HOCHSCHULE
MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Herr
Vincent Duckscheidt

Reverse Engineering von E-Mail Datenbanken auf mobilen Geräten

Mittweida, Mai 2025

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Reverse Engineering von E-Mail Datenbanken auf mobilen Geräten

Autor:
Vincent Duckscheidt

Studiengang:
Allgemeine und Digitale Forensik

Seminargruppe:
FO19w1-B

Erstprüfer:
Prof. Dr. rer. nat. Michael Spranger

Zweitprüfer:
Lukas Jaeckel, M.Sc.

Einreichung:
Mittweida, 26.05.2025

Verteidigung/Bewertung:
Mittweida, 2025

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Reverse Engineering of Email Databases on Mobile Devices

Author:
Vincent Duckscheidt

Course of Study:
General and Digital Forensics

Seminar Group:
FO19w1-B

First Examiner:
Prof. Dr. rer. nat. Michael Spranger

Second Examiner:
Lukas Jaeckel, M.Sc.

Submission:
Mittweida, 26.05.2025

Defense/Evaluation:
Mittweida, 2025

Bibliografische Beschreibung

Duckscheidt, Vincent:
Reverse Engineering von E-Mail Datenbanken auf mobilen Geräten. – 2025. – 39 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2025.

Referat

In dieser Arbeit werden die Daten- und Datenbankstrukturen hinter der Mail-Applikation unter iOS und der Gmail-Applikation unter Android untersucht. Hierbei wird sich auf die Ortung und Analyse forensisch relevanter Informationen, sowie die Verknüpfung komplexerer Zusammenhänge zwischen verschiedene Dateien und Datenbanken fokussiert. Sekundär wird auch der mögliche Zugriff auf gelöschte Daten besprochen. Die hierfür nötigen Testdaten werden in mehreren Versuchsreihen mit drei Mobiltelefonen generiert. Daraufhin werden mittels forensischer Tools die Daten extrahiert und untersucht.
Es wurde festgestellt, wo forensisch relevante Daten vor und nach einer Löschung zu finden sind. Weiterhin wurde herausgefunden, in welchen komplexeren Zusammenhängen sowohl sie, als auch die ihnen zugrundeliegenden Strukturen, zueinander stehen. Allerdings waren auch viele Informationen nicht auslesbar oder aufgrund fehlender Strukturen unverständlich. Bei diesen Daten ist eine weitere, tiefer gehende Analyse nötig.

Inhaltsverzeichnis

Ergänzende Verzeichnisse

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Über die Jahre ist die digitale Welt immer relevanter geworden. So hat sich die Anzahl der Internetnutzer in Deutschland seit 1997 nicht nur mehr als versechzehnfacht, sondern es ist auch die Frequenz der Nutzung gestiegen [1]. Ebenso stieg der Anteil der mobilen Nutzer zwischen 2015 und 2021 von 54% auf 82% [2]. Mit wachsender allgemeiner Nutzung geht auch eine Nutzung durch Straftäter einher, wobei hier sowohl die Menge, als auch der technologische Fortschritt für Strafverfolgungsbehörden neue Probleme aufstellt [3]. So muss nicht mehr nur eine Telefonleitung mitgeschnitten werden, sondern es müssen verschiedenste Kommunikationsmethoden analysiert werden, die alle auf verschiedensten Kombinationen von Systemen und Infrastrukturen ausgeführt werden [3]. Trotz solcher Schwierigkeiten, hat die schiere Menge an verfügbaren Informationen, die bei der Auswertung von mobilen Geräten erlangt werden kann, einen kaum vergleichbaren Nutzen für die Strafverfolgung [4].

Ebenso wie sich die Menge der Nutzer verschoben hat, hat sich auch die Art der Nutzung geändert. Bei Befragungen zwischen 2013 und 2017 zur Nutzung von Messaging-Diensten [5] hat der Anteil der Befragten, die innerhalb der vergangenen Woche SMS genutzt haben, sich von 69% auf 38% fast halbiert. Dafür ist die Nutzung von E-Mails im gleichen Zeitraum von 37% auf 60% gestiegen. Bei Social Network Messaging hat sich die Nutzung von 26% auf 50% knapp verdoppelt und vor allem Mobile Instant Messaging Nutzung hat sich von 24% auf 73% mehr als verdreifacht, wie in Abbildung 1.1 zu sehen [5].

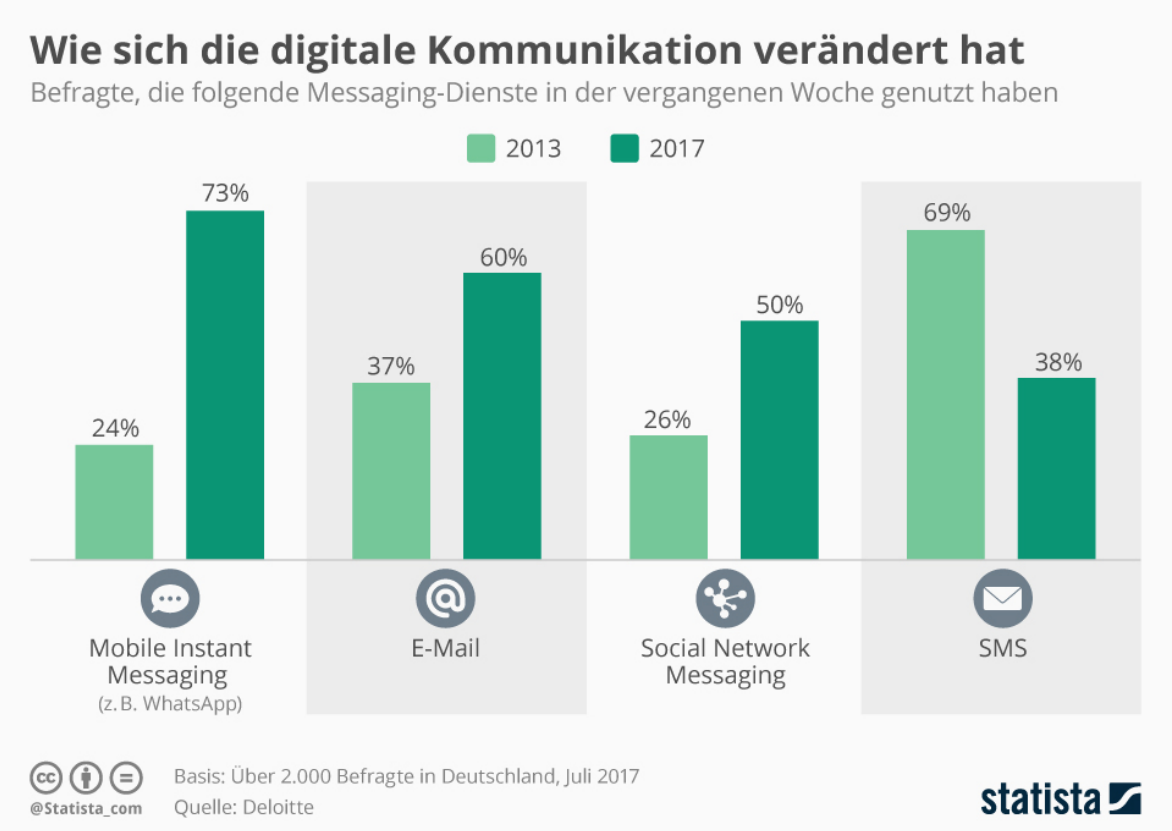

Abbildung 1.1: Balkendiagramm zum Vergleich der Ergebnisse von Befragungen zur Nutzung von Messaging-Diensten in 2013 und 2017 [5]

Geräte übergreifende Kommunikationsmethoden wie die E-Mail, werden zudem zu 43% auf mobilen Geräten genutzt, wie in Abbildung 1.2 zu sehen [6].

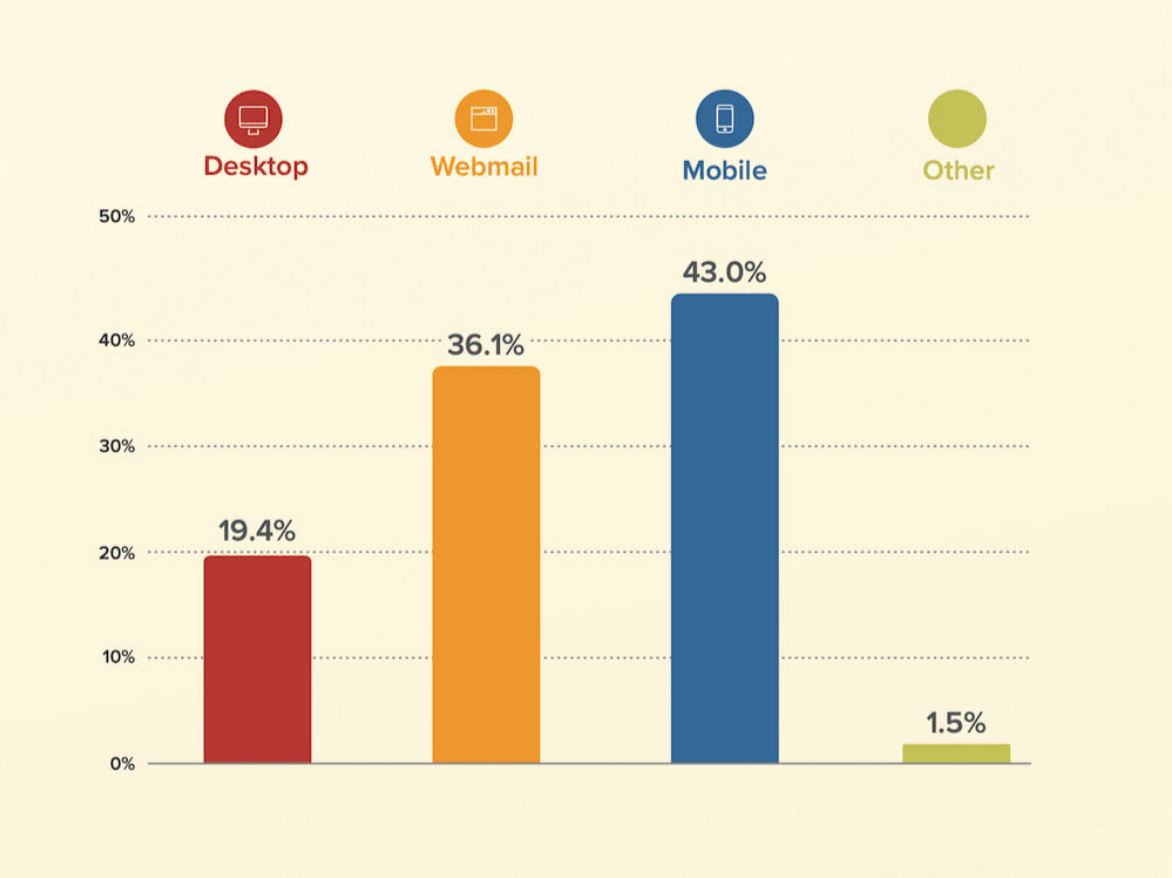

Abbildung 1.2: Balkendiagramm zum Vergleich des Anteils an geöffneten E-Mails in verschiedenen Nutzerumgebungen im ersten Quartal 2021 [6]

Aufgrund einer Vielzahl von Forschung zur Auswertung Mobiler Messanger [7–16] und der geringen Menge an vergleichbaren Arbeiten zu E-Mail-Applikationen [17] wird sich diese Arbeit auf mobile E-Mail-Applikationen fokussieren, da die Verteilung der wissenschaftlichen Arbeiten eine starke Diskrepanz zur Verteilung der Nutzer aufweist. Um sowohl möglichst viele potenzielle Nutzer abzudecken, als auch gleichzeitig den Umfang der Arbeit einzugrenzen, wird sich auf vorinstallierte Applikationen beschränkt. Die meist verbreitetsten Betriebssysteme für mobile Geräte sind Android mit 65,45% und iOS mit 33,59% Marktanteil, wie in Abbildung 1.3 zu erkennen ist [18]. Vorinstalliert sind jeweils die Gmail-Applikation bei Android Geräten [19], weiterführend als Gmail bezeichnet. Und die Mail-Applikation bei iOS Geräten [20], weiterführend als Mail bezeichnet.

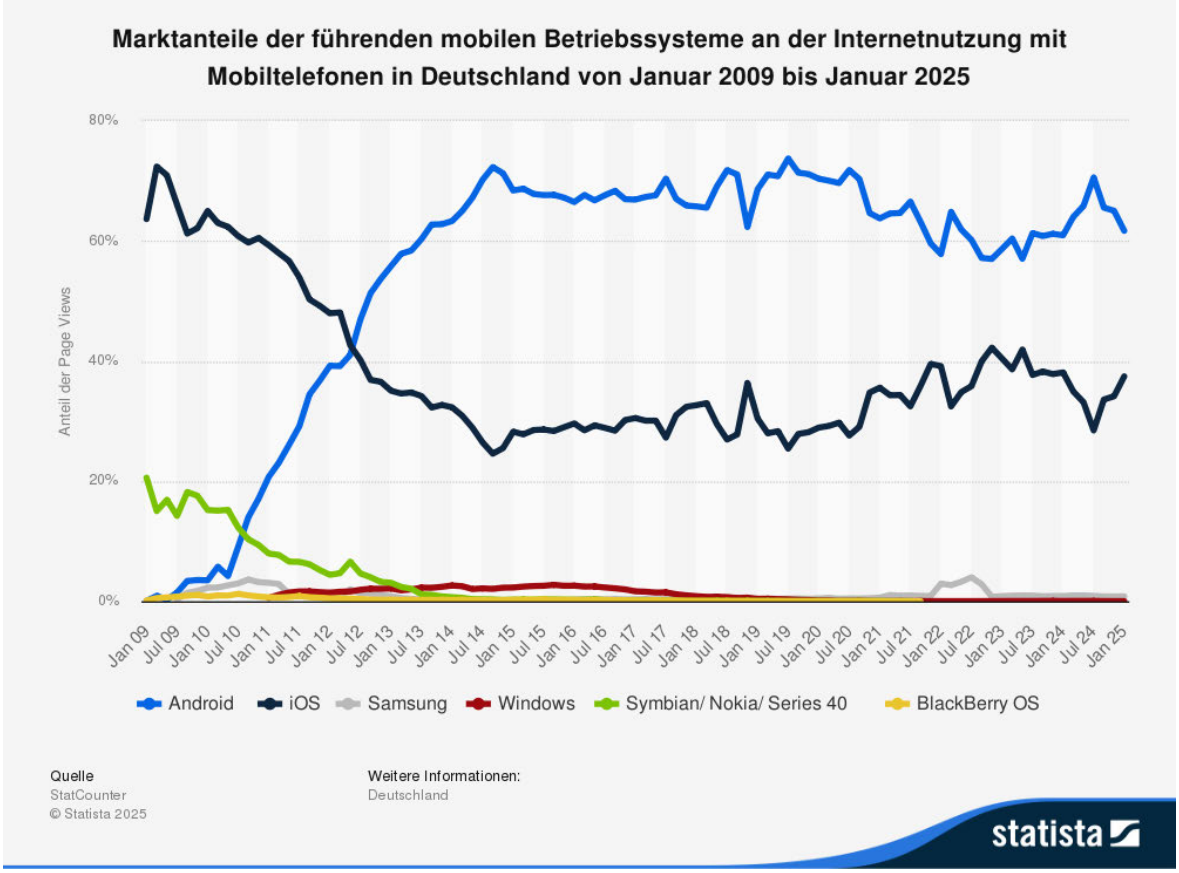

Abbildung 1.3: Diagramm der Marktanteile der führenden mobilen Betriebssysteme an der Internetnutzung mit Mobiltelefonen in Deutschland von Januar 2009 bis Januar 2025 [18]

1.1 Aufbau der Arbeit

In Kapitel 2 wird der allgemeine Forschungsstand der IT-Forensik bzgl. mobiler Endgeräte und spezifisch mobiler Messenger und E-Mail-Applikationen diskutiert. Durch die Zusammenhänge und Unterschiede zu vorangegangenen Arbeiten wird die Forschungslücke etabliert, die diese Arbeit füllen soll. Daraufhin werden in Kapitel 3 die Grundlagen etabliert, die in Kapitel 4 zum generieren der Testdaten verwendet werden und zum Verstehen der Ergebnisse in Kapitel 5 nötig sind. In Kapitel 5 wird zuerst die grundsätzliche Ordnung der Daten unter beiden Applikationen beschrieben. Danach wird detaillierter auf die Datenbanken, Dateien und deren Verbindungen eingegangen, soweit es sich aus den generierten Daten schließen lies. In Kapitel 6 werden dann die Ergebnisse zusammengefasst und wie diese und eventuelle Lücken in weiteren Arbeiten genutzt werden können.

