of data on the other hand, one can see that data retrieved from forensic

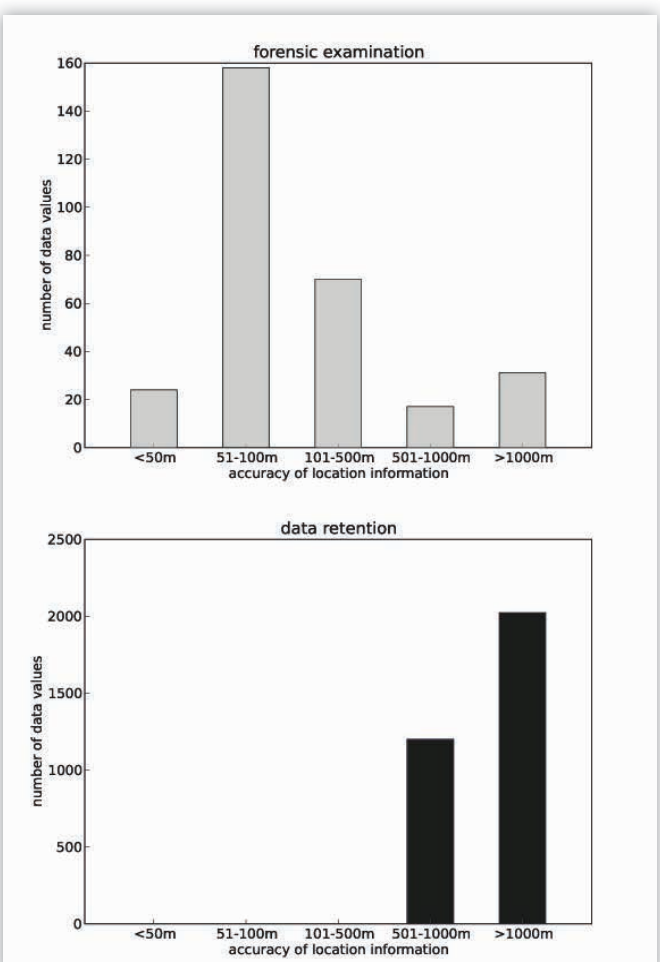

Figure 3. *Number of data values from data retention compared to forensic investigation*

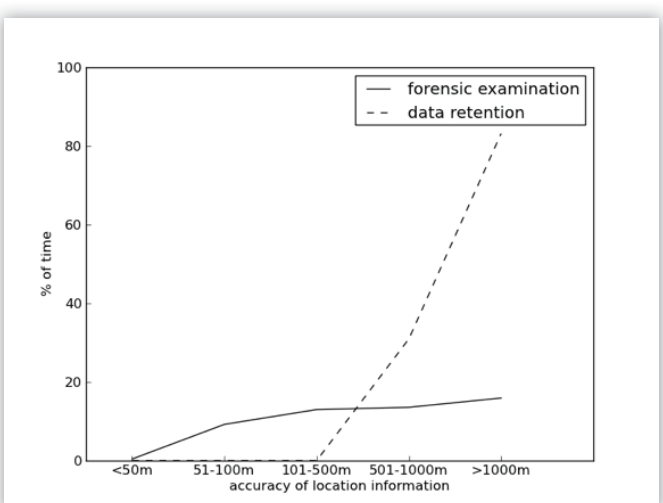

Figure 4. *Percentage of time where the smartphones were traceable*

analysis has its majority in the interval of 50 to 100 meters. The data of the data–retention, in contrast, has its focus in the realm of above 500 meters. From this it can be concluded that the analysis of stored data with the help of ADEL allows for a far more exact positioning of the user.

To draw further conclusions we set the number of data points, including the stored timestamps, in relation to the maximum possible time period (see Figure 4). Since the data basis of our experiment bears on a time frame of two weeks, the maximum time in which the user is traceable sums up to 20.160 minutes. Taking the dashed part of the figure into consideration, it is evident that in our case, when dealing with data–retention, the user is traceable in about 83% of the time. On the contrary, the smartphones of our forensic analysis are on average traceable for about 18% of the time only (see the bold line in Figure 4).

Limitations:
We could also add some privacy enhancing techniques, e.g., to store less information on the smartphone from the beginning. For examples, the option *Use wireless networks* in the device's *Location and Security* settings menu could be disabled. After this step the cache.wifi and cache.cell will be deleted. Further possibilities to reduce storage of location information are to turn off the options *Geotagging* in the camera settings and *Use my location* in the privacy settings of the device. In any case, when dealing with location information one has to consider the possibility that retrieved data may not be reliable to a certain extent. This holds true for location data regarding wifi routers in particular since this data is sent to Google as soon as a wifi router is found for the first time. Furthermore, when dealing with apps like Facebook and Google+ it is possible to link to a certain location although the user is currently not there.

Conclusion

On the headline of this article we raise the question if data retention is still necessary. Unfortunately, the answer is not obvious. Comparing the two analyses it is evident that the data of the forensic analysis are far more precise with respect to the positioning. However, data also exhibits clearly more time–related gaps. In case of crime–related analysis a positioning of 18% is quite low as compared to the data–retention with about 83%. However, if the eligible time lies within the range of available data, a forensic analysis will deliver considerably better results since the exactness of retrieved data is significantly greater, allowing for a more precise assignment of user and location.

MICHAEL SPREITZENBARTH

is a PhD student at the Friedrich–Alexander–University, Erlangen–Nuremberg. He is doing his main research in mobile phone forensics and the analysis of mobile security threats like malicious applications and information leakage. If you are interested in further news and insights in the field of Android forensics as well as mobile security threats feel free to visit http://forensics.spreitzenbarth.de

SVEN SCHMITT

is an external PhD Student at the Friedrich–Alexander–University, Erlangen–Nuremberg. His research interests in the area of digital forensics include database forensics and live forensics.