To Possess or Not to Possess - WhatsApp for Android Revisited with a Focus on Stickers

Samantha Klier(✉) and Harald Baier

Research Institute CODE, Universität der Bundeswehr München, Munich, Germany
{samantha.klier,harald.baier}@unibw.de
https://www.unibw.de/digfor

Abstract. WhatsApp stickers are a popular hybrid of images and emoticons that can contain user-created content. Stickers are mostly sent for legitimate reasons, but are also used to distribute illicit content such as Child Sexual Abuse Material (CSAM). As the process of creating stickers becomes easier for users from version to version, a digital forensic analysis is still lacking. Therefore, we present the first comprehensive digital forensic analysis of WhatsApp's sticker handling on Android, with a special focus on the legal context, i.e. the definition of possession of illicit content. Our analysis is based on 40 scenarios that reflect the full lifecycle of community-created stickers. We show how the distribution channel of a sticker found on a device can be reconstructed, partially even when its traces have been removed from WhatsApp and are not visible through WhatsApp's user interface. In addition, we show that Google Drive backups recover stickers, making device seizure dispensable; however, stickers can still be permanently deleted. Most importantly, we show that simply finding a sticker on a device is not sufficient to meet the requirements of the legal definition of possession. Therefore, prosecution for possession of a sticker requires additional evidence, which we provide.

Keywords: WhatsApp · sticker · possession · CSAM · digital forensics · Android

1 Introduction

WhatsApp by Meta is the most used messenger app globally [22] and the third most used Social Network [23]. In 2018 Meta introduced stickers to WhatsApp, which are a hybrid of images and emojis and were designed to "share your feelings in a way that you can't always express with words" in an "easy and fun" way [27]. Unlike emojis, stickers can be created by users based on any image, except for format restrictions. Back in 2018 sticker creation was complicated, as it involved the development of a dedicated sticker app [26]. Later, so-called sticker makers became popular, such as the "Sticker.ly - Sticker Maker" which has been downloaded more than 100 million times from Google's Play Store alone [24].

L. Fritsch et al. (Eds.): NordSec 2023, LNCS 14324, pp. 281–303, 2024.
https://doi.org/10.1007/978-3-031-47748-5_16

Not surprisingly, stickers are used to distribute not only memes, but also illegal content such as CSAM [5,14] and Nazi propaganda [21]. Even worse, WhatsApp automatically saves every sticker received in a chat, hence users may have incriminating stickers on their devices without knowing or wanting to. This sounds absurd at first, but the German police reported that CSAM stickers were posted to a group chat of climate activists ("Fridays for Future") [2] placing all members of the group in the precarious position to have CSAM on their devices, which can have serious consequences for the device owner, such as jail sentences, if, for example, a sticker depicts CSAM, the possession of which is illegal in 140 countries [12].

Right now, WhatsApp is rolling out the ability to create stickers directly in WhatsApp [25,28] based on any common image format, making the creation and sending of personalized stickers even easier. Therefore, a digital forensic understanding of stickers is of increasing importance. With this paper, we contribute a comprehensive digital forensic analysis of WhatsApp's Sticker handling, with particular attention to the legal definition of possession.

Contributions and Organization of Paper. First off, we introduce stickers from a user perspective and concentrate on their peer-to-peer nature and find that stickers can be collected but are otherwise concealed from a user's perception, in contrast to other media types commonly shared per WhatsApp (Sect. 2).

Importantly, we highlight the legal prerequisites that must be met to assume that a sticker file is possessed by a user, namely control, knowledge, and intent, as these are the drivers of a digital forensic examination (Sect. 3). Furthermore, we analyze and compare existing research on WhatsApp to put our work in perspective (Sect. 4).

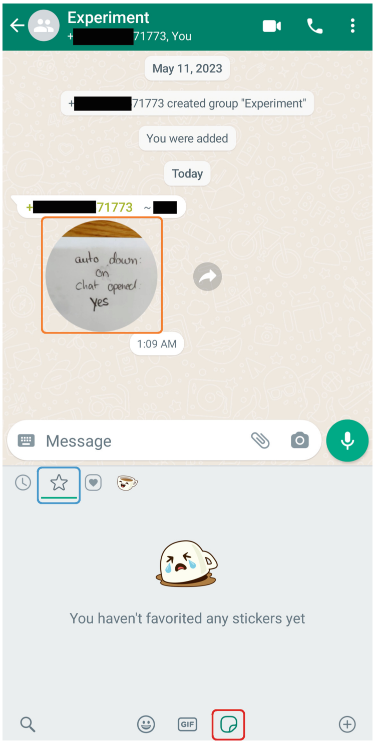
To examine the real evidential weight of artifacts, we design and evaluate 40 scenarios, which cover all possibilities a user has to interact with stickers. Our scenarios incorporate the areas of reception, interaction, removal, and backup (Sect. 5). Subsequently, we execute the scenarios on one physical and two emulated devices running Android 9, 10 and 13, as well as two different versions of WhatsApp, respectively. We base our evaluation on WhatsApp's directories in both the Android media and data partition, and reduce the data to eleven relevant artifacts.

