

Zur Komplexität der Mobilfunkforensik am Beispiel des iPhone

Harald Baier¹ • Achim Brand² • Christian Dichtelmüller³ • Björn
Roos³

¹Hochschule Darmstadt, Center for Advanced Security Research Darmstadt
harald.baier@h-da.de

²Hochschule Darmstadt, Center for Advanced Security Research Darmstadt
achim.brand@stud.h-da.de

³Hochschule Darmstadt
{christian.dichtelmueLLer, bjoern.roos}@stud.h-da.de

Zusammenfassung

Trotz der stetig wachsenden Bedeutung mobiler Endgeräte wie Mobiltelefonen, Smart-Phones, PDAs und Musikgeräten gilt die forensische Untersuchung solcher Geräte als teuer und kompliziert. Teuer, weil der Forensiker typischerweise spezielle, gerätespezifische Soft- und Hardware benötigt. Kompliziert, weil viele proprietäre Datenstrukturen auf Datenträger-, Betriebssystem-, Dateisystem- sowie Anwendungsebene angetroffen werden. Dieser Beitrag verfolgt das Ziel, am Beispiel des iPhone zu zeigen, dass man auch mit begrenzten Ressourcen (finanziell wie zeitlich) eine umfassende forensische Untersuchung mobiler Endgeräte im Rahmen einer praxisorientierten Lehrveranstaltung durchführen kann. Wir zielen auch darauf ab, dass solche Inhalte an anderen Hochschulen verstärkt gelehrt werden.

Schlüsselwörter: Mobile Endgeräte, Computer-Forensik, Lehre, iOS, Datenschutz

1 Einleitung

Laut einer Schätzung des BITKOM sind momentan weltweit ca. fünf Milliarden Mobiltelefone im Einsatz [BITK09]. Alleine im Jahr 2010 wurden beinahe 1,6 Mrd. Geräte an Endkunden verkauft, wovon knapp 300 Mio. Geräte Smartphones waren [Gart10]. Die weitreichenden Einsatzmöglichkeiten eines Smartphones im privaten sowie geschäftlichen Leben sind es, die diese Geräte vor allem für Forensiker interessant machen. Denn die ubiquitäre Verwendung eines solchen Mobiltelefons hinterlässt viele Spuren, die im Falle einer mobilforensischen Auswertung äußerst relevant sein können.

Die eierlegende Wollmilchsau unter den forensischen Tools für mobile Endgeräte gibt es aber auf Grund der Diversität und Proprietät der Hard- und Software bis heute nicht. Zwar gibt es zahlreiche kommerzielle Tools, die behaupten, die meisten Hardware-Schnittstellen sowie Smartphone-Software zu unterstützen (z.B. das celebre Universal Forensics Extraction

Device [UFED]¹); diese kosten aber ca. 10.000 EUR und sind damit zumindest für den Einsatz in der akademischen Ausbildung zu teuer. Bereits 2007 bringt das US-amerikanische National Institute of Standards and Technology die begrenzte Einsatzmöglichkeit selbst teurer Tools in [NIST07] wie folgt auf den Punkt: "Cellular phone manufacturers also tend to rely on assorted proprietary operating systems rather than the more standardized approach found in personal computers. Because of this, the variety of toolkits for mobile devices is diverse and the range of devices over which they operate is typically narrowed to distinct platforms for a manufacturer's product line, an operating system family, or a type of hardware architecture." Seit 2007 hat sich die Situation eher verschlechtert als verbessert, so dass eine forensische Untersuchung mobiler Endgeräte als teuer und kompliziert gilt.

Dieser Beitrag verfolgt daher zwei zentrale Ziele:

- Wir zeigen am Beispiel des iPhone auf, wie der forensische Sicherungs- und Analyseprozess mit frei verfügbaren Tools durchgeführt werden kann. Bei dem verwendeten Gerät handelt es sich um ein iPhone 3G mit 8 GB internem Speicher und der iOS Version 3.1.3.
- Das forensische Vorgehen erfordert für eine Person mit forensischen Grundkenntnissen keinen überproportional hohen Einarbeitungsaufwand. Konkret haben wir die Vorbereitung und Durchführung der forensischen Untersuchung im Rahmen einer praktisch orientierten Lehrveranstaltung an der Hochschule Darmstadt durchgeführt. In einer Gruppe von 3 Studierenden betrug der formale Workload 22,5 ECTS und damit 675 Arbeitsstunden. Grundlage der Lehrveranstaltung war Jonathan Zdziarskis Buch „iPhone Forensics“ [Zdzi08].

Wir wollen mit diesem Beitrag auch für die weitere Verbreitung forensischer Kompetenzen mobiler Endgeräte in der Lehre an anderen Hochschulen werben.