2 Verwandte Arbeiten

Im Rahmen der Literaturrecherche wurden vor allem Paper und Artikel gefunden, die sich mit der Analysen von verschiedensten Messengern beschäftigten. Diese weichen zwar meistens in Thema und Fokus stark vom Ziel dieser Arbeit ab, können aber neben der Etablierung des allgemeinen Forschungsstandes weiter als Grundlage dienen. Durch eine Analyse von ein- und ausgehenden Daten verschiedener Datenbanken unter Android ist bereits bekannt, dass die Datenbanken der Android Gmail Version teils auf SQLite basieren [21]. Weitere Arbeiten, die sich mit der Analyse solcher Datenbanken befassen, sprechen von der Verbreitung von SQLite unter mobilen Applikationen [22, 23]. Da die folgenden Messenger teils ebenfalls SQLite benutzen, lassen sich durch die Ähnlichkeiten in mobilen Datenstrukturen und ihrer Extraktionsmethoden Parallelen ziehen.

Husain und Sridhar [7] beschäftigen sich mit der Wiederherstellung und Analyse von Messenger Daten unter iOS anhand von drei, zum Zeitpunkt des Verfassens, beliebter Messenger: AOL Instant Messenger, Yahoo! Messenger und Google Talk. Der Fokus liegt hier auf der Extraktion der Daten ohne Jailbraking oder andere Veränderung der Daten, um die Verwertbarkeit vor Gericht zu gewährleisten. Zu diesem Zweck und mangels der Tools um eine physische Kopie zu erstellen, wurde hier zur Extraktion über iTunes ein Backup des iPhones erstellt und die Daten mittels entsprechender Software analysiert [7]. In einer weiteren früheren Arbeit fokussieren sich Mahajan u. a. [8] hier auf die Extraktion von Daten aus WhatsApp und Viber für Android. Hier wird vorwiegend geklärt, welche Daten überhaupt lokal gespeichert werden, wo sie liegen, und was für Informationen sich entnehmen lassen. Hierzu wurden mittels fünf verschiedener Mobiltelefone mit drei verschiedener Android Betriebssysteme Daten erzeugt und physisch extrahiert [8]. Eine andere Arbeit geht auf mögliche Sicherheitslücken und Methoden zur Extraktion unter WhatsApp für Android ein und versucht allgemein anwendbare Methoden der Analyse anderer Messenger zu entwickeln [9].
Trotz der thematischen Distanz und dem unterschiedlichen Fokus auf Extraktion, bilden diese Arbeiten einen wichtigen Grundstein für die Analyse von extrahierten Artefakten.

In der Analyse von ChatSecure unter Android von Anglano u. a. [10] liegt der Fokus eher auf der Entschlüsselung der Datenbänke, teils durch das Entnehmen des Passwortes aus dem flüchtigen Speicher. Um die Experimente möglichst generell, kontrolliert und reproduzierbar zu halten, wurden hier verschiedene Systeme emuliert und deren Ergebnisse mit denen echter Mobiltelefone verglichen. Diese Herangehensweise würde allerdings, durch die Vervielfachung an Betriebssystemen mittels Emulation und dem damit einhergehenden exponentiellen Anstieg an Szenarien und zu analysierenden Daten, den Rahmen dieser Arbeit sprengen [10].
Neben der Extraktion und Entschlüsselung durch verschiedene Tools wurde auch der flüchtige Speicher der emulierten Geräte extrahiert und nach Passwörtern durchsucht. Eine ähnliche Analyse echter Geräte ist nur durch Root-Zugriff und je nach Tool einer Neukonfigurierung des Kernels möglich, womit sich diese Herangehensweise eher schlecht für diese Arbeit eignet [10].

Zwar ist hier ein Wiederherstellen der Nutzerkonten inklusive Logs durch die Analyse verschiedener Datenbanken gelungen, der Nutzen liegt aber primär in der Entschlüsselung von SQLite Datenbanken bzw. SQLCipher, sollte dies relevant werden [10].

Zhang u. a. [11] untersuchen in einer Analyse von WeChat sowohl die Datenbanken und Speicherorte, als auch die Ver- und Entschlüsselung und die Wiederherstellung von gelöschten Daten. Hierbei wurden der File Header und weitere Teile der Datenbank analysiert, um ein Verständnis für die allgemeine Struktur der Datenbank zu erlangen. Durch eine Analyse des SQLCipher Quellcodes ist es gelungen, die Verschlüsselungsmethode und die Methode zur Erstellung eines Schlüssels nachzuvollziehen. Zuletzt wurde bei der Untersuchung gelöschter Daten festgestellt, das hier nur der Page Header gelöscht wurde, wodurch die restlichen Daten als überschreibbar markiert wurden. Daher ist es möglich, gelöschte Daten einfach auszulesen, sollten diese noch nicht wieder überschrieben worden sein. Da dies ein Mechanismus von SQLite ist, sollte dies auch bei anderen Applikationen möglich sein, vorausgesetzt es gab keine weitere Anpassung von SQLite. [11].
In einer weiteren Analyse von WeChat wurde mit verschiedenen Android Mobiltelefonen und Emulatoren experimentiert. Hier wurden teils durch direkte Extraktion, teils über Backups die Konversationen und versandten Daten entnommen und entschlüsselt [12]. Eine Analyse von Telegram auf Android Geräten fokussiert sich auf das Rückverfolgen und Wiederherstellen von Daten und war in der Lage innerhalb der Datenbanken die Speicherorte systemrelevanter Logdateien zu bestimmen [13]. Hierbei sind vor allem die Verbindungen zwischen den Datenbanken und den Logdateien relevant. Durch Logs von Installationen oder der Erstellung von Dateien werden Relationen und Speicherorte klar, die nicht durch Daten zu Sender, Inhalt oder sonstigem klarwerden.

Anglano [14] hat komplexere Strukturen im Android Messanger WhatsApp wiederhergestellt. Dafür wurde in verschiedenen Szenarien Daten auf emulierten Android Geräten erstellt. Hier wurde dann über die Datenbank Dateien Verbindungen zwischen den Rohdaten und Logdateien gezogen. Somit wurde die Chronologie von Nachrichten und Kontaktlisten wiederhergestellt [14]. In einer darauf aufbauenden Arbeit, diesmal beim Android Messanger Telegram, werden von Anglano u. a. [15] solche komplexen Analysen durch die Analyse des Quellcodes erweitert, um weitere Verbindungen der Daten herzustellen. Hierbei werden sowohl mit emulierten Geräten als auch mit physischen Mobiltelefonen Daten erzeugt. Die Analyse des Quellcodes hat weiterhin zur Identifikation verschiedenster Dateneigenschaften geführt. Es war möglich, den Inhalt und die Chronologie von Nachrichten und der Kontaktliste wiederherzustellen. Es war sogar möglich, die daran vorgenommenen Veränderungen und den Unterschied zwischen manuell hinzugefügten Kontakten und solcher, die aus dem Telefonbuch des Geräts importiert wurden, festzustellen [15]. Bei einer Analyse des Telegram Messengers auf einem Windows Phone wurde von Gregorio u. a. [16] neben der Recherche zur Applikation und der Analyse der Artefakte auch auf die Analyse des Quellcodes gesetzt. Vor allem konnte hier durch, aus dem Quellcode erlangten Informationen, bessere Resultate erzielt werden, als bei vergleichbaren Analysen. Da Telegram im Vergleich zu anderen Messengern Open Source ist, ist diese Herangehensweise allerdings schwer zu verallgemeinern und setzt je nach Applikation ein Reverse Engineering des Quellcodes voraus [16]. Dennoch können auch ohne vollständige Analyse des Quellcodes Schlüsse auf die Arbeitsweise der Applikation gezogen werden und somit auf zugrunde liegende Strukturen, die im Rahmen dieser Arbeit untersucht werden.

Neben weiteren Beispielen zur Analyse und Verbindung der Daten, liegen bei diesen Arbeiten [14–16] vor allem die Motivation dieser Bachelorarbeit. Durch eine Analyse nicht nur der Daten, sondern auch der zugrundeliegenden Strukturen, ist es möglich, komplexere Zusammenhänge zu verstehen und verstreute Daten miteinander zu verbinden.

Zur mobilen E-Mail Forensik ist die Forschungslage eher dünn. Eine Arbeit fokussiert sich hier auf die Analyse von flüchtigem Speicher anhand der Android E-Mail-Applikationen MailMaster und QQMail [17]. Hierbei wird, wie schon in anderen Arbeiten [10], besprochen Daten aus dem flüchtigen Speicher vom gerooteten Mobiltelefonen extrahiert und analysiert. Da neben Passwörtern auch andere Daten extrahiert wurden, die andere Arbeiten auch im nichtflüchtigem Speicher gefunden haben, bietet sich hier eine alternative Herangehensweise zur Extraktion [17]. Analysen von E-Mail Datenbanken sind eher für Desktop Applikationen zu finden. Eine Analyse von Windows Mail auf Windows 10 fokussiert sich stark auf die zugrundeliegenden Strukturen wie Tags der Dateitypen in hexadezimaler Form , Analyse der generierten Dateinamen oder die Verbindungen zwischen den einzelnen Tabellen [24]. Durch die Datenextraktion der Firma Forensafe Software Solutions mit ihrem Tool ArtiFast [25] ist bekannt, dass Gmail für Android teils ähnliche Strukturen verwendet wie Windows Mail in der o.g. Arbeit [24], wie z.B. Verknüpfung verschiedener Tabellen via `ROWIDs`. Da solche, zwar simple Strukturen, sowohl firmen- als auch geräteübergreifend vorhanden sind, ist nicht auszuschließen, diese auch in anderen Clients wiederzufinden. Die hier [24] vorgenommene Analyse, dient als primärer Leitfaden dieser Arbeit. Das vollständige Verständnis der Daten, auch solcher die nicht in Klartext vorliegen, durch Rückverfolgung der Tags zu erzeugten Testdaten. Die interne Namenszuweisung zur Erkennung versandter und empfangener Dateien nachvollziehen. Das Verstehen und Aufzeigen der kreuzreferenziellen Verbindungen zwischen den Datenbanken. Und der dadurch wiederhergestellte Zusammenhang aller separat gespeicherten Daten. Angelehnt an [24] sollen diese Ziele erfüllt werden.

Zusammengefasst soll diese Arbeit die Lücke in der Analyse Mobiler E-Mail-Applikationen schließen und in Anlehnung an ähnliche Arbeiten zu Desktop Clients [24], die zugrundeliegende Struktur der Datenbanken rekonstruieren. Dies soll eine Basis für zukünftige Analysen von komplexeren Strukturen bieten da, wie in einigen o.g. Arbeiten gesehen, ein Verständnis von grundsätzlichen Strukturen der Datenbanken [14, 15] und wenn möglich, des Quellcodes [10, 11, 16] zum leichteren und oder besseren Verbinden der extrahierten Informationen führt.

3 Grundlagen

3.1 Einleitung in die IT-Forensik

Laut dem Bundesamt für Sicherheit in der Informationstechnik [26] teilt sich die IT-Forensik allgemein in Online-Forensik und Offline-Forensik auf. Eine Untersuchung der Online- oder auch Live-Forensik fängt schon während eines Vorfalls an um flüchtige Daten abzufangen. Dazu Zählen Informationen zu aktuellen Prozessen und Netzwerkverbindungen sowie Inhalte des Hauptspeichers. Die in dieser Arbeit vorgenommene Analyse ist Teil der Offline-Forensik, also eine sogenannte Post-mortem-Analyse. Hierbei werden mittels Abbilder der Datenträger Informationen gesucht, die auf eine weise obfuskiert wurden; also verschlüsselte, versteckte, gelöschte oder einfach umbenannte Dateien [26]. Neben der Wiederherstellung von gelöschten Daten wird sich diese Arbeit hauptsächlich mit der Analyse von Daten und Datenverbindungen beschäftigen, die von sich aus verstreut und oder schwer nachzuvollziehen sind.

Aufgrund des Fokus auf mehrere mobile E-Mail-Applikationen fällt diese Arbeit außerdem unter die Teilgebiete der Mobilfunkforensik, iOS-Forensik und Android-Forensik. Die Mobilfunkforensik beschäftigt sich mit der Sicherung und Auswertung von Mobiltelefonen während sich die iOS-Forensik und die Android-Forensik jeweils auf die ihnen namensgebenden Betriebssysteme fokussieren [27].

3.2 Mail

Mail wird seit 2001 von Apple kostenlos unter ihren Betriebssystemen vertrieben, ursprünglich unter MacOS später mit dem Erscheinen des iPhones auch unter iOS bzw. damals Mac OS X [28]. Über die Jahre hat die Applikation viele neue Funktionen erhalten, einige für die Versuchsdurchführung relevante Funktionen werden im folgenden erklärt. Wie im durch Apple beschrieben [29] können sowohl E-Mails als auch Konversationen angezeigt und als gelesen markiert werden. E-Mails können neben den Adressaten mittels des CC weitere Kopien versenden bzw. diese mit dem BCC als Blindkopie versenden, also ohne das andere Adressaten dies sehen. Die E-Mails können auch in einem einstellbaren Zeitrahmen verzögert versendet werden oder vorläufig nur als Entwurf gespeichert werden. E-Mails können außerdem Dateien angehangen werden und auf E-Mails kann geantwortet werden oder sie können weitergeleitet werden. Die E-Mails können auf verschiedene weisen markiert werden, teils nur farblich zur Organisation, teils funktional, um z.B. Mitteilungen an oder aus zu schalten. Des weiteren können E-Mails in verschiedene voreingestellte oder eigens erstellbare Postfächer verschoben werden. Beispielsweise können E-Mails archiviert oder gelöscht werden, wobei sie dabei nicht tatsächlich gelöscht werden, sondern nur in den Papierkorb verschoben werden. Von dort aus können sie entweder wieder hergestellt werden oder endgültig gelöscht werden [29].

In Kapitel 5.6 werden IMAP-URLs innerhalb von Einstellungsdateien besprochen, die sich auf die Postfächer von Mail beziehen. IMAP oder Internet Message Access Protocol bezeichnet eine Vorgehensweise auf E-Mails zuzugreifen, bei der die E-Mail beim E-Mail-Dienst gespeichert wird und nur beim tatsächlichen Öffnen heruntergeladen wird [30]. Die URL kann je nach Spezifikation der Abfrage variieren; im vorliegenden Fall bezieht sich sich nur auf das Postfach, daher setzt sie sich wie folgt zusammen: ”imap://[ServerID]/[PosteingangsName]” [31].

3.3 Gmail

Gmail ist eine E-Mail-Applikation die von Google seit 2004 kostenlos vertrieben wird und sowohl ursprünglich im Browser oder auf dem Desktop nutzbar war, als auch über die Jahre als mobile Applikation verfügbar wurde [32]. Einige ihrer Funktionen, die für die Versuchsdurchführung in Kapitel 4 relevant werden, werden im folgenden näher erläutert. Wie Jonnalagadda [33] erklärt kann in Gmail wie bei E-Mails üblich beim versenden von E-Mails neben den Adressaten mittels des CC Kopien versenden bzw. diese mit dem BCC als Blindkopien versenden. Neben dem Betreff der E-Mail und natürlich dem Text ist es weiterhin möglich, verschiedene Dateien als Anhänge mit zu versenden. Es können außerdem E-Mails in verschiedene Postfächer und Labels kategorisiert werden. Sie können z.B. archiviert oder gelöscht werden. Letzteres löscht die E-Mails nicht wirklich, sondern verschiebt sie nur in den Papierkorb. Von dort können sie entweder wiederhergestellt werden oder sie können endgültig gelöscht werden. Empfangene E-Mails können z.B. zurückgestellt, sodass sie zu einem späten Zeitpunkt wieder im Posteingang auftauchen. Oder es werden E-Mails auf verschiedenste weise markiert. So können sie z.B. als wichtig markiert werden oder stumm geschaltet werden, sodass weitere E-Mails innerhalb einer Konversation keine Benachrichtigungen mehr erzeugen [33].

In Kapitel 5.3 wird besprochen wie Gmail `BLOBs` nutzt, um mehrere Einträge teils unterschiedlichen Datentyps in eine Tabellenspalte zu schreiben. Ein `BLOB` oder Binary Large Object bezeichnet in erster Linie eine Binärdatei mit einer Menge an unstrukturiertem Inhalt [34]. Dabei ist der genaue Inhalt egal; ob Zahlen, Text oder komprimierte Dateien, der Inhalt lässt sich schlecht in anderen, strukturierten Formaten speichern [34].

