



**HOCHSCHULE
MITTWEIDA**
University of Applied Sciences

Forensic Analysis of Telegram Messenger on iOS Smartphones

Lukas Jaeckel, M. Sc.

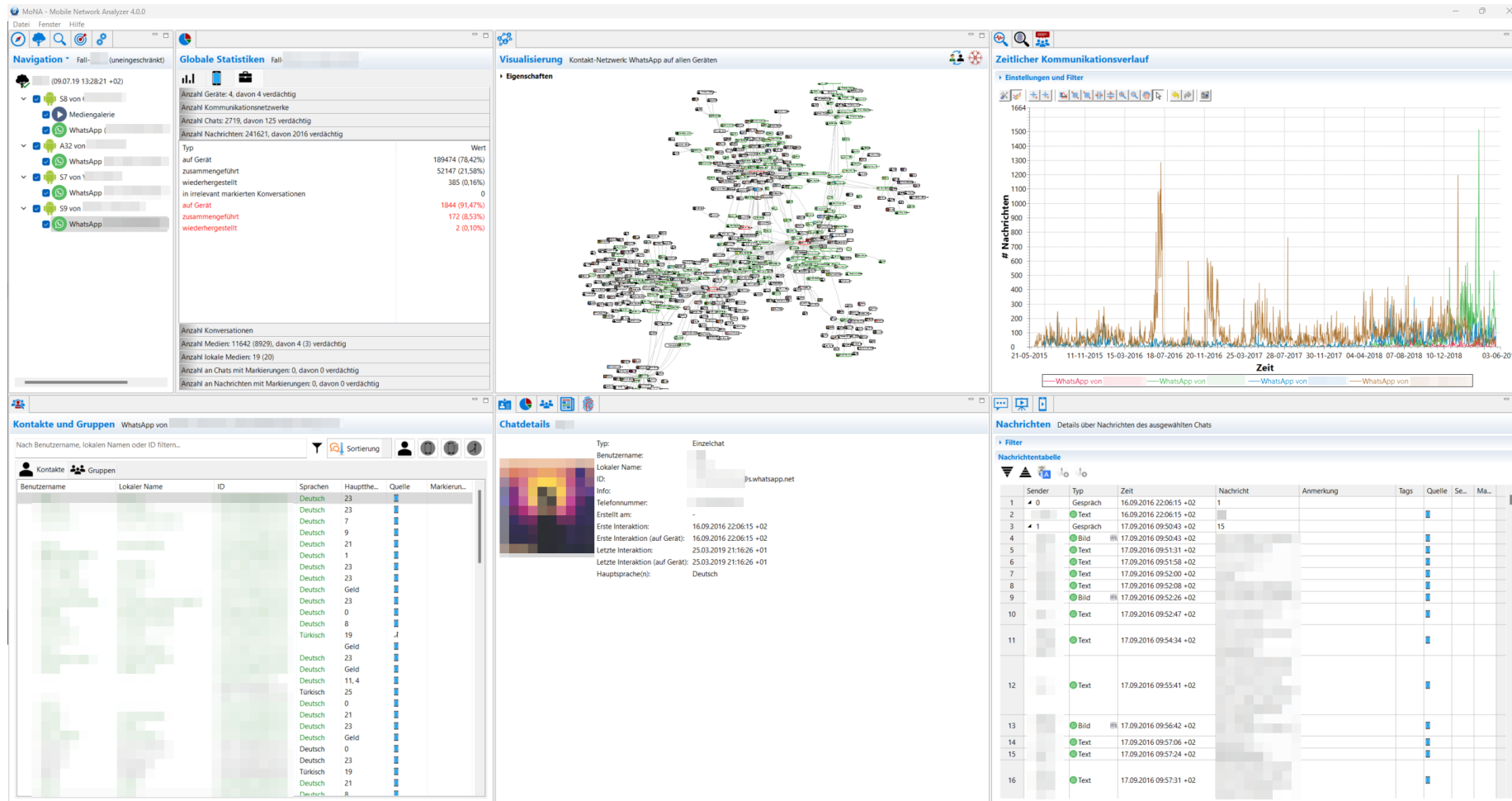
hs-mittweida.de/en

Forensic Science at Mittweida University of Applied Sciences

- **Study courses:**
 - General and digital forensic science (B. Sc.)
 - Cybercrime/Cybersecurity (M. Sc.)
 - IT-Forensics/Cybercrime (Distance learning studies, B. Sc.)
- **Main research areas:**
 - Digitalisation of forensic casework
 - IoT forensics
 - Communication and social network analysis