Der weitere Beitrag ist wie folgt aufgebaut: In Abschnitt 2 gehen wir kurz auf verwandte Arbeiten ein. Dann wiederholen wir in Abschnitt 3 kurz die zentralen forensischen Prinzipien, die unserem Untersuchungsprozess zugrunde liegen. In Abschnitt 4 beschreiben wir unser Vorgehen zur Datensicherung und in Abschnitt 5 an Hand ausgewählter Themen unsere Analyse. Wir schließen diesen Beitrag mit einem Fazit in Abschnitt 6.

2 Related Work

Neben den vorgestellten Analysen gibt es noch ein Reihe von weiteren Untersuchungen über mobile Endgeräte. Diese Untersuchungen beziehen sich nicht nur auf das iPhone, sondern auch auf andere Typen von mobilen Endgeräten z.B. solche mit dem Betriebssystem Android.

Jens Heider und Matthias Boll beschreiben in einem von ihnen vorgestellten Paper [HeBo11] die Möglichkeiten zum Auslesen von Passwörtern von einem gesperrten iPhone aus dem "key-chain".

In [Mart08] werden Techniken, Methoden und Werkzeuge aufgezeigt, mit denen sich für einen Forensiker interessante Informationen wie Anrufliste, SMS, Adressbuch, Kalender oder auch Internetverlauf aus mobilen Endgeräten gewinnen lassen.

¹ Siehe: <http://www.cellebrite.com/forensic-products/forensic-products/ufed-physical-pro.html>

Andrew Hoog erläutert in seinem Paper [HoSt10] welche persönlichen Informationen auf einem iPhone gespeichert werden und wie diese Informationen gewonnen und/oder wiederhergestellt werden. Beispiele verbreiteter gespeicherter Informationen sind Anrufliste, SMS, Kontakte, WiFi-Passwörter, Bilder (auch gelöschte) und je nach Untersuchungskriterium weitere.

Im Paper [Alla11] wird das kürzlich bekannt gewordene Location Tracking durch Apple-Geräte wie iPhone oder iPad anhand von Mobilfunkzellen und WiFi Netzen erläutert.

In [LeKe10] wird eine forensische Untersuchung eines mobilen Endgerätes mit dem Betriebssystem Android vorgenommen. Zentrales Ergebnis dieser Publikation ist, dass sich mit ähnlichen und teilweise den gleichen Methoden, welche wir in diesem Paper vorstellen, die gleichen Informationen gewinnen lassen wie beim iPhone.

Das Unternehmen Elcomsoft hat eine Möglichkeit gefunden, die AES-Verschlüsselung des Dateisystems zu brechen [Kata11]. Dadurch sollte es in Zukunft auch möglich sein, für iOS verwendete, verschlüsselte Dateisysteme ohne Mitwirkung des Gerätebesitzers forensisch zu sichern und zu analysieren.

3 Forensische Prinzipien und Ausgangslage

Das sog. Secure-Analyse-Present (S-A-P) Modell kann als Leitfaden für jegliche Art forensischer Untersuchung herangezogen werden. Das Modell teilt die Untersuchung in drei Phasen ein. In der Secure-Phase geht es um die sorgfältige Erfassung aller Daten. Die Analyse-Phase beschäftigt sich mit der Analyse und der objektiven Bewertung der Ergebnisse. In der Present-Phase geht es darum, in welcher Form die gewonnenen Informationen präsentiert werden sollen, um somit die entsprechenden Personen von den Ermittlungsergebnissen zu überzeugen [BSI09].

Da die Arbeiten im Rahmen einer technischen Lehrveranstaltung durchgeführt wurden und die Zielgruppe für die Präsentationsphase nicht definiert war, beschränken wir uns in diesem Beitrag auf die Sicherungs- und Analysephase.

Die Ausgangslage für unsere Untersuchung ist wie folgt: Bei dem untersuchten Gerät handelte es sich um ein bereits jailbreaktes iPhone 3G mit iOS 3.1.3 und 8 GB internem Flash-Speicher, welches sich seit ca. 2 Jahren im täglichen Gebrauch befand. Eine Möglichkeit den Speicher mittels einer Speicherkarte zu erweitern, besteht nicht. Das Dateisystem des Smartphones ist das von Apple entwickelte HFS+ mit einer Blockgröße von 4.096 Bytes. Diese Blockgröße ist abhängig von der iOS-Version. Der gesamte Speicher ist in 2 Partitionen aufgeteilt: Root- und User-Partition. Auf der Root-Partition (Größe: 500 MB) befindet sich das Betriebssystem und auf der User-Partition (Größe: 7,07 GB) werden alle benutzerspezifischen Daten gespeichert. In unserer Untersuchung beschränken wir uns ausschließlich auf die Analyse der User-Partition, da sich hierauf die für eine forensische Untersuchung relevanten Daten befinden.