3.4 SQLite

Da mindestens Gmail SQLite [21] nutzt wird dieses im folgenden erklärt. SQLite ist eine Programmbibliothek mit eingebetteter SQL-Datenbank-Engine, das sowohl Open Source, als auch gemeinfrei und mit verschiedensten Programmiersprachen kompatibel ist [35]. Außerdem funktioniert es auf mehreren, auch eingebetteten, Betriebssystemen und in Kombination mit der kompakten Größe und hohen Zuverlässigkeit ist es daher sehr verbreitet auf mobilen Geräten [35].
Besonders wichtig, vor allem für die Untersuchung gelöschter Daten, ist der ”Write-Ahead Log” oder ”WAL”. Das Write-Ahead Logging ist ein optionaler Modus von SQLite in dem nicht direkt auf die Datenbank sondern auf eine separate Datei geschrieben wird, diese treten jeweils zusammen auf und enden auf `.db` und `.db-wal` [36]. Auf der WAL-Datei werden alle

Änderungen gesammelt, bis eine vorher festgelegte Menge an Einträgen erreicht ist. Beim erreichen dieses Checkpoints werden alle Einträge der Wal-Datei gleichzeitig auf die Datenbank überschrieben [37]. Da nicht jeder Eintrag direkt ausgefüllt wird finden sich beim separaten Betrachten auf der WAL-Datei alle neueren Einträge. Aber vor allem finden sich auf beiden Dateien eigentlich schon gelöschte Einträge, da auch der Befehl zum Löschen erst ausgeführt wird, wenn ein Checkpoint erreicht wird.
`ROWID` Tabellen sind eine bestimmte Art von Tabellen unter SQLite, in der jede Reihe, einer dafür designierten Spalte, einen für diese Tabelle einzigartigen Integer bekommt, um die Einträge einfacher suchen oder sortieren zu können [38]. Dies ist der meist verbreitetste Weg, auf Einträge in anderen Tabellen zu referenzieren. Daher wird bei der Diskussion der Verbindungen zwischen verschiedenen Tabellen und Dateien in Kapitel 5 sehr häufig über verschiedene `ROWID`s gesprochen.

4 Versuchsdurchführung

4.1 Versuchsaufbau und verwendete Tools

Die zur Analyse benötigten Testdaten werden zwischen drei Mobiltelefon erstellt, um möglichst viele Szenarien abdecken zu können. Hierbei werden Gmail auf Android und Mail auf iOS abgedeckt, da diese auf den Betriebssystemen mit dem größten Marktanteil vorinstalliert sind. Es liegen folgende Geräte vor: ein Galaxy S9 unter Android 10 mit Gmail der Version 2023.11.26.586591930.Release, ein iPhone 7 unter iOS 15.8.3 wo Mail am 28.11.2024 heruntergeladen wurde und ein Blackview A80Pro unter Android 10 mit Gmail der Version 2024.11.03.695901069.Release. Das Blackview A80Pro dient nur zur Abdeckung der Szenarien. Alle Daten werden auf dem Galaxy S9 und dem iPhone 7 erzeugt und von diesen extrahiert.

Sämtliche Daten der Clients werden vor Beginn der Experimente gelöscht, Mail wird deinstalliert und neu installiert, Gmail nicht, da dies Root-Zugriff voraussetzt. Bei Mail wird eine Einstellung zum Schutz von E-Mail Aktivitäten nicht aktiviert, diese wird beim ersten Öffnen zwangsweise abgefragt. Alle Postfächer werden auf sichtbar gestellt und ein weiteres Postfach ”Neues Postfach” wird hinzugefügt. Weitere Einstellungen werden nicht verändert. Bei Gmail werden unter Einstellungen>[Kontoname]>Posteingangskategorie alle möglichen Postfächer aktiviert: Werbung, Soziale Netzwerke, Benachrichtigungen und Foren. Weitere Einstellungen werden nicht verändert.

Die ausgeführte Testvariante wird innerhalb der E-Mail zur leichteren Identifizierung vermerkt, so ist z.B. der Name der PDF-Datei ”PDF_Datei.pdf”, der Betreff der E-Mail ist ”PDF Datei Betreff”, der Text ”PDF Datei Text”. Wenn mehrere Varianten zwischen den Geräten versandt werden, wird die Nummer der Variante in Betreff und Text notiert. Des weiteren wird der Versandzeitpunkt protokolliert.

Nach dem Generieren von Testdaten in den folgenden Versuchsreihen werden mithilfe der Sicherungssoftware Cellebrite UFED in der Version 7.66.1.150 die Daten des Galaxy S9 und des iPhone 7 gesichert [39]. Zusätzlich wurde der Device Adapter 3 des Cellebrite UFED als Schreibblocker verwendet. Die Daten zu Mail und Gmail wurden darauf mit dem Cellebrite Inseyets Physical Analyzer in der Version 10.3.0.3169 extrahiert [40, 41].

Nach der ersten Extraktion werden alle E-Mails auf dem Galaxy S9 und dem iPhone 7 endgültig gelöscht, indem sie aus dem Papierkorb entfernt werden. Danach werden die Mobiltelefone wieder gesichert und die Daten Extrahiert. Hiermit soll ein Vergleichspunkt geschaffen werden, um herauszufinden welche der ersten generierten Daten unmittelbar nach dem Löschen noch zu rekonstruieren sind. Und welche nach dem generieren weiterer Daten noch nicht überschrieben wurden.

Die extrahierten Daten wurden nach bzw. zwischen den Versuchsreihen mit verschiedenen Programmen analysiert. Zur Sichtung der Datenbänke wurde der DB Browser für SQLite für 64-Bit Windows in der Version 3.13.1 verwendet [42]. Zur Sichtung der `.db-wal` Dateien

wurde der Hexeditor HxD in der Version 2.5.0.0 verwendet [43]. Zur besseren Sichtung der `.emlx` Dateien wurde der Kernel EML Viewer von Kernel Data Recovery in der Version 11.05.01 verwendet [44].

Die folgenden Versuchsreihen werden sowohl schriftlich verfasst, um sie zu begründen, als auch tabellarisch, damit sie möglichst einfach zu reproduzieren sind.

4.2 Erste Versuchsreihe

Die ersten E-Mails werden mit den im Client integrierten Funktionen zwischen S9 und iPhone 7 verschickt. Am Anfang werden alle E-Mails von beiden Geräten verschickt, später werden einseitig vorhandene Funktionen verwendet. Zuerst wurden unveränderte E-Mails verschickt, danach wurden Entwürfe gespeichert und verworfen, empfangene Mails als gelesen und Spam markieren und archiviert, gelöscht und es wurde auf E-Mails geantwortet. Diese Szenarien werden in Tabelle 4.1 aufgeführt.

Versuch	Sender	Empfänger	Antwort
Unveränderte E-Mail	S9	iPhone 7	
	iPhone 7	S9	
Entwurf Speichern	S9	iPhone 7	
	iPhone 7	S9	
Entwurf verwerfen	S9	iPhone 7	
	iPhone 7	S9	
Als gelesen markieren	S9	iPhone 7	
	iPhone 7	S9	
Als Spam markieren	S9	iPhone 7	
	iPhone 7	S9	
Archivieren	S9	iPhone 7	
	iPhone 7	S9	
Antworten	S9	iPhone 7	iPhone 7
	iPhone 7	S9	S9

Tabelle 4.1: Szenarien der ersten Versuchsreihe: genutzte Funktionen, die sowohl unter Android als auch iOS vorhandenen sind

Unter Android wurden E-Mails in die folgenden Postfächer verschoben: Werbung, Soziale Netzwerke, Benachrichtigungen, Foren. E-Mails wurden nach Plan und im Modus Vertraulich verschickt. Empfangene E-Mails wurden zurückgestellt und das Label Posteingang entfernt. Sie wurden als wichtig und mit einem Stern markiert oder ignoriert und es wurde mit einem Emoji reagiert. Diese Szenarien werden in Tabelle 4.2 aufgeführt.

Versuch	Sender	Empfänger	Antwort
Verschieben: Werbung	iPhone 7	S9	
Verschieben: Soziale Netzwerke	iPhone 7	S9	
Verschieben: Benachrichtigungen	iPhone 7	S9	
Verschieben: Foren	iPhone 7	S9	
Nach Plan	S9	iPhone 7	
Modus Vertraulich	S9	iPhone 7	
	A80	S9	
Zurückstellen	iPhone 7	S9	
Label Posteingang entfernen	iPhone 7	S9	
Als wichtig markieren	iPhone 7	S9	
Mit Stern markieren	iPhone 7	S9	
Ignorieren	iPhone 7	S9	
Mit Emoji reagieren	S9	A80	A80
	A80	S9	S9

Tabelle 4.2: Szenarien der ersten Versuchsreihe: genutzte Funktionen, die nur unter Android vorhanden sind

Unter iOS wurden E-Mails in die folgenden Postfächer verschoben: Gesendet, Spam, Neues Postfach, Archiv. Es wurde eine Zeichnung versandt und bei eingegangenen E-Mails Mitteilungen an und stumm gestellt. Weitere E-Mails wurden in folgenden Farben Markiert: rot, grün, orange, blau, lila, gelb, grau. Diese Szenarien werden in Tabelle 4.3 aufgeführt.

Versuch	Sender	Empfänger
Verschieben: gesendet	S9	iPhone 7
Verschieben: Spam	S9	iPhone 7
Verschieben: neues Postfach	S9	iPhone 7
Verschieben: Archiv	S9	iPhone 7
Zeichnung	iPhone 7	S9
Mitteilungen an	S9	iPhone 7
Mitteilungen stumm	S9	iPhone 7
Markieren: rot	S9	iPhone 7
Markieren: grün	S9	iPhone 7
Markieren: orange	S9	iPhone 7
Markieren: gelb	S9	iPhone 7
Markieren: blau	S9	iPhone 7
Markieren: lila	S9	iPhone 7
Markieren: grau	S9	iPhone 7

Tabelle 4.3: Szenarien der ersten Versuchsreihe: genutzte Funktionen, die nur unter iOS vorhanden sind

Danach werden Anhänge in den beliebtesten Dateiformaten ihrer Kategorie verschickt [45]: .DOCX, .PDF, .JPG, .PNG, .MP3, .WAV, .MP4, .MOV, .EXE, .APK, .ZIP, .RAR, .CSV und .SQL. Hierfür wurden Beispieldateien von einschlägigen Seiten für Testdaten genommen [46–49]. Diese Szenarien werden in Tabelle 4.4 aufgeführt.

Versuch	Sender	Empfänger	Versuch	Sender	Empfänger
DOCX	S9	iPhone 7	MOV	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
PDF	S9	iPhone 7	EXE	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
JPG	S9	iPhone 7	APK	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
PNG	S9	iPhone 7	ZIP	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
MP3	S9	iPhone 7	RAR	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
WAV	S9	iPhone 7	CVS	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9
MP4	S9	iPhone 7	SQL	S9	iPhone 7
	iPhone 7	S9		iPhone 7	S9

Tabelle 4.4: Szenarien der ersten Versuchsreihe: Arten von Anhängen, die zwischen den Mobiltelefonen versandt wurden

Zuletzt werden Szenarien mit drei Geräten abgedeckt. Hierbei wurden die Szenarien so wiederholt, dass beide auszuwertenden Geräte an allen möglichen Positionen der jeweiligen Konstellation anzutreffen sind. Zuerst wurden E-Mails mit zwei Adressaten versandt, weitere A-Mails wurden weitergeleitet. Diese Szenarien werden in Tabelle 4.5 aufgeführt. Es wurden E-Mails mit Adressat und CC versandt, dies wird drei mal durchgeführt: beim ersten Mal wird nicht geantwortet, beim zweiten mal antwortet das Gerät im CC und beim dritten mal antwortet es allen. Diese Szenarien werden in Tabelle 4.6 aufgeführt. Bei E-Mails mit Adressat und BCC wird ebenso vorgegangen, das antwortende Gerät ist hierbei im BCC. Diese Szenarien werden in Tabelle 4.7 aufgeführt.

Versuch	Sender	erster Empfänger	zweiter Empfänger
Zwei Adressaten	S9	iPhone 7	A80
	iPhone 7	S9	A80
Weiterleiten	A80	S9	iPhone 7
	A80	iPhone 7	S9

Tabelle 4.5: Szenarien der ersten Versuchsreihe: Dreierkonstellationen zum Testen der mehrerer Adressaten und des Weiterleitens

Versuch	Sender	Empfänger	CC	Antwort
CC 1	S9	iPhone 7	A80	
CC 2	A80	S9	iPhone 7	
CC 3	iPhone 7	A80	S9	
CC antworten 1	S9	A80	iPhone 7	iPhone 7
CC antworten 2	iPhone 7	A80	S9	S9
Versuch	**Sender**	**Empfänger**	**CC**	**Antwort an alle**
CC allen antworten 1	S9	iPhone 7	A80	A80
CC allen antworten 2	A80	S9	iPhone 7	iPhone 7
CC allen antworten 3	iPhone 7	A80	S9	S9

Tabelle 4.6: Szenarien der ersten Versuchsreihe: alle Konstellationen zum Testen des CC

Versuch	Sender	Empfänger	CC	BCC	Antwort
BCC 1	S9	iPhone 7	A80	A80	
BCC 2	A80	S9	A80	iPhone 7	
BCC 3	iPhone 7	A80	A80	S9	
BCC antworten 1	S9	A80	A80	iPhone 7	iPhone 7
BCC antworten 2	A80	A80	A80	S9	S9
Versuch	**Sender**	**Empfänger**	**CC**	**BCC**	**Antwort an alle**
BCC allen antworten 1	S9	iPhone 7	A80	A80	A80
BCC allen antworten 2	A80	S9	A80	iPhone 7	iPhone 7
BCC allen antworten 3	iPhone 7	A80	A80	S9	S9

Tabelle 4.7: Szenarien der ersten Versuchsreihe: alle Konstellationen zum Testen des BCC

4.3 Zweite Versuchsreihe

In einer zweiten Versuchsreihe wurden vor allem Variationen der ersten Versuchsreihe vorgenommen, um Fragen und Unklarheiten nach der ersten Auswertung zu beantworten. Um zu klären, wann und ob Daten von E-Mails im Modus Vertraulich gespeichert werden, wurden mehrere dieser an das S9 und das iPhone 7 verschickt. Die ersten wurden nicht geöffnet, die zweiten schon. Weitere wurden weitergeleitet und es wurde, da dies keine integrierte Funktion von iOS ist, mit dem iPhone 7 der enthaltene Link geöffnet. Zum Überprüfen des Speichervorgangs der Anhänge, wurden erst E-Mails mit mehreren Anhängen versandt. Weitere E-Mails mit Anhängen wurden ungeöffnet gelassen, geöffnet; es wurde die Vorschau angeklickt und die Anhänge wurden heruntergeladen. Die gleichen Varianten wurden mit APKs als Anhang durchgeführt, um Interaktionen mit den blockierten E-Mails zu testen. Hierbei wird mit der automatischen Antwort interagiert. Diese Szenarien werden in Tabelle 4.8 aufgeführt.

Versuch	Sender	erster Empfänger	zweiter Empfänger
Vertraulich: ungeöffnet	A80	S9	
	A80	iPhone 7	
Vertraulich: geöffnet	A80	S9	
	A80	iPhone 7	
Vertraulich: Link öffnen	A80	iPhone 7	
Vertraulich: weiterleiten	A80	S9	iPhone 7
	A80	iPhone 7	S9
Mehrere Anhänge	A80	S9	
	A80	iPhone 7	
Anhänge ungeöffnet	A80	S9	
	A80	iPhone 7	
Anhänge geöffnet	A80	S9	
	A80	iPhone 7	
Anhänge Vorschau	A80	S9	
	A80	iPhone 7	
Anhänge download	A80	S9	
	A80	iPhone 7	
APK ungeöffnet	iPhone 7	S9	
APK geöffnet	iPhone 7	S9	
APK Vorschau	iPhone 7	S9	
APK download	iPhone 7	S9	

Tabelle 4.8: Szenarien der zweiten Versuchsreihe: Variationen der vorherigen Szenarien zu vertrauliche E-Mails und Anhängen zum klären des Speicherpunktes

Um zu testen, wie genau das Antworten und Weiterleiten notiert wird, wurden in mehreren Varianten abwechselnd und wiederholend auf E-Mails geantwortet oder sie wurden weitergeleitet. Die jeweils letzte E-Mail wurde nicht geöffnet, um die Aufzeichnung der Korrespondenzen zu kontrollieren. In der folgenden Tabelle 4.9 wird bei der Versuchsbenennung Antwort zu Re und weiterleiten zu Fwd gekürzt.

Versuch	Sender	erster Empfänger	zweiter Empfänger	dritter Empfänger	vierter Empfänger
Re Re	S9	iPhone 7	S9	iPhone 7	
	iPhone 7	S9	iPhone 7	S9	
Re Fwd	S9	iPhone 7	S9	iPhone 7	
	iPhone 7	S9	iPhone 7	S9	
Fwd Fwd	S9	iPhone 7	S9	iPhone 7	
	iPhone 7	S9	iPhone 7	S9	
Fwd Re	S9	iPhone 7	S9	iPhone 7	
	iPhone 7	S9	iPhone 7	S9	
Re Re Re	S9	iPhone 7	S9	iPhone 7	S9
	iPhone 7	S9	iPhone 7	S9	iPhone 7
Fwd Fwd Fwd	S9	iPhone 7	S9	iPhone 7	S9
	iPhone 7	S9	iPhone 7	S9	iPhone 7

Tabelle 4.9: Szenarien der zweiten Versuchsreihe: erweiterte Szenarien zu den Funktionen Antworten und Weiterleiten

Neben einer E-Mail, die mehrere Adressaten, CCs und BCCs enthielt, um die Reihenfolge der Notierten Empfänger zu kontrollieren, wurden die folgenden Funktionen jeweils dreifach wiederholt: Entwürfe Speichern, E-Mails Archivieren, Mitteilungen einschalten und stumm schalten. Dies wurde getan, um zu klären, ob und evtl. mit welchem Muster, sie Schlüssel und IDs generieren. Diese Szenarien werden in Tabelle 4.10 aufgeführt.