MoNA: The Mobile Network Analyzer



Analysis of messenger services

- E.g. WhatsApp Android

	key_id	key_remote_jid	key_from_me	timestamp	received_timestamp	data
1	1329115800-1	393485[REDACTED]@s.whatsapp.net	0	1329116347000	1329116349643	Message 1
2	1329116349-1	393485[REDACTED]@s.whatsapp.net	1	1329116423505	1329116423532	Reply 1
3	1329115800-2	393485[REDACTED]@s.whatsapp.net	0	1329116791000	1329116793357	Message 2
4	1329116349-2	393485[REDACTED]@s.whatsapp.net	1	1329116941607	1329116941626	Reply 2

Source: Anglano, C. (2014). Forensic Analysis of WhatsApp Messenger on Android Smartphones. Digital Investigation. 11. 10.1016/j.diin.2014.04.003.

The Challenge: Encoding messenger data

- Converting the data into an analysable format is usually necessary
 - Contact entries from a local Telegram database on an iOS device :

Table: t2

	key	value
	Filter	Filter
1	777000	BLOB
2	136817688	BLOB
3	140267078	BLOB
4	557478932	BLOB
5	9814559261	BLOB

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 01 5F 05 2B A5 68 9E 6E 01 00 00 01 69 01 28 DB ..+¥hžn....i.(Û
00000010 0B 00 00 00 00 00 02 61 68 01 45 A1 95 5B 50 7D .....ah.E;.[P}
00000020 89 87 03 61 68 74 00 00 00 00 00 02 66 6E 04 08 %#.aht.....fn..
00000030 00 00 00 54 65 6C 65 67 72 61 6D 01 70 04 05 00 ...Telegram.p...
00000040 00 00 34 32 37 37 37 02 70 68 08 02 00 00 00 7E ..42777.ph.....~
00000050 87 C9 69 6C 00 00 00 02 64 78 00 50 00 00 00 02 +Ëil....dx.P....
00000060 64 79 00 50 00 00 00 01 72 05 89 3E 05 37 1F 00 dy.P.....r.%.7..
00000070 00 00 01 64 00 01 00 00 00 01 70 01 B1 A7 31 1B ...d.....p.±$l.
00000080 28 DB 0B 00 01 73 00 00 00 00 00 01 76 0B 01 6C (Û...s.....v..l
00000090 0B 02 70 73 06 00 00 00 00 02 74 68 0A 10 00 00 ..ps.....th....
000000A0 00 01 08 08 90 79 7E 59 E4 EF A2 8A 2B A6 C7 25 .....y~YÄiçŠ+!Ç%
000000B0 CF 02 68 76 02 00 02 69 70 02 00 02 74 68 00 00 ĩ.hv...ip...th..
000000C0 00 00 00 7E 87 C9 69 6C 00 00 00 02 64 78 00 80 ...~+Ëil....dx.e
000000D0 02 00 00 02 64 79 00 80 02 00 00 01 72 05 89 3E ....dy.€.....r.%.>
000000E0 05 37 1F 00 00 00 01 64 00 01 00 00 00 01 70 01 .7.....d.....p.
000000F0 B1 A7 31 1B 28 DB 0B 00 01 73 00 01 00 00 00 01 ±$l.(Û...s.....
00000100 76 0B 01 6C 0B 02 70 73 06 00 00 00 02 74 68 v..l..ps.....th
00000110 0A 10 00 00 00 01 08 08 90 79 7E 59 E4 EF A2 8A .....y~YÄiçŠ
00000120 2B A6 C7 25 CF 02 68 76 02 00 02 69 70 02 00 02 +!Ç%ĩ.hv...ip...
00000130 74 68 00 00 00 00 02 62 69 0B 02 72 69 0B 02 th.....bi..ri..
00000140 66 6C 00 03 00 00 00 04 65 6D 6A 73 0B 03 75 6E fl.....emjs..un
00000150 73 08 00 00 00 00 03 73 74 68 02 00 04 6E 63 6C s.....sth....ncl
00000160 72 0B 04 62 67 65 6D 0B 04 70 63 6C 72 0B 04 70 r..bgem..pclr..p
00000170 67 65 6D 0B 03 73 73 63 0B gem...ssc.
```

- **Aim of the work:**
 - Detection, decoding & interpretation of relevant Telegram data on iOS devices

Communication on Telegram

- Free instant messaging service
- 2013: Publication by Pavel & Nikolai Durov
- Today: 900 million active users per month
- Communication in:
 - **Usual chats** between 2 users
 - **Secret chats** between 2 users
 - **Groups** with up to 200,000 users
 - **Channels** with an unlimited number of users

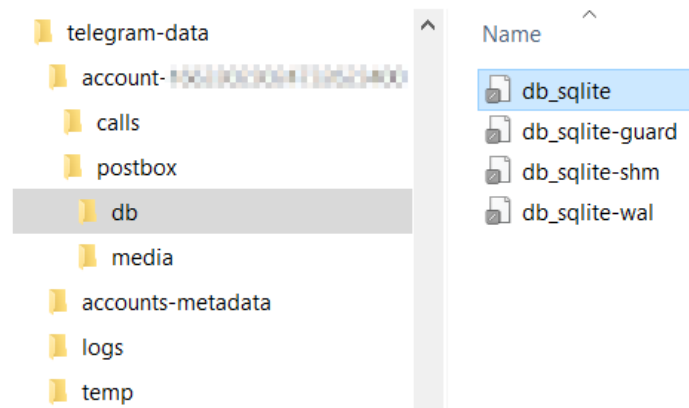


Source: Telegram
https://telegram.org/img/t_logo.png

All chats are stored locally!

Telegram on iOS smartphones