4 Datensicherung

Das zu untersuchende Gerät befand sich in eingeschaltetem Zustand und besaß keine Code-Sperre. Das bedeutet., dass man das iPhone ohne jegliche Authentifizierung nutzen kann. Um eine Sicherung des internen Speichers durchzuführen, muss das iPhone eingeschaltet sein bzw. werden. Des Weiteren handelt es sich bei dem zu untersuchenden Gerät um ein bereits

gejailbreaktes iPhone und somit kann eine forensisch saubere Datensicherung ohne weitere Probleme erfolgen. Denn mit Hilfe des Jailbreaks kann mit "root"-Rechten auf das iPhone zugegriffen werden und somit wiederum ein Image der User-Partition erstellt werden. Auch wenn das Gerät nicht bereits gejailbreakt gewesen wäre, wäre die saubere Sicherung der Daten dennoch möglich gewesen, ohne die benutzerspezifischen Daten zu ändern. Denn der Jailbreak ändert nur Daten auf der Root-Partition und lässt die Benutzer-Daten bzw. die User-Partition unverändert. Somit widerspricht eine solche Vorgehensweise nicht dem Unverändertheits-Paradigma der Datensicherung.

Zur forensischen Untersuchung des iPhones wurde ein bitgenaues Abbild der Media-Partition mit Hilfe von dd erstellt. Die einzelnen Schritte hierfür, werden im Folgenden erläutert:

- Verbindung zwischen iPhone und Untersuchungs-System via SSH (Secure Shell) über WLAN.
- Start von Netcat (nc) auf dem Untersuchungs-System (siehe Abb. 1).
- Datensicherung (siehe Abb. 2; Eingabe in Shell auf dem iPhone):
 - Aushängen („Unmounten“: `umount -f`, wobei `-f` für force, also das Erzwingen des unmountens steht) der standardmäßig gemounteten User-Partition (`/private/var` ist der Mountpoint von `/dev/rdisk0s2`)
 - Einhängen („Mounten“) der User-Partition mit Read-Only-Rechten, damit sich während der Übertragung nichts mehr auf der Partition ändern kann. Die Partition wird deshalb gemountet, um eventuelle Fehler/Abstürze zu verhindern, falls während der Sicherung Applikationen von der Partition lesen möchten.
 - Disk-Dump (dd) der Partition in Netcat (nc) übergeben und somit auf das Untersuchungs-System übertragen.

```
sherlock@forensics:~$ nc -l 4321 | dd of=./rdisk0s2 bs=4096
sherlock@forensics:~$ shalsum rdisk0s2
83c900e406b2797b99615eb16032271620c7581a rdisk0s2
```

Abb. 1: Netcat (listening) auf Untersuchungs-System

```
iPhone:~ root# umount -f /private/var/
iPhone:~ root# shalsum /dev/rdisk0s2
83c900e406b2797b99615eb16032271620c7581a /dev/rdisk0s2
iPhone:~ root# mount -o ro /private/var
iPhone:~ root# dd if=/dev/rdisk0s2 bs=4096 | nc 192.168.56.101 4321
```

Abb. 2: Kopieren der Media-Partition auf Untersuchungs-System

Das Image hat eine Größe von 7,07 GB und dient als Datenbasis der gesamten Untersuchung. Um zu gewährleisten, dass das soeben erstellte Abbild eine exakte Kopie des Originals darstellt und somit forensisch verwertbar ist, wurde vor der Sicherung ein SHA1-Hashwert der User-Partition erstellt und im Anschluss mit dem SHA1-Hashwert der Kopie verglichen (siehe Abb. 1 und Abb. 2).

5 Analyse

In unserer Analyse sollten gelöschte Daten/Dateien wiederhergestellt, vorhandene sowie gelöschte Dateien untersucht und die Möglichkeit des Erstellens eines Bewegungsprofils des Benutzers geprüft werden. Die gesamte Untersuchung erfolgte auf dem dd-Image.

Es existieren 2 Möglichkeiten das Image zu untersuchen: Logisch und physisch. Unter der physischen Untersuchung verstehen wir eine Analyse, welche sich auf die raw-Daten des Images beziehen, d.h. z.B. eine Suche nach gelöschten Dateien mittels File Carving. Unter der logischen Untersuchung wird eine Analyse und Interpretation der Dateiinhalte verstanden.