Versuch	**Sender**	**Empfänger**	**CC**	**BCC**
Reihenfolge Check	A80	alle Geräte	alle Geräte	alle Geräte
Mitteilungen an Check	S9	iPhone 7		
Mitteilungen stumm Check	S9	iPhone 7		
Archivieren Check	S9	iPhone 7		
	iPhone 7	S9		

Tabelle 4.10: Szenarien der zweiten Versuchsreihe: Wiederholung von Szenarien Aufgrund von Unklarheiten bzgl. ihrer Ergebnisse

Um zu kontrollieren, ob Antworten anders eingetragen werden, wenn sie sich nicht auf den ursprünglichen Absender beziehen, wurden E-Mails mit mehreren Adressaten, CCs und BCCs versandt. Danach haben Empfänger jeweils anderen Empfängern geantwortet. Diese Szenarien werden in Tabelle 4.11 aufgeführt.

Versuch	**Sender**	**erster Empfänger**	**zweiter Empfänger**	**Antwort**	**Empfänger der Antwort**
Empfänger an Empfänger	A80	S9	iPhone 7	S9	iPhone 7
	A80	S9	iPhone 7	iPhone 7	S9
Versuch	**Sender**	**Empfänger**	**CC**	**Antwort**	**Empfänger der Antwort**
CC an Empfänger	A80	S9	iPhone 7	iPhone 7	S9
	A80	iPhone 7	S9	S9	iPhone 7

Versuch	**Sender**	**Empfänger**	**CC**	**BCC**	**Antwort**	**Empfänger der Antwort**
BCC an Empfänger	A80	S9	A80	iPhone 7	iPhone 7	S9
	A80	iPhone 7	A80	S9	S9	iPhone 7

Tabelle 4.11: Szenarien der zweiten Versuchsreihe: erweiterte Szenarien bzgl. der Korrespondenz zwischen Empfängern

Zuletzt wurde geklärt, wie eine Korrespondenz gespeichert wird, die mit unterschiedlichen Farben markiert wurde. Hierzu wurde jede eingehende E-Mail in einer Reihe von Antworten mit einer anderen Farbe markiert. Die folgende Tabelle 4.12 bildet eine zusammenhängende Korrespondenz ab.

Versuch	Sender/Antwort	Empfänger	Flaggenfarbe
Mehrfarbig	S9	iPhone 7	Orange
	iPhone 7	S9	
	S9	iPhone 7	Rot
	iPhone 7	S9	
	S9	iPhone 7	Lila
	iPhone 7	S9	
	S9	iPhone 7	Blau
	iPhone 7	S9	
	S9	iPhone 7	Gelb
	iPhone 7	S9	
	S9	iPhone 7	Grün
	iPhone 7	S9	
	S9	iPhone 7	Grau

Tabelle 4.12: Szenarien der zweiten Versuchsreihe: Überprüfen des Speicherns unterschiedlicher Markierungen innerhalb einer Korrespondenz

5 Ergebnisse

5.1 Überblick über Ordner und Datenbanken

Unter `\private\var\mobile\Library\Mail` liegen die Daten der Mail Applikation; in weiteren Dateipfaden in diesem Kapitel wird dieser Teil verkürzt als "...\Mail" dargestellt, um die Pfade eindeutig einer Applikation zuzuordnen zu können. Neben einigen leeren Ordnern und nicht relevanten Informationen, liegen hier die im Folgenden diskutierten Dateien und Datenbanken, welche in Tabelle 5.1 aufgelistet werden.

Ordner	**Inhalt**
AttachmentData	E-Mail Anhänge
AttachmentPlaceholders	blockierte Anhänge
D38DDD4C-9BC0-41F5-B06D-527D7778A7AA	.plist Datei bzgl. Posteingänge
LocalAccountId	.plist Datei bzgl. lokaler Posteingänge
MessageData	E-Mail Dateien
NonpurgeableMessageData	leer
Protected Index Journals	leer
RemoteContentURLCache	mehrere Cache Dateien
Datenbanken	**Inhalt**
Protected Index	E-Mail Text, Betreff, Adressaten
Envelope Index	verschiedene E-Mail Daten und Referenzen
Plist Dateien	**Inhalt**
MailboxCollections	Posteingang Einstellungen
Metadata	Wiederherstellungsschlüssel

Tabelle 5.1: Übersicht der Dateistrukturen unter iOS

Unter `\data\data\com.google.android.gm` liegen die Daten von Gmail, aufgeführt in Tabelle 5.2; in weiteren Dateipfaden in diesem Kapitel wird dieser Teil verkürzt als ".gm" dargestellt, um die Pfade eindeutig einer Applikation zuzuordnen zu können. Kopien der gesamten Daten liegen unter `\sbin\.magisk\mirror\data\data\com.google.android.gm`. In mehreren der Ordner sind Unterordner und Dateien verschiedenster Dateitypen mit wenigen bis keinen Informationen. Sie sind meist leer oder enthalten nur wenige Zeilen. Die meisten der hier zu findenden Informationen beziehen sich auf generelle Einstellungen der Applikation, Cookies, Caches oder Logs. Im Folgenden werden alle relevanten Ordner und Dateien diskutiert.

Ordner	Inhalt
app_textures	leer
app_webview	Einstellungen, Cookies, Logs
cache	Ordner: uploader, Einstellungen, Caches, Logs, leere Datenbanken
code_cache	leer
databases	verschiedene Datenbanken, Einstellungen
files	Ordner: downloads, Einstellungen
no_backup	Datenbank zu Interaktionen mit anderen Applikationen
shared_prefs	Einstellungen

Tabelle 5.2: Übersicht der Dateistrukturen unter Android

5.2 Datenbanken in Mail

In der Datenbank `Protected Index` sind `adresses`, `subjects` und `summaries` die relevant Tabellen. Aufgeführt in Tabelle 5.3. In `adresses` stehen die in E-Mails verwendeten Adressen, in der `comment` spalte stehen hier meist mit den Adressen assoziierte Namen. In `subjects` und `summaries` stehen jeweils alle Betreffe und Texte der E-Mails. Diese Tabellen werden jeweils über ihre `ROWIDs` im `Envelope Index` referenziert.

Protected Index Tabellen	Inhalt
additional_remote_content_links	leer
adresses	E-Mail Adressen und Namen
protected_message_data	leer
sqlite_sequence	Datenbank-Sequenzverwaltung
subjects	Betreffe
summaries	Texte

Tabelle 5.3: Übersicht der Tabellen der Protected Index Datenbank

In der Datenbank `Envelope Index` stehen die meisten relevanten Informationen, zu sehen in Tabelle 5.4; die verschiedenen Tabellen drehen sich hauptsächlich um die Tabellen `messages` und `threads`. Neben Tabellen mit allgemeineren Informationen, wie in der Tabelle `mailboxes` bzgl. der Postfächer, folgen die meisten Tabellen dem Schema ”messages_[XY]” und ”threads_[XY]” und enthalten Informationen in Relation zu diesen beiden Tabellen.

Envelope Index Tabellen	Inhalt
accounts	leer
action_flags	leer
action_labels	leer
action_messages	leer
attachments	Daten zu gespeicherten Anhängen
conversation_id_message_id	Verknüpfung conversation_id und message_id
conversations	Verknüpfung conversation_id und sync_key
local_message_actions	leer
mailboxes	URL der Posteingänge
message_attachments	Verknüpfung global_message_id und attachments
message_data	leer
message_data_deleted	leer
message_global_data	message_id (als global_message_id referenziert)
message_metadata	Codeschnipsel
message_references	Verknüpfung der IDs von beantworteten und weitergeleiteten E-Mails
messages	verschiedene Daten zu E-Mails
pop_uids	leer
properties	Werte zur Versions- und Synchronisationskontrolle
recipients	Aufzählung aller E-Mail Empfänger inklusive Typ und Position
remote_content_links	leer
searchable_attachments	Verknüpfung messages und attachments
searchable_message_tombstones	transaction_id gelöschter E-Mails
searchable_messages	Verknüpfung messages zu transaction_id
server_labels	leer
server_messages	verschiedene Daten zu E-Mails
sqlite_sequence	Datenbank-Sequenzverwaltung
table_metadata	höchste gelöschte ROWID mehrerer Tabellen
thread_mailboxes	Posteingänge aller Konversationen
thread_recipients	Empfänger aller Konversationen
thread_scopes	Referenzieren aller Posteingänge
thread_senders	Absender aller Konversationen
threads	verschiedene Daten zu Konversationen

Tabelle 5.4: Übersicht der Tabellen der Envelope Index Datenbank

In `message_references` werden die `ROWIDs` aller E-Mails, die nicht an erster Stelle einer Korrespondenz stehen, mit den `message_ids` der E-Mails in Verbindung gebracht, auf die sie sich beziehen. In message_metadata stehen in Referenz zu den global_message_ids unter json_values verschiedene Codeschnipsel, die allerdings mit Ausnahme der Größe der E-Mail, keine nachvollziehbar relevanten Informationen enthalten. Diese Tabelle wird zwar nicht gelöscht, es entstehen aber auch nicht für jede E-Mail ein Eintrag.
In `conversation_id_message_id` wird, wie vielleicht erkenntlich, ein Relation zwischen den beiden IDs hergestellt. Die `message_id` ist eine zufällige und nichtsequenzielle Zahl, die allen E-Mails zugewiesen wird. Sie kann sowohl positiv als auch negativ sein und hat bis zu 19 Ziffern. Die im Rahmen der Versuchsreihen erzeugten IDs hatten mindestens 17 Ziffern. Basierend auf den Tabellen, lässt sich nicht beweisen unter welchen Parametern die IDs er-

zeugt wurden. Es lassen sich aber Rückschlüsse ziehen. Aufgrund der Größe der Werte lässt sich auf einen signed 64 Bit Integer schließen, die minimalen und maximalen Werte eines solchen werden von den IDs nicht überschritten. Die Verteilung an 19, 18, und 17 stelligen Zahlen entspricht der Verteilung bei zufälligen signed 64 Bit Integern. Das Fehlen von 16 oder weniger stelligen Zahlen ist damit zu erklären, dass solche nur mit einer Wahrscheinlichkeit von $\sim$ 0,1% auftreten, wodurch sie bei der generierten Menge an IDs im Schnitt weniger als einmal vorkommen. Niedrig stellige Zahlen sollten bei einer ausreichend großen Menge an Testdaten allerdings aufzufinden sein, gegenteiliges würde für limitierungen bei der Erstellung der IDs sprechen.
Die `conversation_id` entspricht immer der ersten `message_id` innerhalb einer Korrespondenz. Sie wird immer erstellt, auch wenn es nur eine Nachricht gibt. Es liegen auch IDs ohne dazugehörige E-Mails vor, warum ist unklar. Um die genaue Erstellung und Zuweisung der IDs nachzuvollziehen, müsste entweder der Quellcode analysiert werden, oder die aktiven Prozesse der Applikation mitgeschnitten werden.
In `recipients` werden alle E-Mail Empfänger ihren jeweiligen E-Mails zugeordnet. Sie werden nach Empfänger, Cc und Bcc kategorisiert und ihre Position innerhalb dieses E-Mail Feldes festgehalten. Neben den nicht bekannten Bccs wird ebenfalls nicht notiert, wenn das iPhone selber im Bcc steht.
In der Tabelle `conversations` werden die `conversation_id`, eine `flag` und die `sync_key` in Verbindung gesetzt. Die `sync_key` tritt ausschließlich auf, wenn bei E-Mails Mitteilungen angeschaltet oder stumm geschaltet werden, dabei werden jeweils die Flaggen 1 und 4 zugewiesen. Sie ist Hexadezimal und hat 32 Stellen und wird an keiner weiteren Stelle verwendet. Da sie ausschließlich im Zusammenhang mit Mitteilungen auftaucht, wird sie wahrscheinlich nur von Funktionen außerhalb der Datenbanken referenziert.

Die Tabelle `messages` ist die größte Tabelle im `Envelope Index`, sie dient als Knotenpunkt und enthält die meisten relevanten Informationen. Neben `message_id`, `global_message_id` und `conversation_id`, steht hier noch die `document_id`. Diese ist hexadezimal, hat 32 Stellen und ist gleich des `X-Universally-Unique-Identifier` im Header der E-Mail. Beide dieser Kennungen tauchen ausschließlich in gespeicherten Entwürfen auf und werden nicht anderweitig referenziert. Der Header enthält außerdem genauere Informationen zu z.B. Textformat, die sonst nicht in gesendeten E-Mails auftauchen, sondern nur in empfangenen. Diese Besonderheiten lassen sich nur auf das Speichern in einem lokalem Postfach zurückführen. Da neben den Entwürfen das andere lokale Postfach der Postausgang ist, ließe sich dies nur überprüfen, indem E-Mails dort blockiert werden. Hier müsste aber zuerst die Grenze zwischen dem Postausgang und gesendeten Nachrichten bestimmt werden.
`ROWIDs` aus dem `Protected Index` bzgl. des Senders, Betreffs und des Textes werden Ihren entsprechenden E-Mails zugeordnet. Unter `subject_prefix` ist mit ”Re:” und ”Fwd:” gekennzeichnet, ob die E-Mail eine Antwort ist oder weitergeleitet wurde. Dies bezieht sich allerdings nur auf die Funktion, die bei dieser einen E-Mail genutzt wurde. Antworten auf Antworten oder mehrfach weitergeleitete E-Mails werden hier nicht gekennzeichnet. Es folgen weitere kleinere Informationen; die Sende- und Empfangsdaten in Form von Unix Zeitstempeln, die `ROWIDs` des Postfaches und des ursprünglichen Postfaches im Falle des Verschiebens, ob die E-Mail gelesen, gelöscht oder geflagged wurde und die Größe der E-Mail in Byte. Bei den Flaggen wird noch die Farbe mit Zahlen zwischen 0 und 6 gekennzeichnet. Aufgeführt in Tabelle 5.5.

Flag	0	1	2	3	4	5	6
Farbe	Rot	Orange	Gelb	Grün	Blau	Lila	Grau

Tabelle 5.5: Zuordnung von Farben der Flaggen des iOS-Markierungssystems zur verwendeten Ziffer in den Tabellen der Datenbanken

Die Größe der E-Mail schließt sowohl die `.emlx` Datei und `.emlxpart` Dateien zusammen, die im Ordner `...\Mail\MessageData` liegen, als auch die Dateien der Anhänge die im Ordner `...\Mail\AttachmentData` liegen. Dies geht aber bei E-Mails die in `partial.emlx` und `.emlxpart` Dateien getrennt sind, nicht genau auf. Eine genauere Betrachtung dieser Dateien folgt im Kapitel 5.4. Es bleiben wenige hundert bis wenige tausend Byte Unterschied zwischen den Werten in den Tabellen und den Summen der Dateien. Das Aufnehmen der Daten in die Tabelle und das Aufteilen der Dateien passiert also an unterschiedlichen Zeitpunkten zwischen dem Empfangen und Speichern der Daten.
Neben einigen leeren Spalten sind hier auch mit der `remote_id` und der `external_id` zwei ungenutzte IDs. Aufgrund ihrer Namen beziehen sie sich vermutlich auf Tabellen innerhalb der iCloud oder ähnlichen Konstrukten, sie werden aber sonst nicht genutzt. Zuletzt gibt es noch ungeklärte Spalten mit `flags` und `content_type`, auch sie tauchen nur hier auf. Die hier stehenden Zahlenkombinationen enthalten wahrscheinlich weitere Informationen über die E-Mails, diese konnten allerdings im Rahmen dieser Arbeit nicht entschlüsselt werden. Durch eine Analyse des Quellcodes oder intensivere und breitere Versuchsreihen, könnten zukünftige Arbeiten diese Informationen erlangen.

Informationen zu den Korrespondenzen enthält `threads`, ähnlich wie `messages` zu den einzelnen E-Mails. Auch hier gibt es weitere Tabellen, die spezifische Informationen in Referenz zu `threads` beinhalten. Da sich hier viel doppelt, wird im Folgenden hauptsächlich auf die Differenzen eingegangen. Da es sich um Korrespondenzen aus mehreren E-Mails handelt, fehlen die meisten einzigartigen Informationen wie IDs oder Referenzen zum Inhalt der E-Mail. Weitere Informationen, wie der Ankunftszeitpunkt oder ob die E-Mail gelesen, weitergeleitet etc. wurde, beziehen sich immer nur auf die letzte E-Mail innerhalb der Konversation. Mithilfe der `ROWID` aus `messages` werden die zuletzt eingegangene E-Mail und die zuletzt gelesene E-Mail referenziert; ein komplettes nachvollziehen der Konversation ist nur mit dieser Tabelle nicht möglich. Auf die ganze Korrespondenz beziehen sich nur `count` mit der Anzahl der enthaltenen E-Mails und die Spalten `has_[FARBE]_flag`. Hier steht für jede Farbe in einer eigenen Spalte, ob diese in der Konversation vorhanden ist.
Da hier keine kompletten Korrespondenzen abgebildet werden müssten diese mühsam über die IDs in `message_references` und `messages` zusammengesetzt werden. Dies ist zwar aufwendig, aber im Gegensatz zum Auslesen des E-Mail Textes, nicht durch die Schreiber dieser zu beeinflussen. Sollte man längere Korrespondenzen suchen bietet es sich also an die letzte E-Mail von `threads` mithilfe ihrer ID über `messages` zur E-Mail-Datei zu verfolgen und für eine erste Sichtung der Informationen den Text dieser auszulesen. Danach sollte aber eine lückenlose Verfolgung der Korrespondenz über die Tabellen folgen.