- October 2018: Replacement of the conventional Telegram Messenger on iOS with **Telegram X** → Fundamental structural differences
- Main directory: \private\var\mobile\Containers\Shared\AppGroup\<id>**telegram-data**
 - Subdirectory for each user account used on the device
 - **Main database** for each account: postbox/db/**db_sqlite**
 - **Media files** for each account: postbox/**media/**



Methods

I. Analysis of the Telegram functionalities

- Identification of forensically relevant functionalities
- Design of various test scenarios
 - Preparation and installation
 - Creating user profiles
 - Creating groups, channels and secret chats
 - Communication in chats of different types

II. Generation of test data of the Telegram Messenger under iOS

- Execution of test scenarios using an iPhone 5, iPhone 6 and iPhone SE
- After each scenario: extraction of the Telegram folder using **UFED Touch**

Methods

III. Analysis of the generated data

- Analysis of the main database using **DB Browser for SQLite**
- In-depth analysis at binary level using **HxD Hex Editor**
- **Understanding data by analysing the Telegram source code**
 - <https://github.com/TelegramMessenger/Telegram-iOS>
 - Localization & interpretation of relevant data from main database
- if necessary, new data generation (II.)

Methods

➤ <https://github.com/TelegramMessenger/Telegram-iOS>

The screenshot shows the GitHub repository page for TelegramMessenger/Telegram-iOS. The repository is public. The 'Notifications' button is highlighted with a red box. The '189 Tags' link is also highlighted with a red box. The repository has 514 issues, 39 pull requests, and 27,859 commits. The repository is maintained by laktyushin. The repository is tagged with 189 tags. The repository is tagged with 189 tags.

Results for the main database tables

- Table name: Composition of prefix 't' & identification number
- Two columns per table: **key** (integer or BLOB) & **value** (BLOB)
- BLOB = Binary Large Object → **data is encoded, not encrypted !**
- Most relevant tables:
 - **MetadataTable t0**: Metadata, in particular peer ID of the local user
 - **PeerTable t2**: Information about users & chats
 - **CachedPeerDataTable t18**: further information about users & chats
 - **MessageHistoryTable t7**: Information about messages, calls & media

Encoding of objects

- Mostly, coding of entire objects in BLOBs in **little-endian format**
- At the top level: **RootObject**
 - 1 byte key length + 1 byte key + 1 byte data type + 4 byte object class hash + 4 byte attribute length n + n byte encoded attributes

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	01	5F	05	2B	A5	68	9E	6E	01	00	00	01	69	01	28	DB	._.+¥hžn....i.(Û
00000010	0B	00	00	00	00	00	02	61	68	01	45	A1	95	5B	50	7D	ah.E;•[P}
00000020	89	87	03	61	68	74	00	00	00	00	00	02	66	6E	04	08	%‡.aht.....fn..
00000030	00	00	00	54	65	6C	65	67	72	61	6D	01	70	04	05	00	...Telegram.p...
00000040	00	00	34	32	37	37	37	02	70	68	08	02	00	00	00	7E	..42777.ph.....~

- For each class: specific **MurmurHash3 value** & attributes
 - e.g. 0x 2B A5 68 9E → TelegramUser

Code Insight: TelegramUser

release-11.1.1 Telegram-iOS / submodules / TelegramCore / Sources / SyncCore / SyncCore_TelegramUser.swift ↑ Top

Code Blame 487 lines (416 loc) · 22.2 KB Raw Copy Download