5.1 Gelöschte Dateien

Da als Dateisystem HFS+ verwendet wird, ist eine Wiederherstellung gelöschter Dateien mit Hilfe des Dateisystems nicht möglich. Denn HFS+ verwendet einen B-Baum, um die Metadaten zu organisieren. Wird eine Datei gelöscht, wird der B-Baum unmittelbar danach reorganisiert und somit die Metadaten der Datei überschrieben. Dies wiederum hat zur Folge, dass die gelöschte Datei mit Hilfe des Dateisystems nicht mehr aufgefunden werden kann [BuFe08]. Somit besteht die einzige Möglichkeit, gelöschte Dateien wiederherzustellen, in dem sog. File Carving.

Mit Hilfe eines durchgeführten File Carvings mit Scalpel² wurde eine Vielzahl gelöschter Daten wiederhergestellt. Darunter befanden sich vor allem eine Vielzahl an Bildern (>27.000). Ein Teil dieser Bilder waren forensisch sehr interessant, da diese mit der iPhone eigenen Kamera erstellt wurden und somit mit GPS-Daten versehen sind. Teilweise waren diese Bilder bereits seit beinahe zwei Jahren gelöscht. Mit diesen Bildern und den darin enthaltenen Geoinformationen lässt sich ein Bewegungsprofil des iPhone Benutzers erstellen und mit Hilfe von Google Maps visualisieren. Abb. 3 zeigt ein solches Profil, bei dem die einzelnen Marker in chronologischer Folge nummeriert wurden. Hiermit lässt sich feststellen, dass der Benutzer sich hauptsächlich im Rhein-Main-Gebiet, sowie in Österreich/Kärnten aufgehalten hat. Auf die Erstellung des Bewegungsprofils wird in Abschnitt 5.2 noch genauer eingegangen.

Des Weiteren wurde eine Menge (>100) von Snapshots (Screenshots des Bildschirminhalts) gefunden, welche das Gerät eigenständig bei jedem Drücken des sog. „Home Buttons“ anlegt. Auch hier können viele interessante Informationen gewonnen werden, da es sich hierbei auch um Snapshots der Anrufliste, von E-Mails, Notizen u.v.m. handelt (siehe Abb. 4).

² Siehe: <http://www.digitalforensicssolutions.com/Scalpel/>



Abb. 3: Bewegungsprofil anhand der gefundenen Bilder



Abb. 4: Wiederhergestellte Snapshots

5.2 Bewegungsprofil anhand gespeicherter Bilder

In dem sichergestellten Image befinden sich 28 Bilder im 100APPLE-Ordner (siehe Abb. 5). In diesem Ordner werden Bilder gespeichert, die mit der iPhone-eigenen Kamera erstellt und welche, die mit iTunes synchronisiert wurden. Von diesen 28 Bildern enthalten 18 Bilder Geoinformationen. Mit Hilfe von Scalpel konnten jedoch 46 Bilder mit Geoinformationen gefunden werden. Somit wurden 28 gelöschte Bilder mit Geoinformationen wiederhergestellt. Diese Informationen können mit Hilfe von „exifprobe“ (siehe Abb. 6) ausgelesen und ein Bewegungsprofil des Benutzers erstellt werden. Darüber hinaus gibt exifprobe auch Aufschluss darüber, wann ein Bild erstellt wurde.

```
sherlock@forensics:/mnt/iphone$ ls mobile/Media/DCIM/100APPLE/ | wc
28      28     364
```

Abb. 5: Anzahl Dateien im 100APPLE-Ordners

```
sherlock@forensics:~/scalpel-output/pics$ exifprobe -L 00060572.jpg
...
JPEG.APP1.If0.Exif.DateTimeOriginal      = '2009:06:27
15:04:20'
...
JPEG.APP1.If0.Gps.LatitudeRef            = 'N'
```

JPEG.APP1.Ifد0.Gps.Latitude	= 47,11.98,0
JPEG.APP1.Ifد0.Gps.LongitudeRef	= 'E\000'
JPEG.APP1.Ifد0.Gps.Longitude	= 14,44.7,0
JPEG.APP1.Ifد0.Gps.TimeStamp	= 15,4,17.87
...	