5.3 Datenbanken in Gmail

In vielen der in Tabelle 5.6 aufgeführten Datenbanken sind keine oder wenige Informationen. Im Folgenden werden erst einige übergreifende Informationen diskutiert, danach wird auf bestimmte Datenbanken eingegangen. Nicht genannte Datenbanken enthalten keine relevanten Informationen.
In einigen Datenbanken steht nur unter der Tabelle `android_metadata` in einer einzigen Spalte, `locale`, der Eintrag: de_DE. Diese Information wird in anderen Datenbanken nicht weiterverwendet, es wird nur hinterlegt, in welchem Land bzw. in welcher Sprache die Applikation arbeitet. Mehrere Datenbanken haben die Tabelle `room_master_table` mit der Spalte `identity_hash`. Der hier notierte Hash ist zwischen den Datenbanken unterschiedlich und wird nicht wieder verwendet. Es ist nicht bekannt, worauf er basiert oder wofür er genutzt wird.

databases Datenbanken	**Inhalt**
1_tasks.notifications	Lage, leere Tabellen
1_threads.notifications	Lage, leere Tabellen
-1_threads.notifications	Lage, leere Tabellen
accounts.notifications	IDs bzgl. Account, Login Status, letzte Login Zeiten
antonmeier2020@ gmail.com_room_notifications	Lage, Hash, leere Tabellen
bigTopDataDB.-2112818715	verschiedene Informationen bzgl. E-Mails
downloader	logt download Anfragen
EmailProvider	Lage, leere Tabellen
EmailProviderBody	Lage, leere Tabellen
gnp_database	IDs bzgl. Account, Login Status, letzte Login Zeiten, Lage, Hash, leere Tabellen
gnp_fcm_database	IDs bzgl. Account, Login Status, letzte Login Zeiten, Lage, Hash, leere Tabellen
growthkit	Zeitstempel und IDs zu Aktionen, leere Tabellen
metadata.-2112818715	verschiedene Informationen bzgl. Anhängen
og_cards	Lage, Hash, leere Tabellen
og_education	Einstellungen bzgl. Tooltips, Lage, Hash, leere Tabellen
peopleCache_antonmeier2020@ gmail.com_com.google_11	verschiedene Informationen bzgl. Adressen, Hash
pseudonymous_room_notifications	Lage, Hash, leere Tabellen
suggestions	Lage, leere Tabellen

Tabelle 5.6: Übersicht der Datenbanken im Unterordner `databeses`

Die Datenbanken `accounts_notifications`, `gnp_database` und `gnp_fcm_database` haben ähnliche Informationen bzgl. des Gmail Accounts. In `accounts_notifications` ist die Tabelle `accounts`, während die anderen beiden `gnp_accounts` Tabellen haben. Die Tabellen haben zwar leicht unterschiedliche Spalten, das beschränkt sich aber meist nur auf den Namen. Wirklich unterschiedliche Spalten sind meist leer. Alle drei führen die genutzte Gmail Adresse als Account Namen oder Account ID. Sie enthalten auch eine `obfuscated_gaia_id`; die Gaia-ID, Google Accounts and ID Administration ID, wird von Google in der Verwaltung verschiedenster Funktionen und Applikationen verwendet [50]. Sie wird hier zwar nicht an anderer Stelle verwendet, kann aber aufgrund der breiten Nutzung z.B. für OSINT Untersuchungen verwendet werden [51]. Alle Datenbanken enthalten weiterhin Informationen zum aktuellen Login, der letzten und ersten Login Zeit, Hashes des letzten Anmeldeversuches und einige IDs. Diese IDs sind bis auf eine `internal_target_id` unterschiedlich und werden nicht weiter verwendet.

In `dowloader` werden download Anfragen festgehalten. Neben einigen kleineren Informationen wird der Account, der Typ der Daten, die ID der entsprechenden E-Mail, die Zeit der Anfrage, die URL, der geplante Dateipfad und die Größe der Datei notiert. Die Einträge, die hier noch stehen, wurden aber anhand der Spalte `disallowed_over_metered` aufgrund einer getakteten Internetverbindung blockiert. Es wird weiterhin eine `download_id` vergeben, diese taucht nur hier auf und ist nicht fortlaufend. Es ist daher möglich, dass hier alle Anfragen herein geschrieben werden und nur nach erfolgreichem Ausführen gelöscht werden. Um dies zu überprüfen müsste allerdings der laufende Prozess mitgeschnitten werden.

In `metadata.-2112818715` steht in der Tabelle `attachment_metadata` Informationen zu den empfangenen Anhängen. Jeder Eintrag hat eine ID: ”[IDderE-Mail]_0.[NummerierungdesAnhangs]_1” für den Anhang und ”[IDderE-Mail]_0.[NummerierungdesAnhangs]_2” für das Vorschaubild. Sollten diese heruntergeladen worden sein, werden in anderen Spalten der Dateipfad und die Größe in Byte notiert. Es wird auch die Empfangszeit und der Zeitpunkt des letzten Öffnens hinterlegt. In der Spalte `retention_length_ms` steht für alle Einträge 2592000000, von Millisekunden umgerechnet ergibt dies 30 Tage. Es lässt sich daher schlussfolgern, dass durch das Anklicken der Vorschau die Dateien vorläufig für 30 Tage in der Applikation gespeichert werden. Dies geschieht unabhängig vom herunterladen der Dateien.

bigTopDataDB Tabellen	Inhalt
ads	ad_server_id, BLOB bzgl. Werbung, Rang
android_metadata	Lage
cluster_tombstones	leer
clusters	Einstellungen und IDs bzgl. Cluster
entity_changes_to_sync_up	leer
entity_sync_state	Token bzgl. Einstellungen
filters	leer
item_changes	verschiedene Informationen bzgl. Änderungen an E-Mails
item_message_attachments	verschiedene Informationen bzgl. Anhängen
item_message_doc_attachments	leer
item_message_tombstones	leer
item_message	verschiedene Informationen bzgl. E-Mails
item_sync_reasons	Verbindung von Korrespondenzen und sync_reasons
item_tombstones	IDs entfernter Korrespondenzen
item_visibility	Verbindung von E-Mails und Clustern, weitere Informationen
item_visibility_tombstones	leer
item_visibility_update_work	leer
items	verschiedene Informationen bzgl. Korrespondenzen
items_sync_state	Einstellungen und Informationen bzgl. Synchronisationsstatus
labels_count	Anzahl an Labels
local_bulk_op_work	leer
schema_version	Versions Nummer
settings	IDs und BLOBs von Einstellungen
sqlite_sequence	Datenbank-Sequenzverwaltung
sync_reasons	Synchronisationsgründe
system_properties	BLOBs bzgl. Systemeigenschaften
write_sequence_id	constant_key

Tabelle 5.7: Übersicht der Tabellen der bigTopDataDB Datenbank

In `bigTopDataDB.-2112818715` stehen die meisten Informationen bzgl. der E-Mails, aufgeführt in Tabelle 5.7. Neben leeren und bereits besprochenen, Datenbanken übergreifenden, Tabellen liegen hier auch einige Tabellen, mit geringem relevanten Informationsgehalt. So stehen in `ads` z.B. `BLOBs` mit URLs und Scripten für Werbungen, die innerhalb der Applikation abgespielt werden. Im Folgenden werden daher nur Tabellen mit Informationen zu E-Mials besprochen.

In der Tabelle `cluster` liegen Label und `BLOBs` mit Einstellungen zu diesen. Details konnten den `BLOBs` nicht entnommen werden. Die Label sind unterschiedlich zu denen, die im Rahmen der Versuchsreihen manuell zugewiesen werden konnten und tauchen in mehreren der folgenden Tabellen wieder auf. In `item_visibility` wird z.B. über die `ROID` von `cluster` E-Mails entsprechende Label zugewiesen. Basierend auf dem Namen `cluster` dienen diese Label der Kategorisierung der E-Mails. Da sie nicht manuell vergeben werden konnten und viele ”smartlabel” im Namen tragen, werden sie folglich automatisch vergeben. Die Anzahl dieser und weiterer Labels steht unter `label_count`.

Sämtliche Veränderungen an E-Mails stehen in `item_changes`. Die genauen Veränderungen stehen allerdings in einem `BLOB`, daher ist neben den E-Mail und Korrespondenz IDs und den zugewiesenen Labels nichts nachzuvollziehen. Teilweise wird auch Betreff oder E-Mails Adressen vermerkt. In einem separaten `BLOB` stehen nochmal extra die IDs der E-Mails, die verändert wurden und in einer weiteren Spalte ist die genaue Zeit vermerkt. Zuletzt steht bei den E-Mails mit den APK und EXE Anhängen vermerkt, dass diese nur lokal sind, da sie vom Versenden blockiert wurden. Sie haben auch einen anderen `change_state` von 1 im Gegensatz zur 3 aller anderen E-Mails. Weitere Informationen zu deren Bedeutung liegen nicht vor. Unter `item_message_attachments` liegen die Informationen aller empfangenen und gesendeten Anhänge, blockierte APK und EXE Dateien ausgeschlossen. Neben der `ROWID` der entsprechenden E-Mail liegen hier die URL, der Dateiname, Hash, Zeitpunkt der Synchronisation und eine weitere ID, die allerdings nicht weiter genutzt wird. Diese Tabelle dient wie vorher schon demonstriert Hauptsächlich zum Verbinden der Informationen.
Die Tabelle `item_messages` enthält die Hauptinformationen zu allen E-Mails, abzüglich verworfener Entwürfe. Die IDs aller E-Mails entsprechen einem bestimmten Muster. IDs gesendeter E-Mails fangen mit ”msg-a:r” an, gefolgt von einer Zahl die aus dem gleichen Grund wie die `message_id` von iOS ein zufälliger signed 64 Bit Integer ist. IDs empfangener E-Mails fangen mit ”msg-f:” an, gefolgt von einer 19-stelligen positiven Zahl die fortlaufend aber nicht konsekutiv ist. Diese Zahl ist die gleiche, wie die `legacy_storage_id` in einer weiteren Spalte. Ausnahme dieser Regel ist eine empfangene E-Mail, die zurückgestellt wurde. Die ursprüngliche E-Mail erhält die normale ID, aber die später erneut zugestellte E-Mail erhält einen eigenen Eintrag und eine ID, die mit ”msg-a:bump-” beginnt und mit der Zahl aus der ursprünglichen ID endet. Die ID fängt somit ähnlich wie die einer gesendete E-Mail an, behält aber einen Teil der ID der empfangenen E-Mail. Ihr wird außerdem keine `legacy_storage_id` generiert. Neben einer weiteren ungenutzten ID stehen hier noch in der Spalte `zipped_message_proto` `BLOBs`. Diese sind, vermutlich durch eine Komprimierung, nicht leserlich. Versuche die Art der Komprimierung zu erkennen oder die Daten zu entkomprimieren, waren nicht erfolgreich. Zuletzt werden in `is_invalidated` alle nicht versendeten E-Mails vermerkt; gespeicherte Entwürfe und die blockierten E-Mails mit APK und EXE Dateien.
Die Tabelle `items` ist das Äquivalent zu `item_messages` bzgl. Korrespondenzen. Die ID der Korrespondenzen ist genauso aufgebaut wie die der E-Mails, nur statt ”msg” fangen sie mit ”thread” an. Die darauf folgenden Ziffern und Zahlen sind gleich der ID der ersten E-Mail in der Korrespondenz. Es werden noch einige andere IDS aus `item_messages` wiederholt, am relevantesten sind aber die `BLOBs` in der Spalte `item_summary_proto`. Hier stehen IDs, E-Mail Adressen, Betreffe, Inhalte, Informationen zu den Anhängen, Label etc. zu den E-Mails in der Korrespondenz. Da es sich um einen `BLOB` handelt, sind diese Informationen natürlich sehr unübersichtlich. Die IDs und E-Mail Adressen sind für alle E-Mails vorhanden, somit lässt sich die Korrespondenz nachverfolgen. Betreff und Inhalt sind allerdings nicht immer für alle E-Mails vorhanden. Bei empfangenen vertraulichen E-Mails wird der Inhalt nie aufgeführt. Weitere Informationen sind nur nachzuvollziehen, wenn sie Teil des Inhalts der Nachricht sind und somit im Klartext vorliegen. In einigen weitergeleiteten E-Mails steht z.B., dass diese weitergeleitet wurden. Da im Gegensatz zu iOS die E-Mails Dateien nicht separat gespeichert werden und die `BLOBs` aus `item_messages` nicht zu lesen sind, ist der Inhalt mancher E-Mails nicht ohne Tools auszulesen.
Unter `item_tombstones` werden die IDs der archivierten und aus dem Posteingang entfernten Korrespondenzen vermerkt, sie sind allerdings immer noch unter `items` zu finden.

peopleCache_antonmeier2020@ gmail.com_com.google_11 Tabellen	Inhalt
CacheInfo	Zeitpunkt des letzten Updates, Anzahl der Kontakte, BLOB bzgl. Affinität
Contacts	Affinität, Typ, BLOB bzgl. Kontakt
ContextualCandidateContexts	leer
ContextualCandidateInfo	leer
ContextualCandidateTokens	leer
ContextualCandidateTokens_content	leer
ContextualCandidateTokens_docsize	leer
ContextualCandidateTokens_segdir	leer
ContextualCandidateTokens_segments	leer
ContextualCandidateTokens_stat	leer
ContextualCandidate	leer
RpcCache	Typ, E-Mail Adressen, Zeitstempel, BLOB bzgl. E-Mail Adresse
Tokens	Teile von E-Mails Adressen, Affinität, Typ
Tokens_content	Verknüpfung Tokens mit docic
Tokens_docsize	docid, BLOB bzgl. Größe
Tokens_segdir	BLOB bzgl. Tokens, leere Tabellen
Tokens_segments	leer
Tokens_stat	BLOB
android_metadata	Lage
room_master_table	Hash
sqlite_sequence	Datenbank-Sequenzverwaltung

Tabelle 5.8: Übersicht der Tabellen der peopleCache_antonmeier2020@gmail.com_com.google_11 Datenbank

In der Tabelle `peopleCache_antonmeier2020@gmail.com_com.google_11` liegen verschiedene Informationen bzgl. der Kontakte, aufgeführt in Tabelle 5.8. Unter `Contacts` steht eine ID, Affinität, ein Typ und ein `BLOB`. Die ID der Kontakte wird in mehreren Tabellen dieser Datenbank verwendet, die Affinität taucht auch mehrfach auf, hat aber keine konsistenten Werte per ID. Sie wird außerdem nur in dieser Datenbank verwendet und es ist nicht ersichtlich, wie sie sich zusammensetzt oder zwischen was die Affinität dargestellt wird. Unter Typ steht bei den Testdaten nur Person, und im `BLOB` stehen E-Mail Adresse, teils Namen des Kontaktes und teils Links zu Profilbildern.
`RpcCache` ist eine Liste aller E-Mail Adressen. Neben den Adressen liegen hier auch `BLOBs`; diese enthalten nochmal die Adressen, teils den Namen und teils Links zu Profilbildern, allerdings nicht deckungsgleich mit `Contacts`.
Unter `Tokens` liegen zu den `ROWIDs` aus `Contacts` die E-Mail Adressen und aus ihnen herausgebrochene Wortbruchstücke. Die genaue Stückelung ist allerdings nicht konsistent. Neben einer E-Mail Adresse liegt nur der Teil vor dem @, beim nächsten zusätzlich noch der E-Mail Anbieter und die Domainendung einzeln und bei einem anderen steht stattdessen der volle Name, der Vorname und der Nachname mit entsprechender Groß- und Kleinschreibung. In `Tokens_content` werden diese Bruchstücke einer `docid` zugewiesen und in `Tokens_docsize` wird diese einem `BLOB` zugewiesen. In diesem steht, aus wie vielen mit einem Leerzeichen separierten Worten das Token besteht. Weitere Tabellen sind entweder leer, wurden schon besprochen oder enthalten keine nennenswerten Informationen.