```
103 public final class TelegramUser: Peer, Equatable {
232
233     public init(decoder: PostboxDecoder) {
234         self.id = PeerId(decoder.decodeInt64ForKey("i", orElse: 0))
235
236         let accessHash: Int64 = decoder.decodeInt64ForKey("ah", orElse: 0)
237         let accessHashType: Int32 = decoder.decodeInt32ForKey("aht", orElse: 0)
238         if accessHash != 0 {
239             if accessHashType == 0 {
240                 self.accessHash = .personal(accessHash)
241             } else {
242                 self.accessHash = .genericPublic(accessHash)
243             }
244         } else {
245             self.accessHash = nil
246         }
247
248         self.firstName = decoder.decodeOptionalStringForKey("fn")
249         self.lastName = decoder.decodeOptionalStringForKey("ln")
250
251         self.username = decoder.decodeOptionalStringForKey("un")
252         self.phone = decoder.decodeOptionalStringForKey("p")
253     }
```

Symbols

Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

- class TelegramUser
 - func init
 - func init
 - func encode
 - func isEqual
 - func withUpdatedUsername
 - func withUpdatedUsernames
 - func withUpdatedNames
 - func withUpdatedPhone
 - func withUpdatedPhoto
 - func withUpdatedEmojiStatus
 - func withUpdatedFlags
 - func withUpdatedStoriesHidden

User & chat information from t2

- **key**: Integer (= Peer ID)
- **value**: BLOB in little-endian format
 - 4 possible classes for a RootObject in **t2**:
 - 0x 2B A5 68 9E → **TelegramUser**
 - 0x 51 7D A1 66 → **TelegramGroup**
 - 0x DA 11 6B 63 → **TelegramChannel**
 - 0x 5A 6E C9 21 → **TelegramSecretChat**

Table:  t2

	key	value
	Filter	Filter
1	777000	BLOB
2	136817688	BLOB
3	140267078	BLOB
4	557478932	BLOB
5	9814559261	BLOB
6	17026867968	BLOB
7	35471233848	BLOB
8	36513321142	BLOB
9	37209443782	BLOB
10	37414260450	BLOB

User & chat information from t18

- **key**: Integer (= Peer ID)
- **value**: BLOB in little-endian format
 - 4 possible classes for a RootObject in **t18**:
 - 0x 91 9D 1F 76 → **CachedUserData**
 - 0x 21 E8 2D A9 → **CachedGroupData**
 - 0x 55 15 C2 16 → **CachedChannelData**
 - 0x 4B D0 6F 4E → **CachedSecretChatData**

Table:  t18

	key	value
	Filter	Filter
1	557478932	BLOB
2	9814559261	BLOB
3	35471233848	BLOB
4	36513321142	BLOB
5	37209443782	BLOB
6	37414260450	BLOB
7	37639128123	BLOB
8	37738031735	BLOB
9	38911051632	BLOB

Message & call information from t7

- No root objects in BLOBs, but a **sequence of attribute values in a defined order**
- **key:** BLOB in big-endian format
 - Peer ID of the other chat instance, timestamp of the message or call, chat internal ID
- **value:** BLOB in little-endian format
 - For **messages**: Peer ID of the sender, status, textual content, media information, ...
 - For **calls**: Peer ID of the caller, status, duration, ...

Table:  t7		
	key	value
	Filter	Filter
1	BLOB	BLOB
2	BLOB	BLOB
3	BLOB	BLOB
4	BLOB	BLOB
5	BLOB	BLOB
6	BLOB	BLOB
7	BLOB	BLOB
8	BLOB	BLOB
9	BLOB	BLOB
10	BLOB	BLOB

Summary & future work

Summary

- Enabling the identification and interpretation of data about users, chats, messages, calls and media
 - Contact & communication network can be created by correlating the data
- **Challenge:** changing table structures due to Telegram updates

Future work

- Analysis of further database tables
- Further experiments to analyse deleted data
- Analysis of log files, e.g. to draw conclusions about deleted data

Thank you



**HOCHSCHULE
MITTWEIDA**
University of Applied Sciences

Lukas Jaeckel, M. Sc.
Doctoral Student
jaeckel1@hs-mittweida.de

Hochschule Mittweida | University of Applied Sciences
Technikumplatz 17 | 09648 Mittweida | GERMANY
Faculty Applied Computer Sciences & Biosciences

hs-mittweida.de/en