Abb. 6: Geoinformationen eines wiederhergestellten Bildes

Um zu bestimmen, wo das Bild ein Bild aufgenommen wurde, kann Google Maps verwendet werden. Das iPhone verwendet für die Speicherung der GPS Koordinaten das Format "Grad, Dezimalminuten": DD,MM.MM (D = Degrees; M = Minutes). Für Google Maps wird jedoch das "Grad"-Format DD.DDDD benötigt. Hierfür müssen die iPhone-Koordinaten in folgender Weise umgerechnet werden: $DD + \frac{MM.MM}{60}$

Tab. 1: Umrechnung von GPS Koordinaten

Breitengrad	Längengrad
N 47,11.98,0	E 14,44.7,0
$47 + \frac{11.98}{60}$	$14 + \frac{44.7}{60}$
= 47.19966666666667	= 14.745
⇒ Eingabe bei Google Maps : 47.19966666666667, 14.745	

Nachstehende Abbildung zeigt das Ergebnis der Koordinaten-Eingabe in Google Maps:

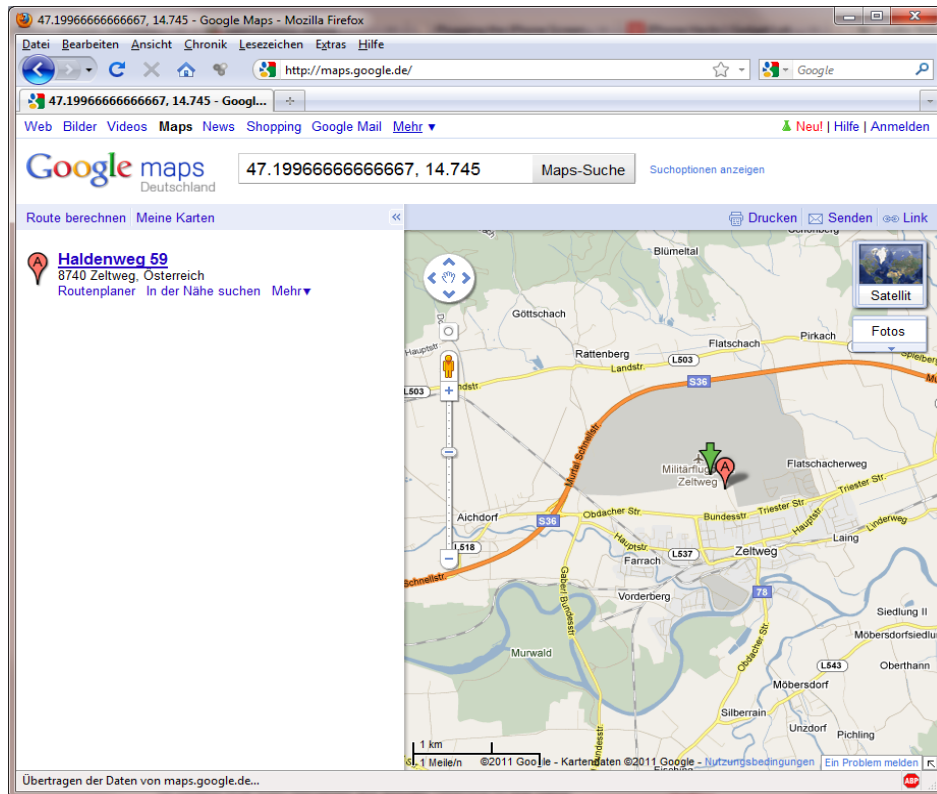


Abb. 7: Darstellung der Koordinaten in Google Maps

Um ein Bewegungsprofil zu erstellen, wurden die Geoinformationen aller 46 Bilder automatisiert ausgelesen und in Google Maps als POIs³, samt Datum und Uhrzeit der Aufnahme, dargestellt. Dies wurde mit einem PHP-Skript realisiert (siehe Abb. 3).

Im Anschluss können weitere Informationen aus den Bildern extrahiert werden. Zum Beispiel zeigen die Details des Bildes 00060572.jpg, dass das Bild am 27.06.2009 um 15:04:20 Uhr am Militärflugplatz Zeltweg (Österreich) aufgenommen wurde. Eine Google Suche nach dem Datum und dem Ort der Aufnahme ergibt, dass sich der Benutzer bzw. das iPhone auf der Flugshow „AIRPOWER09“⁴ des Militärs befunden haben muss. Das zu erkennende Bild – eine Kampfjet – untermauert dies. Solche Informationen könnten für eine forensische Untersuchung sehr hilfreich sein, da es z.B. Augenzeugen geben könnte, die den Benutzer des iPhones dort gesehen haben.