5.4 Anhänge in Mail

Unter `...\Mail\AttachmentData` liegen in nummerierten Ordnern die empfangenen Anhänge. Die Nummerierung der Ordner entspricht der `ROWID` des enthaltenem Anhangs in der Tabelle `attachments` in der später besprochenen `Envelope Index` Datenbank. Hier stehen auch Informationen zur Datei: der Name der gespeicherten Datei, die Größe in Byte, ein Hash und der Zeitpunkt des Downloads. In der Tabelle `message_attachments` werden die `global_message_id` die `ROWID` aus `attachments` und der ursprüngliche Name der Datei einander zugeordnet. Die `global_message_id` wird an verschiedenen Punkten verwendet, die später noch genauer betrachtet werden. Über diese IDs können z.B. Verbindungen zwischen einem Anhang und einer E-Mail hergestellt werden. In Abbildung 5.1 wird dies beispielhaft für eine MOV-Datei dargestellt, indem über mehrere Datenbänke der Speicherort des Anhangs mit dem der E-Mail-Datei verbunden wird. Die Dateien der Anhänge werden erst lokal gespeichert, wenn die E-Mail geöffnet wurde. Sie sind in ”attachment.[Dateiendung]” umbenannt und sonst unverändert. Bereits vorhandene bzw. inhaltlich gleiche Dateien werden nicht erneut abgespeichert. Wie bei den DOCX-Dateien zu sehen, referenzieren daher verschiedene E-Mails auf unterschiedlich benannte Anhänge, welche aber auf eine einzelne gespeicherte Datei verweisen.

Die E-Mails mit APK und EXE Dateien als Anhänge wurden nach dem versenden durch einen Google Server blockiert. Hierdurch wurde vom `mailer-daemon@icloud.com` eine E-Mail erstellt, die über die fehlerhafte Zustellung informiert und einen genaueren Bericht als Anhang enthält. Da hier durch die empfangenden Google Server blockiert wurde und, wie im Kapitel zu Android beschrieben, Gmail bereits das Versenden solcher Anhänge blockiert. Daher konnte nicht untersucht werden, wie Mail mit solchen eingehenden Anhängen umgeht.
In `...\Mail\AttachmentPlaceholder` liegen die blockierten Dateien und im Falle der APK liegen in einem Unterordner PNG, die aus diesen extrahiert wurden. Da alle anderen Anhänge inklusive der Berichte über die fehlerhafte Zustellung unter `...\Mail\AttachmentData` zu finden sind, ist hier der exklusive Speicherort für blockierte Dateien anzunehmen. Warum diese separat gespeichert werden oder warum einzelne PNG extrahiert wurden, ist weder durch den Bericht noch andere Merkmale zu erkennen.
Die extrahierten PNG könnten durch einen Fehler der Extraktionssoftware entstanden sein oder sie sind, wie das separate Speichern, Teil eines möglichen Sicherheitskonzeptes der Applikation. Diese Fragen könnten durch die Verwendung unterschiedlicher Tools und eine genauere Analyse der IT-Sicherheit in anderen Arbeiten untersucht werden. Des weiteren könnten Versuchsreihen zwischen zwei iOS Geräten Ergebnisse bringen die in dieser Arbeit aufgrund der initialen Geräteauswahl nicht untersucht werden konnten.

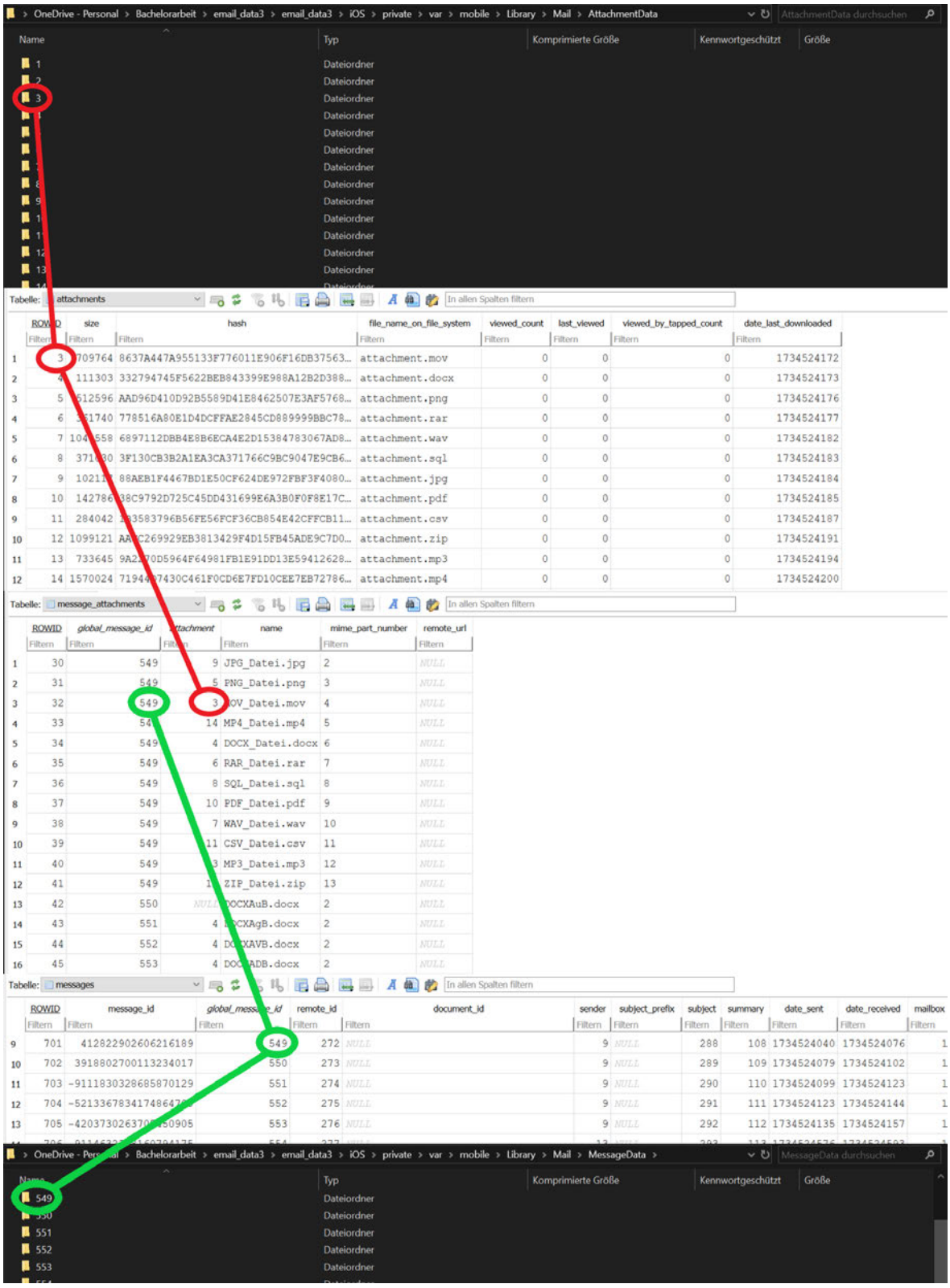

Abbildung 5.1: Beispielhafte Aufführung der Verbindungen zwischen den Datenbanken `attachments`, `message attachments`, `messages`, den Mail Dateien und den Dateien ihrer Anhänge: In rot wird über die `ROWID` (3) aus `attachments` eine Verbindung zwischen der Anhang Datei und den zu ihr hinterlegten Daten gezogen. In grün wird über die `global message id` (549) eine Verbindung zwischen diesen Daten zu den Daten bzgl. der E-Mail und der E-Mail Datei gezogen.

5.5 Anhänge in Gmail

Unter `.gm\cache` liegen neben nicht relevanten Ordnern verschiedene Tabellen die auch unter `.gm\databases` zu finden sind. In `.gm\databases` enthalten diese teils Informationen bzgl. der Anhänge von E-Mails; die Datenbanken in `.gm\cache` sind allerdings leer. Hier sind aber im Ordner `.gm\cache\uploader` die gesendeten Anhänge. Dort liegen Ordner deren Namen fast gleich der IDs der entsprechenden Nachrichten sind; der Name fängt mit einem Unterstrich an und ersetzt den Doppelpunkt der ID mit einem weiteren Unterstrich. In diesen Ordner liegen die jeweiligen Anhänge der E-Mails; ihren ursprünglichen Namen wurde eine von Null aufsteigende Zahl und ein Unterstrich vorangesetzt. Welche der versendeten Anhänge hier gespeichert werden, ist allerdings nicht konsistent.

Unter `.gm\files` liegen neben nicht relevanten Ordnern und verschiedenen Einstellungsdateien der Ordner `.gm\files\downloads`. In weiteren Unterordnern liegen die Anhänge der empfangenen E-Mails. `.gm\files\downloads\7d2012922ca76e80e8740f5ef43624f6\attachments` enthält Ordner mit den Namen ”d_0_0_[HashDesAnhangs]” für Anhänge und ”d_640_256_[HashDesAnhangs]” für deren Vorschaubilder. Die 640 und 256 beziehen sich meist auf die Maße des Bildes, das entweder als PNG oder sein originales Bildformat gespeichert wird. Einige textbasierte Anhänge haben auch größere Vorschaubilder, das verändert aber die Zahlen im Ordnernamen nicht.
Wodurch ein Vorschaubild erzeugt wird, ist nicht nachzuvollziehen. Die tatsächliche Datei wird hier aber erst durch das Anklicken der Vorschau heruntergeladen. Die Dateien und die Vorschaubilder werden mit den Namen der Dateien gespeichert. Dateien bzw. Vorschaubilder mit gleichem Hash, aber unterschiedlichen Namen, werden im Gleichen Ordner gespeichert.

Es gibt mehrere Datenbanken mit Informationen zu den Anhängen. In `downloader` und `metadata.-2112818715` stehen unteranderem die ID der E-Mail und der Dateipfad bestimmter Anhänge. Während in `bigTopDataDB.-2112818715` neben dem Inhalt der E-Mail auch die ID und der Hash stehen, die für die jeweiligen Speicherorte namensgebend sind. Es ist also mit allen drei Tabellen möglich, eine Verbindung zwischen dem Inhalt einer E-Mail und dessen Anhang herzustellen. Allerdings stehen nur in `bigTopDataDB.-2112818715` Informationen zu allen Anhängen. Beide Varianten sind beispielhaft in Abbildung 5.2 und 5.3 dargestellt. Vorausgesetzt alle Datenbanken sind vorhanden, ist es also sinnvoller `downloader` und `metadata.-2112818715` für zusätzliche Informationen und nicht zur Rückverfolgung zu nutzen. In Abbildung 5.2 wird die Verbindung zwischen dem Speicherort der von Gmail versendeten DOCX-Datei mit dem `BLOB` hergestellt, der Informationen zu der entsprechenden E-Mail und ihren Inhalt enthält. In Abbildung 5.3 wird eine solche Verbindung für die von Gmail empfangene Zeichnung hergestellt.

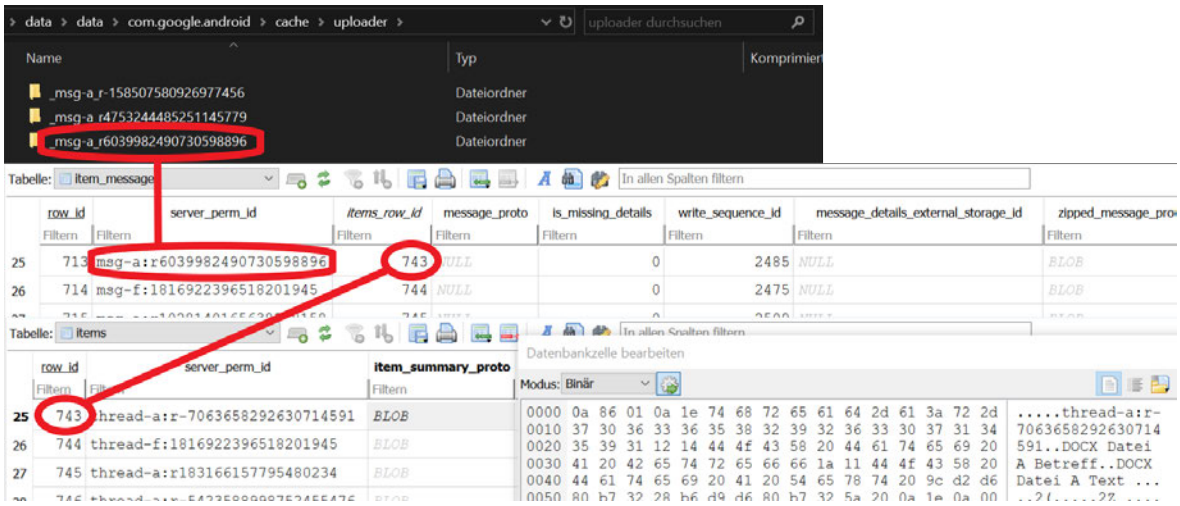

Abbildung 5.2: Beispielhafte Aufführung der Verbindungen zwischen Datenbanken und Dateien anhand der Anhänge: Über die ID der E-Mail und die `ROWID` (743) aus `items` wird eine Verbindung zwischen dem Ordner der gespeicherten Datei und dem Inhalt der E-Mai hergestellt.

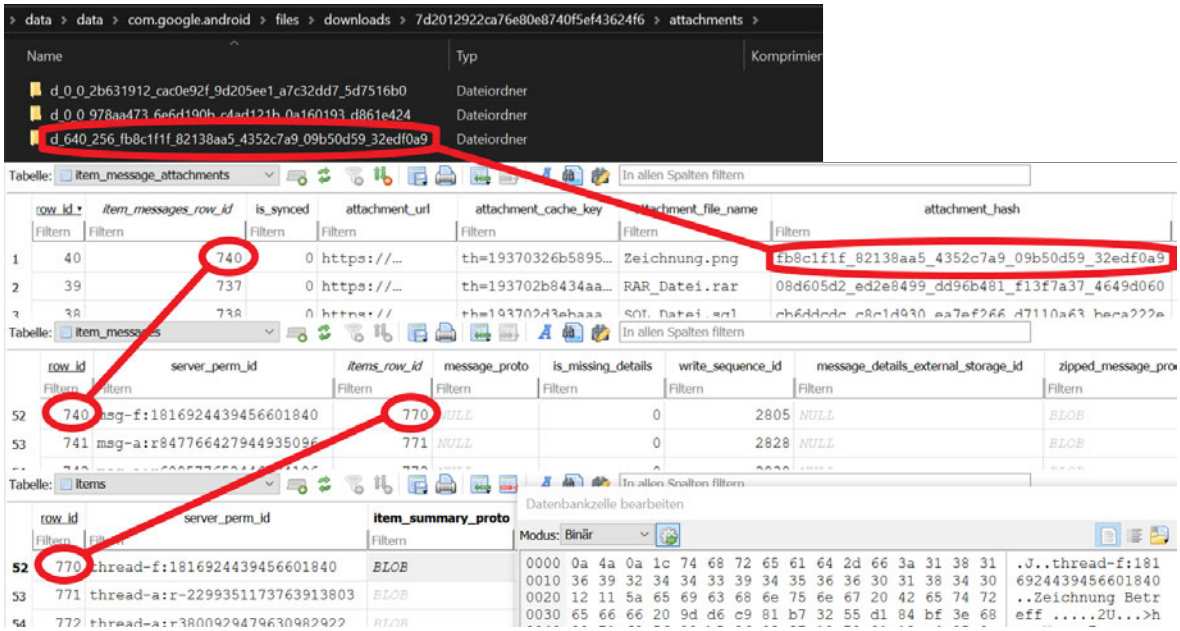

Abbildung 5.3: Beispielhafte Aufführung der Verbindungen zwischen Datenbanken und Dateien anhand der Anhänge: Über den Hash des Anhangs und die `ROWID` (740) aus `item_messages` und die `ROWID` (770) aus `items` wird eine Verbindung zwischen dem Ordner der gespeicherten Datei und dem Inhalt der E-Mai hergestellt.

5.6 Postfächer und `.plist` Dateien in Mail

Da Android keine `.plist` Dateien verwendet und vergleichbare Einstellungs-Dateien wenig bis keine relevanten Informationen enthalten, folgen hier nur Funde bzgl. Mail.

`...\Mail\LocalAccountId` enthält eine `.plist` Datei mit Einstellungen zu den lokal gespeicherten Postfächern; dem Postausgang und den Entwürfen. Eine `.plist` Datei im Ordner `...\Mail\D38DDD4C-9BC0-41F5-B06D-527D7778A7AA` enthält die Einstellungen zu den restlichen Postfächern. Apple verwendet `.plist` Dateien um im XML Format Eigenschaften und Konfigurationseinstellungen zu speichern [52]. Der Name des Ordners ist eine Referenz zum verwendeten IMAP Server und taucht in der URL der Postfächer in der Tabelle `mailboxes` unter `Envelope Index` auf, wie in Abbildung 5.4 zu sehen ist.