5.3 Untersuchung der Anrufliste

Die Datei `"/mobile/Library/CallHistory/call_history.db"` ist eine SQLite3 Datenbank, welche die Anrufliste darstellt. Sie speichert Informationen zu den letzten 100 getätigten Telefonaten, welche mit diesem Gerät durchgeführt wurden. Wird das 101. Telefonat geführt, wird der älteste Anruf aus der Tabelle gelöscht und dafür Informationen zu dem neuen Gespräch eingefügt. In der Tabelle „call“ befinden sich Informationen in folgender Form (siehe Abb. 8): Indexnummer des Eintrages | Telefonnummer des Gesprächspartners | Anfangszeit des Gespräches (Unix Zeitstempel) | Dauer des Gesprächs in Sekunden | Eingehender oder ausgehender Anruf, wobei ungerade Zahlen einen ausgehenden und eine gerade Zahl einen eingehenden Anruf kennzeichnen.

```

sherlock@forensics:/mnt/iphone$ sudo sqlite3
mobile/Library/CallHistory/call_history.db
SQLite version 3.7.2
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> select * from call;
1049|+49151121XXXXX|1292530170|0|4|-1
1051|+49151121XXXXX|1292530445|647|5|-1
...
1149|+49160978XXXXX|1293039479|0|4|-1
sqlite>

```

Abb. 8: Auszug aus der Tabelle „call“

Die Interpretation der Information soll am Beispiel des Eintrages mit dem Index 1051 verdeutlicht werden:

- Nummer des Gesprächspartners: +49151121XXXXX
- Beginn des Gesprächs: 1292530445 = Thu Dec 16 21:14:05 2010
- Dauer des Gesprächs: 647 Sekunden

³ Point of Interest

⁴ Siehe: <http://www.airpower09.at/>

- Eingehendes oder ausgehendes Gespräch: 5 = ungerade = ausgehend

Abb. 9 zeigt, wie ein Unix Zeitstempel in ein für Menschen leserliches Format umgewandelt werden kann:

```
sherlock@forensics:/mnt/iphone$ perl -e 'require "ctime.pl"; print
ctime(1292530445) . "\n";'
Thu Dec 16 21:14:05 2010
```

Abb. 9: Umwandlung eines Unix Zeitstempels

Neben den in der Tabelle gespeicherten Informationen besteht die Möglichkeit über einen Strings Dump der call_history.db zusätzliche Informationen zu gewinnen: Ein Telefonat könnte zwar in der Tabelle gelöscht worden sein, jedoch noch immer in der Datenbank-Datei vorhanden sein (siehe Abb. 10).

```
sherlock@forensics:/mnt/iphone$ sudo strings
mobile/Library/CallHistory/call_history.db
...
+43699195XXXXM
...
call
data
```

Abb. 10: Strings Dump der Anrufliste

Anmerkung:

OpenSource Tools zur effizienten und komfortablen Wiederherstellung gelöschter SQLite3-Datenbank-Einträge wurden nicht gefunden. Jedoch konnte eine kommerzielle Software namens "epilog⁵" gefunden werden, welche eine weiterführende Analyse von SQLite-Datenbanken zulässt.

Dieses Verfahren der Datenanalyse lässt sich auf alle SQLite3 Datenbanken anwenden. Weitere forensisch interessante Informationen finden sich u.a. in folgenden Dateien:

- /mobile/Library/Calendar/Calendar.sqlitedb: In dieser Datenbank werden sowohl vom Benutzer eingestellte Alarmer als auch eingetragene Termine gespeichert.
- /mobile/Library/SMS/sms.db: Hierin werden alle gesendeten und erhaltenen Textnachrichten gespeichert. Inhalte bleiben auch nach einem manuellen Löschen über das „Nachrichten App“ erhalten.
- /mobile/Library/Notes/notes.db: Alle Notizen werden in dieser Datenbank gespeichert.
- /mobile/Library/Caches/MapTiles/MapTiles.sqlitedb: Speichert den zuletzt angeschauten Kartenausschnitt
- Ab iOS 4: /root/Library/Caches/locationd/consolidated.db: Location-Tracking des Benutzers, in dem darin Daten über ausgewählte Funkmasten und umgebende WLANs gespeichert werden [Alla11]

⁵ Trial Version: <http://ccl-forensics.com/Software/epilog-from-ccl-forensics.html>

5.4 Untersuchung des Keyboard Caches

Vom Benutzer eingegebener Text wird in binärer Form in der Datei „/mobile/Library/Keyboard/de_DE-dynamic-text.dat“ (da das iOS auf Deutsch eingestellt ist) gespeichert, um die Texterkennung zu verbessern, was wiederum eine erneute Texteingabe erleichtern soll. Eingaben aus vielen verschiedenen Apps werden hier gespeichert. Z.B. aus SMS, Notizen oder auch Eingaben in einen Web Browser. Passwörter, welche in spezielle Passwortfelder eingegeben wurden, werden hier nicht gespeichert, dennoch könnten dort relevante Informationen zu finden sein. Wird z.B. eine SMS mit dem Inhalt "Mein Passwort lautet XYZ" versendet, könnte diese Information dort zu finden sein. Abbildung 11 zeigt einen Auszug dieser Datei.

```
sherlock@forensics:~$ sudo xxd
/mnt/iphone/mobile/Library/Keyboard/de_DE-dynamic-text.dat|less
0000000: 4479 6e61 6d69 6344 6963 7469 6f6e 6172  DynamicDictionar
0000010: 792d 3400 0000 0280 6461 7300 6461 6368  y-4.....das.dach
0000020: 7465 006d 6972 0064 6f63 6800 6d65 696e  te.mir.doch.mein
0000030: 0068 6f6c 656e 0077 6972 006e 6163 6800  .holen.wir.nach.
0000040: 6861 7500 6175 6600 6469 6500 6465 7200  hau.auf.die.der.
0000050: 7765 7474 656e 0064 6173 7300 6170 6f74  wetten.dass.apot
0000060: 6865 6b65 0067 7574 656e 006d 6f72 6765  heke.guten.morge
0000070: 6e00 6672 6175 0064 6572 006b 6e6f 7066  n.frau.der.knopf
0000080: 0076 6f6e 0061 6e66 616e 6700 616e 0064  .von.anfang.an.d
...
```

Abb. 11: Auszug der de_DE-dynamic-text.dat

5.5 Untersuchung von Google Maps Daten

Um Informationen über die letzten Anfragen bei Google Maps zu erhalten, lohnen sich drei Dateien:

- /mobile/Library/Maps/History.plist: Speichert die letzten Suchanfragen
- /mobile/Library/Preferences/com.apple.Maps.plist: Speichert Koordinaten des zuletzt angeschauten Ortes
- /mobile/Library/Caches/MapTiles/MapTiles.sqlitedb: Speichert den zuletzt angeschauten Kartenausschnitt

Anmerkung:

Die Informationen in den Dateien History.plist und com.apple.Maps.plist sind als binäres XML Format gespeichert. Zur Konvertierung wird ein Perl-Skript – `plutil.pl`⁶ – verwendet, welches eine Konvertierung von binärem XML zu XML vornimmt. In der „History.plist“ werden u.a. Ort der Suche, Längen- und Breitengrad des Ortes, sowie das Zoom Level gespeichert. Anhand der Datei MapTiles.sqlitedb kann der zuletzt angeschaute Kartenausschnitt wie folgt rekonstruiert werden:

⁶ Quelle: <http://scw.us/iPhone/plutil/plutil.pl>

```

sherlock@forensics:~$ sudo sqlite3
/mnt/iphone/mobile/Library/Caches/MapTiles/MapTiles.sqlitedb
SQLite version 3.7.2
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .output MapTiles.sql
sqlite> .dump images
sqlite> .exit
sherlock@forensics:~$ perl parse_maptiles.pl MapTiles.sql

```

Abb. 12: MapTiles aus Datenbank exportieren

Das Perl-Skript `parse_maptiles.pl`⁷ konvertiert die Datei `MapTiles.sql`. Es entsteht der Ordner „`maptiles-output`“ mit folgendem Inhalt:

```

sherlock@forensics:~$ ls maptiles-output/
17228,11127@15.png 17229,11128@15.png 17230,11128@15.png
17231,11128@15.png
17228,11128@15.png 17229,11129@15.png 17230,11129@15.png
17231,11129@15.png
17228,11129@15.png 17229,11130@15.png 17230,11130@15.png
17232,11127@15.png
17229,11127@15.png 17230,11127@15.png 17231,11127@15.png
17232,11128@15.png

```


Abb. 13: Inhalt des Ordners `maptiles-output`

Durch entsprechende Anordnung der MapTiles kann der komplette Kartenausschnitt, welcher zuletzt angeschaut wurde, rekonstruiert werden (siehe Tab. 2).

Tab. 2: Google Maps – Map Tiles

	17228	17229	17230	17231	17232
11127					
11128					
11129					

⁷ Dieses Skript wurde vom Buch iPhone Forensics entnommen [Zdzi08]

11130					
-------	--	---	---	--	--

6 Fazit

Da wir für unsere Untersuchungen nur Open-Source-Software und ein nicht sehr umfangreiches Buch verwendet haben, können auch Forensiker mit wenig Erfahrung auf dem Gebiet der Mobilfunkforensik gute Ergebnisse liefern. Ein Einsatz in Lehrveranstaltungen mit begrenzten zeitlichen und finanziellen Ressourcen ist sehr gut möglich.