Tabelle: mailboxes

	ROWID	url
	Filtern	Filtern
1	1	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/INBOX
2	2	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Sent%20Messages
3	3	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Drafts
4	4	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Junk
5	5	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Deleted%20Messages
6	6	local://LocalAccountId/x-apple-transient-drafts
7	7	local://LocalAccountId/Outbox
8	9	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Neues%20Postfach
9	10	imap://D38DDD4C-9BC0-41F5-B06D-527D7778A7AA/Archive

Abbildung 5.4: Übersicht der URls der Postfächer in der Tabelle `mailboxes` inklusive ihrer Namen

Die `ROWIDs` der Postfächer werden in anderen Tabellen referenziert, abgesehen von den URLs liegen hier keine relevanten Informationen. Die Postfächer haben keinen Einfluss auf den tatsächlichen Speicherort der Daten. Die genauen Einstellungen der Postfächer haben ebenfalls wenig Relevanz, hier können aber die verwendeten Postfächer und die IMAP Adressen entnommen werden, falls dies bei den Datenbanken nicht möglich ist.
Weitere `.plist` Dateien enthalten keine struktur- oder inhaltsrelevanten Informationen.

5.7 E-Mail Dateien in Mail

Da die Inhalte von E-Mails bei Gmail in BLOBs innerhalb der Datenbanken stehen und in Informationsgehalt und Lesbarkeit nur schwer mit den E-Mail-Dateien von Mail zu vergleichen sind, werden diese hier separat betrachtet.

In `...\Mail\MessageData` sind in nummerierten Ordnern alle E-Mails, ausgenommen endgültig gelöschte E-Mails. Die Nummerierung ist gleich der `ROWID` der jeweiligen E-Mail in der Tabelle `message_global_data` in der Datenbank `Envelope Index`. Diese Nummer wird in anderen Tabellen und im Folgenden als `global_message_id` referenziert.
Die E-Mails liegen als `.emlx` Dateien vor und heißen entweder `full.emlx` oder `partial.emlx`. Die `.emlx` Dateien sind das standard Format für E-Mails Dateien der Mail Applikation [53]. Die Benennung dient hier zur Differenzierung zwischen kompletten und aufgeteilten Dateien. Eine solche Aufteilung erfolgte bei empfangenen E-Mails mit Anhängen und bei E-Mails die auf verschiedene Weisen verändert wurden z.B. aus dem Spamordner wieder in den Posteingang verschoben wurden. Welche Veränderungen zur Speicherung als `partial.emlx` führen, ist allerdings nicht konsistent. In fast allen Fällen enthält der Ordner eine nummerierte .emlxpart Datei die nur zwischen `<div dir="auto">` und `</div>` den Text der E-Mail enthält. Bei den automatischen Antworten durch die blockierten E-Mails liegen zwei `.emlxpart` Dateien. Dieses Dateiformat dient bei Mail als Anhang zu den `.emlx` Dateien [54]. Die eine enthält den generierten Bericht, die andere einen zusätzlichen E-Mail Header bzgl. der Rücksendeadresse. Diese werden im Kapitel zu den E-Mail Headern genauer betrachtet. Die dazugehörigen `partial.emlx` enthalten dementsprechend keinen Text, unterscheiden sich sonst allerdings nicht von den `full.emlx` Dateien.
Bei E-Mails mit Anhängen gibt es noch einen Ordner der entsprechend der `.emlx` Datei entweder `full_1.emlx` oder `partial_1.emlx` heißt. Dieser enthält alle Anhänge der E-Mail. Bei den versendeten E-Mails befinden sich im `full_1.emlx` Ordner die angehangenen Dateien, vollständig und mit korrektem Namen. Bei empfangenen E-Mails befinden sich im `partial_1.emlx` Ordner zwar korrekt benannte, aber leere Dateien, der Inhalt dieser befindet sich wie bereits genannt unter `...\Mail\AttachmentData`.

5.7.1 E-Mail Header

Die Hauptunterschiede der Header liegen zwischen den gesendeten und empfangenen E-Mails. Bei gesendeten E-Mails stehen nur wenige Informationen, wie in Abbildung 5.5 zu sehen: die Art des Inhalts, die Art der Zeichenkodierung und eine ID, die allerdings mit keiner ID in den Datenbanken übereinstimmt. Relevant sind hier hauptsächlich Betreff, Text, Sender, Empfänger und das Absendedatum. Diese Informationen sind zwar auch in verschiedenen Tabellen vorhanden, hier liegen sie aber zusammengefasst vor. Vor allem Korrespondenzen sind hier recht einfach auszulesen, denn in Antworten und weitergeleiteten E-Mails steht im Text auch der Text und weitere Informationen der vorherigen E-Mail, sollte dieser nicht absichtlich entfernt worden sein. Da die `.emlx` Dateien Klartext sind lassen sich diese Informationen im Zweifelsfall sogar mit einfachen Texteditoren auslesen.

```
Content-Type: text/plain;
        charset=us-ascii
Content-Transfer-Encoding: 7bit
From: =?utf-8?Q?Anja_M=C3=BCller?= <anjamueller2020@icloud.com>
Mime-Version: 1.0 (1.0)
Subject: Bcc Re all 3 Betreff
Date: Thu, 28 Nov 2024 02:30:21 +0100
Message-Id: <1A102D3E-ECAA-4EA6-9EBD-4E5886644BCF@icloud.com>
Bcc: Anton Meier <antonmeier2020@gmail.com>
Cc: vincentduckscheidt@gmail.com
To: vincentduckscheidt@gmail.com

Bcc Re all 3 Text

Von meinem iPhone gesendet
```

Abbildung 5.5: Beispiel für den Header einer gesendeten E-Mail

Header von empfangenen E-Mails, hier in Abbildung 5.6 zu sehen, sind dagegen wesentlich komplizierter. Neben den gleichen Informationen, die gesendete E-Mails enthalten, liegen noch Informationen zu passierten Servern und Protokollen vor. Es sind verschiedene Server vermerkt, die die E-Mails passiert haben und welche Aktionen dort durchgeführt wurden. So werden mehrere IDs, Wertungen oder Signaturen vergeben und es wird mehrfach authentifiziert. Eine genauere Betrachtung des Weges einer E-Mail und der verwendeten Server oder Protokolle ist nicht Teil dieser Arbeit, Schlussfolgerungen zwischen diesen Informationen und den zu untersuchenden Datenbanken und Dateien lassen sich nicht ziehen. Ebenfalls wird vermerkt, in welchen Posteingang die E-Mail kommen soll. Im Falle von Adressaten, die schon einmal als Spam markiert oder deren E-Mails in den Spam Ordner verschoben und dann wieder entfernt wurden, ändert sich der Befehl `MOVE_TO_FOLDER/INBOX` zu `WL/INBOX`. Bei empfangenen E-Mails steht außerdem der Text als Anhang am Ende des Headers.

```
Return-path: <vincentduckscheidt@gmail.com>
Original-recipient: rfc822;anjamueller2020@icloud.com
Received: from p00-icloudmta-smtpin-us-west-3a-60-percent-10 by p148-mailgateway-smtp-66dcb5fcb4-xzbhd (mailga
X-Apple-MoveToFolder: INBOX
X-Apple-Action: MOVE_TO_FOLDER/INBOX
X-Apple-UUID: f42a4dcb-dee2-4727-a54d-d200b76d5ec5
Received: from mail-ed1-f54.google.com (mail-ed1-f54.google.com [209.85.208.54])by p00-icloudmta-smtpin-us-wes
X-ICL-Info: GAtbRFYDBVFFS1VHSwQEUVUKE0oWX1gHVQoPB0UCB1tbQVNZSgAOTEBNRhIcDUALGxoDGQxUQwEeCgUfHVlSFjUeCxYRXBgBGH
X-ICL-Score: 3.33303303423
Authentication-Results: bimi.icloud.com; bimi=skipped reason="insufficient dmarc"
X-ARC-Info: policy=fail; arc=none
Authentication-Results: arc.icloud.com; arc=none
Authentication-Results: dmarc.icloud.com; dmarc=pass header.from=gmail.com
X-DMARC-Policy: v=DMARC1; p=none; sp=quarantine; rua=mailto:mailauth-reports@google.com
X-DMARC-Info: pass=pass; dmarc-policy=none; s=r1; d=r1; pdomain=gmail.com
Authentication-Results: dkim-verifier.icloud.com;dkim=pass (2048-bit key) header.d=gmail.com header.i=@gmail.c
Authentication-Results: spf.icloud.com; spf=pass (spf.icloud.com: domain of vincentduckscheidt@gmail.com desig
Received-SPF: pass (spf.icloud.com: domain of vincentduckscheidt@gmail.com designates 209.85.208.54 as permitt
Received: by mail-ed1-f54.google.com with SMTP id 4fb4d7f45d1cf-5cfadf851e2so444456a12.2for <anjamueller2020@i
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;d=gmail.com; s=20230601; t=1732757380; x=1733362180; darr
X-Google-DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;d=1e100.net; s=20230601; t=1732757380; x=173336
X-Forwarded-Encrypted: i=1; AJvYcCXjmnkv+cRKxB0W5ty87ryfVdZnxybEovvcaEs2GMHJgaFqG/A737w926qMfeWlkvCsWnI5JguLa\
X-Gm-Message-State: AOJu0YwpDEH4j7dt1WwgCNb1S6HZhVsaz0DUkk25pVAHcY61NOZY/TUxv+zrChu/vQNnrCPVqSK9hcSibD57wlvj97
X-Gm-Gg: ASbGncuYFdBgJwjxvF1T3TE+hI8bEwGsJwDBWY6thptlZAhQhIhddu8FF2YQmNtPpex1fxY3xD3e22I0nrS68Inm3u0HSFyyguW2I
X-Google-Smtp-Source: AGHT+IGoXrSnWGCcMs+Mb6idVSN9LQ2tVxTuHr4+QXzvJzH1hNx2AHtGeLVuTq0mUDBzCf5QuyT8/w7/GNCXh7T7
X-Received: by 2002:a17:906:2181:b0:aa5:1df4:f57d with SMTP ida640c23a62f3a-aa580ee0131mr442839366b.2.1732757
MIME-Version: 1.0
From: Vincent Duckscheidt <vincentduckscheidt@gmail.com>
Date: Thu, 28 Nov 2024 02:29:27 +0100
Message-ID: <CABom+Bj9VvdS1WZC=+OZNb5Arf7AEa1u6_Vm3kwyhAVtZ8+ibQ@mail.gmail.com>
Subject: BCC re all 2 Betreff
To: Anton Meier <antonmeier2020@gmail.com>
Cc: Vincent Duckscheidt <vincentduckscheidt@gmail.com>
Content-Type: multipart/alternative; boundary="000000000000193d890627ef047b"
X-MANTSH: 1TFkXGxoaHxEKWUQXbx99aHwBS0ZhTVkRC1lNF25PRkNcT1gRC19ZFxsZGxEKX00XZEVETxEKWUkXHR9xGwYbHxp3BhkbBhoGGgY
X-Proofpoint-ORIG-GUID: w_73wjZ-m6iAqR7IhK4R9feYpOa6JYDm
X-Authority-Info: v=2.4 cv=RMjhHJi+ c=1 sm=1 tr=0 ts=6747c784 cx=c_pps a=EmBPEBpMjP/3QNnZKm4stQ==:117 a=V1fZX
X-Proofpoint-GUID: w_73wjZ-m6iAqR7IhK4R9feYpOa6JYDm

--000000000000193d890627ef047b
Content-Type: text/plain; charset="UTF-8"

BCC re all 2 Text

--000000000000193d890627ef047b
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: quoted-printable

<div dir=3D"auto">BCC re all 2 Text=C2=A0</div>

--000000000000193d890627ef047b--
```

Abbildung 5.6: Beispiel für den Header einer empfangenen E-Mail (zur besseren Übersicht wurden mehrzeilig formatierte Punkte auf eine Zeile reduziert)

Bei `partial.emlx` Dateien ist im Gegensatz zur `full.emlx` Datei nur der Inhalt der E-Mail in eine `.emlxpart` Datei ausgelagert. Das spiegelt sich am Ende der Header wieder, wo der Klartext und der HTML Text ähnlich wie Anhänge vermerkt werden. Hier steht dann statt dem Text und dessen Typs nur Typ und Länge, wie in Abbildung 5.7 zu sehen.
Bei Anhängen, wie in Abbildung 5.8, steht hier neben Typ, Länge, Name und Dateiname noch die Art der Zeichenkodierung und eine ID, die allerdings auch nicht in den Datenbanken verwendet wird.

```
--0000000000007cefbc0627ed17d0
Content-Type: text/plain; charset="UTF-8"

Blank A Text

--0000000000007cefbc0627ed17d0
Content-Type: text/html; charset="UTF-8"

<div dir="auto">Blank A Text</div>

--0000000000007cefbc0627ed17d0--
```

```
--000000000000ef6d6706298b1f89
Content-Type: text/plain;
        charset=UTF-8
X-Apple-Content-Length: 22

--000000000000ef6d6706298b1f89
Content-Type: text/html;
        charset=UTF-8
X-Apple-Content-Length: 44

--000000000000ef6d6706298b1f89--
```

Abbildung 5.7: Text im Header einer full.emlx Datei (links) und einer partial.emlx Datei (rechts)

```
--000000000000387ab906298a5aa5
Content-Type: image/jpeg;
        name=JPG_Datei.jpg
Content-Disposition: attachment;
        filename=JPG_Datei.jpg
Content-Transfer-Encoding: base64
Content-Id: <193d83b9c58b51c12d01>
X-Apple-Content-Length: 139738
```

Abbildung 5.8: Beispiel für einen Anhang im Header einer E-Mail

5.8 Löschen von Daten

5.8.1 Löschen von Daten in Mail

Nach endgültigem Löschen sind Daten trotzdem noch vorhanden. Erst beim zusammenziehen der `.db` und `.db-wal` Dateien werden diese überschrieben. Im Betrieb der Applikationen wird beim erreichen eines Checkpoints die `.db-wal` auf die `.db` überschrieben. Dies passiert jedoc auch automatisch, wenn die `.db` mittels eines Datenbank Browsers geöffnet wird. Die `.db` Dateien wurden daher separat mit einem Datenbanken Browser betrachtet. Dadurch fehlen nur die Daten, die seit dem letzten Checkpoint auf der `.db-wal` sind. Um die `.db-wal` zu betrachten ist ein Hex-Editor nötig, da diese nicht als Datenbankdateien eingelesen werden können. Die Daten stehen hier zwar unverschlüsselt, sind allerdings durch die fehlende Struktur schwer nachzuvollziehen. Eine getrennte Betrachtung ist bei der Erstsichtung nötig. Dies gilt aber nur bei einer Sicherung unmittelbar nach der Löschung von E-Mails, da durch neue E-Mails die Daten sehr schnell überschrieben werden. Die Daten aus der ersten Versuchsreihe waren bei der Sicherung der gelöschten Daten noch alle aufzufinden. Bei der Sicherung der zweiten Versuchsreihe waren keine dieser mehr aufzufinden.
Alle `ROWIDs` des `Protected Index` und einige des `Envelope Index` sind Sequenziell, wenn auch nicht immer fortlaufend. Daher lässt sich erkennen, ob größere Mengen Daten gelöscht wurden. Tabellen wie `sqlite_sequence` und `table_metadata` werden nicht gelöscht; sie enthalten Informationen zur Größe anderer Tabellen. Es ist also auch nach dem Löschen zumindest möglich die Menge der gelöschten Daten nachzuvollziehen. Da `ROWIDs` sequenziell

sind werden diese Tabellen allerdings bei erster Gelegenheit überschrieben.
Weitere Methoden zur Datenwiederherstellung, wie z.B. über Tools oder Backups, wurden in dieser Arbeit nicht durchgeführt.

5.8.2 Löschen von Daten in Gmail

Genauso wie bei Mail werden die Daten erst dann entgültig gelöscht, wenn die `.db` und `.db-wal` Dateien zusammengezogen und damit überschrieben werden. Es gelten die gleichen Probleme bzgl. der Betrachtung und die gleiche Empfehlung zur getrennten Erstsichtung. Im Gegensatzt zu Mail werden hier allerdings nicht alle Daten überschrieben.
Die unter `.gm\cache\uploader` oder `.gm\files\downloads`liegenden Daten lassen sich auch teils nach dem Löschen und überschreiben der E-Mails wiederfinden. Teils bleiben die Dateien unverändert, teils liegt neben ihnen eine leere, formatlose Datei namens "[UnixZeitstempel].DELETED". Unter `.gm\cache\uploader` sind auch teils die Ordner leer und in "[VorherigerDateiname].DELETED" umbenannt. Hier liegen keine konsistenten Ergebnisse vor.
Die Datenbank `downloader` wird überhaupt nicht gelöscht aber dafür überschrieben. Ebenfalls werden `metadata.-2112818715` weder gelöscht noch überschrieben und `peopleCache_-antonmeier2020@gmail.com_com.google_11` wird nicht gelöscht und nur teils überschrieben. Leider wird `bigTopDataDB.-2112818715` bis auf die Tabelle `sqlite_sequence` komplett gelöscht und überschrieben.
Damit lassen sich nach dem Überschreiben auch ohne Tools zwar mehr Informationen als bei iOS nachvollziehen, vor allem bei den Anhängen, durch die fehlenden E-Mails sind aber die Hauptinformationen und alle komplexeren Zusammenhänge verloren. Wie bei Mail ist hier wieder für zukünftige Arbeiten offen, ob mithilfe spezieller Tools oder wie in anderen Arbeiten über Backups mehr zur Rekonstruktion gelöschter Daten möglich ist.

6 Fazit und Ausblick

Diese Arbeit hat die Daten- und Datenbankstrukturen hinter der Mail Applikation unter iOS und der Gmail Applikation unter Android untersucht. Durch mehrere Versuchsreihen und die Auswertung der Daten wurden die Speicherorte verschiedener forensisch relevanter Informationen aufgedeckt. Ebenso aufgedeckt wurden die Bedeutung und der Zusammenbau verschiedener IDs, Tags, Kennungen und Dateinamen. Es war möglich, separate Informationen zu verbinden, z.B. wurden die Dateien, die als Anhänge verschickt wurden, mit ihren entsprechenden E-Mail-Dateien bzw. den BL0Bs mit den Informationen der E-Mails in Verbindung gebracht. Des weiteren es war teils möglich, komplexere Strukturen nachzuvollziehen, wie z.B. den Zusammenhang der Korrespondenzen. Zur Veranschaulichung wurden diese auch graphisch aufgearbeitet. Weitere Komplexe wie z.B. Zeitstränge ließen sich auch anhand der aufgedeckten Daten mit entsprechenden Testdaten nachvollziehen.
Zwischen den Applikationen wurden einige Unterschiede festgestellt. So werden bei Gmail die gesendeten und empfangenen Anhänge separat gespeichert und bei Mail werden die E-Mails als Dateien und nicht in einer Tabelle gespeichert. Datenbanken von Mail referenzieren sich untereinander, während Gmail nur direkte Referenzen innerhalb einer Datenbank hat. Informationen in Gmail sind oft in BL0Bs unkenntlich während Mail meist im Klartext speichert. Weitere kleiner Unterschiede wurden in Kapitel 5 vermerkt.
Gemeinsam haben beide allerdings die Möglichkeit, durch das Write-Ahead Logging eigentlich schon gelöschte Daten noch auszulesen. Es wurde auch festgehalten, welche Daten selbst nach dem Löschen und Überschreiben noch nachzuvollziehen sind. Eine Wiederherstellung von Daten über Tools oder Backups könnte in zukünftigen Arbeiten versucht werden.

Neben einigen Flags und Kennungen die mithilfe der vorhandenen Testdaten nicht ergründet werden konnten, lag das Hauptproblem der Arbeit in der Analyse der BL0Bs. Da diese unstrukturiert sind und viele Informationen nur Hexadezimal vorliegen, war es oft nicht möglich sie auszulesen. Dies lässt sich eventuell durch breitere Versuchsreihen und eine aufwendige Analyse dieser lösen. Die Nutzung verschiedener spezifischer Analysetools ist jedoch auch ein Ansatz.
Zukünftige Arbeiten können neben Variationen in Applikationen, Experimenten und Werkzeugen aber vor allem durch einen anderen Fokus, Ergebnisse erzielen, die in dieser Arbeit nicht gesucht wurden. Durch eine Analyse der E-Mail Header, verwendeten Protokolle oder einer Verfolgung von online-IDs wie der Gaia ID könnten online forensische Untersuchungen durchgeführt werden, die in dieser Arbeit nicht angerührt wurden. Sollte es möglich sein, an den Quellcode der Applikationen zu gelangen, könnten Fragen geklärt werden, die nur mit den durch Versuchsreihen ausgegebenen Daten unbeantwortet blieben.

Zusammengefasst ist es mit dieser Arbeit möglich, verschiedene forensisch relevante Daten und deren Zusammenhänge zu finden. Da sowohl Speicherorte der übertragenen Daten als auch der Personenbezogenen Daten von Sendern und Empfängern gefunden wurden. Diese verstreuten Daten können mit Hilfe dieser Arbeit wieder zusammengefügt werden. Das gleiche gilt für komplexere Zusammenhänge wie z.B. Korrespondenzen. Außerdem ist nun bekannt, welche Daten nach dem Löschen und Überschreiben durch das Write-Ahead-Logging bei Mail und Gmail noch aufzufinden sind und wo diese gespeichert werden. Das Verstehen

der zugrundeliegenden Strukturen und des Zusammenbaus von Daten, Datenbanken und Funktionen ermöglicht weitere Schlüsse auf Informationen zu ziehen, die sonst unverständlich blieben. Es war zwar nicht möglich die BL0Bs zu entschlüsseln, dennoch wird es durch die entschlüsselten internen Kennungen einfacher die o.g. komplexeren Zusammenhänge nachzuvollziehen. Zusätzlich zu den Informationen die sie selber preisgeben. Mit Hilfe dieser Arbeit ist es möglich diese Informationen und Zusammenhänge für die Applikationen Mail und Gmail zu finden und nachzuvollziehen.

Literaturverzeichnis

[1] K. Davies. „Internet usage in Germany - statistics facts“. (2024), Adresse: https://www.statista.com/topics/6636/internet-usage-in-germany/#topicOverview (besucht am 23. 02. 2025).

[2] Statista Research Department. „Statistiken zur Internetnutzung in Deutschland“. (2024), Adresse: https://de.statista.com/themen/2033/internetnutzung-in-deutschland/#topicOverview (besucht am 23. 02. 2025).

[3] S. M. Bellovin, M. Blaze, S. Clark und S. Landau, „Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet“, *Privacy Legal Scholars Conference, June 2013*, Aug. 2013. DOI: http://dx.doi.org/10.2139/ssrn.2312107. Adresse: https://ssrn.com/abstract=2312107.

[4] S. Ludewig. „Die Sicherstellung und Auswertung des Smartphones – Kriminalpolitischer Anpassungsbedarf?“ (2019), Adresse: https://kripoz.de/2019/09/18/die-sicherstellung-und-auswertung-des-smartphones-kriminalpolitischer-anpassungsbedarf/ (besucht am 10. 02. 2025).

[5] H. Nier. „Wie sich die digitale Kommunikation verändert hat“. (2017), Adresse: https://de.statista.com/infografik/11426/wie-sich-die-digitale-kommunikation-veraendert-hat/ (besucht am 23. 02. 2025).

[6] M. Le. „2021 kicks off with Apple iPhone increasing its lead in email client market share“. (2021), Adresse: https://www.litmus.com/blog/email-client-market-share-2021-q1 (besucht am 10. 02. 2025).

[7] M. I. Husain und R. Sridhar, „iForensics: Forensic Analysis of Instant Messaging on Smart Phones“, in *Digital Forensics and Cyber Crime*, S. Goel, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, S. 9–18, ISBN: 978-3-642-11534-9. DOI: https://doi.org/10.1007/978-3-642-11534-9_2.

[8] A. Mahajan, M. S. Dahiya und H. P. Sanghvi, „Forensic Analysis of Instant Messenger Applications on Android Devices“, *International Journal of Computer Applications*, Jg. 68, Nr. 8, S. 38–44, Apr. 2013, ISSN: 0975-8887. DOI: 10.5120/11602-6965. Adresse: http://dx.doi.org/10.5120/11602-6965.

[9] N. S. Thakur. „Forensic Analysis of WhatsApp on Android Smartphones“. (2013), Adresse: https://scholarworks.uno.edu/td/1706 (besucht am 10. 02. 2025).

[10] C. Anglano, M. Canonico und M. Guazzone, „Forensic analysis of the ChatSecure instant messaging application on android smartphones“, *Digital Investigation*, Jg. 19, S. 44–59, 2016, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2016.10.001. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287616300950.

[11] L. Zhang, F. Yu und Q. Ji, „The Forensic Analysis of WeChat Message“, in *2016 Sixth International Conference on Instrumentation Measurement, Computer, Communication and Control (IMCCC)*, 2016, S. 500–503. DOI: 10.1109/IMCCC.2016.24.

[12] S. Wu, Y. Zhang, X. Wang, X. Xiong und L. Du, „Forensic analysis of WeChat on Android smartphones“, *Digital Investigation*, Jg. 21, S. 3–10, 2017, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2016.11.002. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287616301220.

[13] G. B. Satrya, P. T. Daely und M. A. Nugroho, „Digital forensic analysis of Telegram Messenger on Android devices“, in *2016 International Conference on Information Communication Technology and Systems (ICTS)*, 2016, S. 1–7. DOI: 10.1109/ICTS.2016.7910263.

[14] C. Anglano, „Forensic analysis of WhatsApp Messenger on Android smartphones“, *Digital Investigation*, Jg. 11, Nr. 3, S. 201–213, 2014, Special Issue: Embedded Forensics, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2014.04.003. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287614000437.

[15] C. Anglano, M. Canonico und M. Guazzone, „Forensic analysis of Telegram Messenger on Android smartphones“, *Digital Investigation*, Jg. 23, S. 31–49, 2017, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2017.09.002. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287617301767.

[16] J. Gregorio, A. Gardel und B. Alarcos, „Forensic analysis of Telegram Messenger for Windows Phone“, *Digital Investigation*, Jg. 22, S. 88–106, 2017, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2017.07.004. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287617301032.

[17] L. Chen und Y. Mao, „Forensic Analysis of Email on Android Volatile Memory“, in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, S. 945–951. DOI: 10.1109/TrustCom.2016.0160.

[18] Statista Research Department. „Marktanteile der führenden mobilen Betriebssysteme an der Internetnutzung mit Mobiltelefonen in Deutschland von Januar 2009 bis Juli 2024“. (2024), Adresse: https://de.statista.com/statistik/daten/studie/184332/umfrage/marktanteil-der-mobilen-betriebssysteme-in-deutschland-seit-2009/ (besucht am 10. 02. 2025).

[19] S. Singh. „The Complete List of Pre-Installed Apps on Android Devices“. (2023), Adresse: https://www.devicemag.com/android-pre-installed-apps-list/ (besucht am 10. 02. 2025).

[20] K. Shaw. „A Complete Guide to Default iPhone Apps and What They Do“. (2024), Adresse: https://geekschalk.com/a-list-of-all-default-iphone-apps-and-what-they-do/ (besucht am 10. 02. 2025).

[21] M. Kim, E. Lim, S. Lee und Y. Won, „Dumb design, dumber usage of mobile database“, in *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, 2015, S. 612–613. DOI: 10.1109/GCCE.2015.7398632.

[22] J. Wagner, A. Rasin und J. Grier, „Database forensic analysis through internal structure carving“, *Digital Investigation*, Jg. 14, S106–S115, 2015, The Proceedings of the Fifteenth Annual DFRWS Conference, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2015.05.013. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287615000584.

[23] H. Pieterse und M. Olivier, „Smartphones as Distributed Witnesses for Digital Forensics“, in *Advances in Digital Forensics X*, G. Peterson und S. Shenoi, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 237–251. DOI: https://doi.org/10.1007/978-3-662-44952-3_1.

[24] H. Chivers, „Navigating the Windows Mail database“, *Digital Investigation*, Jg. 26, S. 92–99, 2018, ISSN: 1742-2876. DOI: https://doi.org/10.1016/j.diin.2018.02.001. Adresse: https://www.sciencedirect.com/science/article/pii/S1742287617303547.

[25] K. Karkazan. „Investigating Android Gmail“. (2023), Adresse: https://forensafe.com/blogs/Android_Gmail.html (besucht am 10. 02. 2025).

[26] Bundesamt für Sicherheit in der Informationstechnik. „Leitfaden „IT-Forensik““. (2011), Adresse: https://www.bsi.bund.de/dok/6620610 (besucht am 22. 05. 2025).

[27] R. Tamma, O. Skulkin, H. Mahalik und S. Bommisetty, *Practical Mobile Forensics: Forensically investigate and analyze iOS, Android, and Windows 10 devices, 4th Edition*. Packt Publishing, 2020, ISBN: 9781838644420. Adresse: https://books.google.de/books?id=TU_cDwAAQBAJ.

[28] Mail Designer 365. „A macOS Timeline: How Apple Mail Has Evolved Over the Years“. (2018), Adresse: https://www.maildesigner365.com/a-macos-timeline-how-apple-mail-has-evolved-over-the-years/ (besucht am 11. 05. 2025).

[29] Apple. „iPhone – Benutzerhandbuch“. (2025), Adresse: https://support.apple.com/de-de/guide/iphone/welcome/ios (besucht am 22. 05. 2025).

[30] Microsoft-Support. „Was sind IMAP und POP?“ (2025), Adresse: https://support.microsoft.com/de-de/office/was-sind-imap-und-pop-ca2c5799-49f9-4079-aefe-ddca85d5b1c9 (besucht am 11. 05. 2025).

[31] A. Melnikov und C. Newman. „IMAP URL Scheme“. (2007), Adresse: https://www.rfc-editor.org/rfc/rfc5092.html (besucht am 11. 05. 2025).

[32] M. Rouse. „What is Gmail? Definition, How It Works Features - Techopedia“. (2024), Adresse: https://www.techopedia.com/definition/gmail (besucht am 11. 05. 2025).

[33] Jonnalagadda, Harish. „Gmail for Android: How to do everything“. (2018), Adresse: https://www.androidcentral.com/gmail-android-how-to (besucht am 22. 05. 2025).

[34] IONOS Redaktion. „BLOBs (Binary Large Objects): Eine Einführung“. (2019), Adresse: https://www.ionos.de/digitalguide/websites/web-entwicklung/binary-large-object-blob/ (besucht am 11. 05. 2025).

[35] S. Bhosale, M. Patil und P. Patil, „International Journal of Computer Science and Mobile Computing SQLite: Light Database System“, *International Journal of Computer Science and Mobile Computing*, Jg. 44, S. 882–885, Apr. 2015.

[36] SQLite Documentation. „WAL-mode File Format“. (2025), Adresse: https://www.sqlite.org/walformat.html (besucht am 22. 05. 2025).

[37] SQLite Documentation. „Write-Ahead Logging“. (2024), Adresse: https://www.sqlite.org/wal.html (besucht am 11. 05. 2025).

[38] SQLite Documentation. „CREATE TABLE“. (2024), Adresse: https://sqlite.org/lang_createtable.html#rowid (besucht am 11. 05. 2025).

[39] Cellebrite. „Cellebrite UFED“. (2024), Adresse: https://cellebrite.com/de/cellebrite-ufed-de/ (besucht am 11. 05. 2025).

[40] Cellebrite. „Cellebrite Inseyets“. (2024), Adresse: https://cellebrite.com/de/cellebrite-inseyets-de/ (besucht am 11. 05. 2025).

[41] Cellebrite. „Cellebrite Physical Analyser“. (2024), Adresse: https://cellebrite.com/de/cellebrite-physical-analyzer-de/ (besucht am 11. 05. 2025).

[42] M. Piacentini. „DB Browser for SQLite - Standard installer for 64-bit Windows“. (2024), Adresse: https://sqlitebrowser.org/dl/ (besucht am 13. 03. 2025).

[43] mh-nexus. „HxD 2.5.0.0“. (2024), Adresse: https://www.heise.de/download/product/hxd-50764 (besucht am 13. 03. 2025).

[44] Kernel Data Recovery. „Free EML Viewer tool to open and view EML files“. (2024), Adresse: https://www.nucleustechnologies.com/eml-viewer.html (besucht am 13. 03. 2025).

[45] FileFlip.app. „Understanding Popular File Types: A Comprehensive Guide“. (2023), Adresse: https://www.fileflip.app/blog/understanding-popular-file-types-a-comprehensive-guide (besucht am 23. 02. 2025).

[46] File Examples. „Sample Files Download“. (2024), Adresse: https://file-examples.com/ (besucht am 26. 03. 2025).

[47] Sample Files. „Download Sample Files“. (2024), Adresse: https://getsamplefiles.com/ (besucht am 26. 03. 2025).

[48] Sample Developer. „Test DPC 9.0.12“. (2024), Adresse: https://www.apkmirror.com/apk/sample-developer/test-dpc/test-dpc-9-0-12-release/test-dpc-9-0-12-android-apk-download/ (besucht am 26. 03. 2025).

[49] M. Kochanski. „Sample databases“. (2016), Adresse: https://www.cardbox.com/downloading/other/samples.htm (besucht am 26. 03. 2025).

[50] Google for Developers. „Dokumentation“. (2025), Adresse: https://developers.google.com/issue-tracker/concepts/access-control?hl=de (besucht am 11. 05. 2025).

[51] FUD0. „Google OSINT with GHunt“. (2020), Adresse: https://www.securitydojo.info/blog/en/article/google-osint-with-ghunt/ (besucht am 11. 05. 2025).

[52] DateiWiki. „.plist Dateierweiterung“. (2025), Adresse: https://datei.wiki/extension/plist (besucht am 11. 05. 2025).

[53] DateiWiki. „.emlx Dateierweiterung“. (2025), Adresse: https://datei.wiki/extension/emlx (besucht am 11. 05. 2025).

[54] DateiWiki. „.emlxpart Dateierweiterung“. (2025), Adresse: https://datei.wiki/extension/emlxpart (besucht am 11. 05. 2025).

Eidesstattliche Erklärung

Hiermit versichere ich – Vincent Duckscheidt – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 26. Mai 2025	
Ort, Datum	Vincent Duckscheidt