Anhand der Untersuchungen der Geräte lässt sich sagen, dass das iPhone als Smartphone im Business Bereich nicht geeignet ist. Es können ohne größere Probleme Daten untersucht bzw. gelöschte Daten wiederhergestellt werden. Durch die Methode von Mitarbeitern des Fraunhofer-Instituts für Sichere Informations-Technologie (SIT) können auch Passwörter (z.B. von WLANs und VPNs) ausgelesen werden [HeBo11]. Nicht nur durch die Standardapplikationen, sondern auch durch weitere Applikationen, welche hier nicht untersucht wurden, lassen sich zusätzliche Informationen gewinnen, die aus forensischer Sicht interessant sein können. Viele Anwendungen nutzen beispielsweise die GPS-Funktion des iPhones, so dass sich weitere Informationen auf Grund der Verknüpfung zwischen den Applikations- und GPS-Daten gewinnen lassen.

Mit dem iOS 4 hat Apple die Möglichkeit geschaffen, seine Daten besser zu schützen. Apple hat Schutzmechanismen eingeführt, über welche die Apps Daten im Dateisystem und in der Keychain zusätzlich sichern können, indem sie mit Attributen wie „NSFileProtectionComplete“ beziehungsweise „kSecAttrAccessible WhenUnlocked“ versehen werden. Dies bewirkt eine Verschlüsselung der Daten, welche nur mit Hilfe des Passcodes aufgehoben werden können.

Darüber hinaus gab es bei forensischen Untersuchungen eines iPhones ab iOS Version 4 eine weitere Hürde zu überwinden: Die standardmäßige AES-Verschlüsselung des internen Flashspeichers. Diese Hürde ist jedoch seit kurzer Zeit nicht mehr vorhanden, da eine Möglichkeit zum Brechen der Verschlüsselung gefunden wurde [Kata11].

Solange Apple und die Hersteller von Applikationen nicht nachbessern, ist es nicht nur für den Forensiker ein Leichtes, an wichtige Daten zu gelangen und somit ist von einem Einsatz dieses Smartphones im Business-Bereich abzuraten.

Literatur

- [BITK09] BITKOM. Mehr als vier Milliarden Handy-Nutzer weltweit (2009, August). [Online; Zugriff: 29.05.2011]. http://www.bitkom.org/de/presse/62013_60608.aspx
- [Gart10] Gartner. Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010. (2011, Februar) [Online; Zugriff: 29.05.2011]. <http://www.gartner.com/it/page.jsp?id=1543014>
- [NIST07] NIST. Guidelines on Cell Phone Forensics. (2007, May) 13 [Online; Zugriff: 29.05.2011]. <http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf>
- [Zdzi08] Jonathan Zdziarski, iPhone Forensics.: O'Reilly Media, Inc., (2008).
- [BSI09] BSI. BSI: M 6.126 Einführung in die Computer-Forensik. (2009) [Online; Zugriff: 29.05.2011]. <https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/m/m06/m06126.html>
- [BuFe08] Aaron Burghardt and Adam J. Feldman. Using the HFS+ journal for deleted file recovery. DFRWS (Digital Forensics Research Conference). (2008) [Online; Zugriff: 29.05.2011]. <http://www.dfrws.org/2008/proceedings/p76-burghardt.pdf>
- [Alla11] Alasdair Allan. Got an iPhone or 3G iPad? Apple is recording your moves. (2011, April) [Online; Zugriff: 29.05.2011]. <http://radar.oreilly.com/2011/04/apple-location-tracking.html>
- [HeBo11] Jens Heider and Matthias Boll. Lost iPhone? Lost Passwords! (2011, Februar) Fraunhofer-SIT. [Online; Zugriff: 29.05.2011]. http://www.sit.fraunhofer.de/Images/sc_iPhone%20Passwords_tcm501-80443.pdf
- [Mart08] Andrew Martin. Mobile Device Forensics, SANS Institute, (2008) [Online; Zugriff: 29.05.2011]. http://www.sans.org/reading_room/whitepapers/forensics/mobile-device-forensics_32888
- [HoSt10] Andrew Hoog, Katie Strzempka. iPhone Forensics White Paper. (2010) [Online; Zugriff: 29.05.2011]. <http://viaforensics.com/education/white-papers/iphone-forensics/>
- [LeKe10] Jeff Lessard, Gary Kessler. Android Forensics: Simplifying Cell Phone Examinations (2010) [Online; Zugriff: 29.05.2011] http://www.ssddfj.org/papers/SSDDFJ_V4_1_Lessard_Kessler.pdf
- [Kata11] Vladimir Katalov. ElcomSoft Breaks iPhone Encryption, Offers Forensic Access to File System Dumps (2011) [Online; Zugriff: 29.05.2011] <http://blog.crackpassword.com/2011/05/elcomsoft-breaks-iphone-encryption-offers-forensic-access-to-file-system-dumps/>