Then, we present our fundamental results and, most importantly, reconstruct the communication from the stored sticker up to its origin (Sect. 6). Next, we identify the `msgstore.db` as the primary source for evidence of distribution, but find that the `stickers.db` and the `whatsapp.log` can be used to prove distribution if the data from the `msgstore.db` is not available (Sect. 7).

In contrast to distribution, proving possession is complex; therefore, we shed light on the aspect of control and knowledge (Sect. 8) and find that the mere existence of a sticker on a device is not sufficient to satisfy the legal requirements of possession. Consequently, we present artifacts that prove that a user had control over a file and knew it existed, namely the favorite and quoted messages in the `msgstore.db`, favorite stickers in the `stickers.db` and cached sticker files, which are only available when the user interacted with a sticker.

Coherently, we sum our findings per artifact in a conclusive table and find that without access to the data stored on the data partition there is no evidence that supports prosecution (Sect. 9) and finally conclude our paper (Sect. 10).

2 Sticker Foundations



Experiment

+[redacted]71773, You

May 11, 2023

+[redacted]71773 created group "Experiment"

You were added

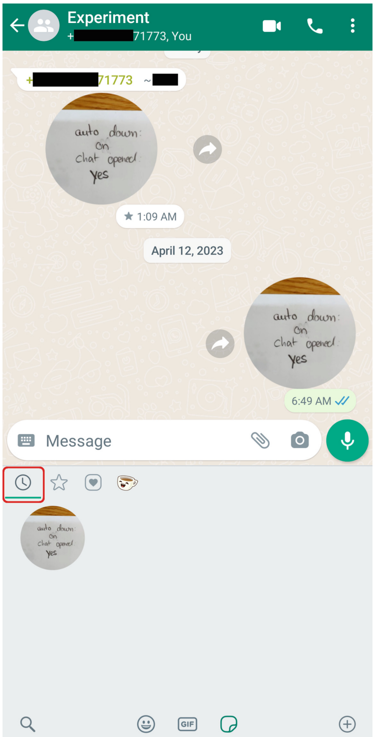
Today

+[redacted]71773 ~[redacted]

auto down: on
Chat opened: yes

1:09 AM

You haven't favorited any stickers yet



Experiment

+[redacted]71773, You

+[redacted]71773 ~[redacted]

auto down: on
Chat opened: yes

★ 1:09 AM

April 12, 2023

auto down: on
Chat opened: yes

6:49 AM ✓✓

Fig. 1. Receiving a sticker from a stranger who added the recipient to a group chat called "Experiment". (Color figure online)

Fig. 2. The user sent the sticker they had just received, which is recorded in the recent sticker menu (red box).(Color figure online)

First of all, we now introduce the foundations of WhatsApp sticker handling with a focus on the user experience, while we dismiss the technical details mostly to Sect. 6.

2.1 Receiving and Sending Stickers

Any user can receive stickers in a chat, as you can see in Fig. 1 in the orange box. To use stickers, a user can collect previously received stickers by clicking and selecting "mark as favorite" or the star button which will add the sticker to the users favorites menu, which is shown at the bottom of Fig. 1, and is activated by clicking first on the sticker button (red box) and then on the star button (blue box). In this example, the user has not yet "favorited" any sticker.

Our user now "favorited" the received sticker and can subsequently redistribute it, as can be seen in Fig. 2. The recipient responded with the sticker they had just received. Another way to obtain stickers is to install official sticker packs, which are available by clicking on the plus button in the bottom right corner. However, these sticker packs are beyond the scope of this paper, as they are subject to Meta's Terms of Service, and are therefore unlikely to contain content of interest to Law Enforcement Agencies (LEAs). Nevertheless, any sent sticker is recorded by an entry in the recently used stickers menu (red box).

2.2 Stickers are Different

This means that stickers can be distributed in a peer-to-peer-like manner, can be collected by users, and hence "going viral" is part of their design. This is very different from the behavior of emojis and GIFs (available with the buttons to the left of the sticker button), which serve a similar purpose but cannot be collected because the available content is the same for every user and is exclusively provided by the respective platform.

```
└com.whatsapp
  └WhatsApp
    └Media
      ├WhatsApp Animated Gifs
      │ └[…]
      ├WhatsApp Images
      │ └[…]
      ├WhatsApp Stickers
      │ └.nomedia
      └[…]
```

Fig. 3. Excerpt of data initially stored by WhatsApp in the user-accessible media partition.

However, just as GIFs, photos, and videos, stickers are stored locally on the device, but unlike these other media files, stickers are only partially affected by the settings for the automatic download of media files. By default the automatic download of media files is enabled but can be disabled in the settings, then, WhatsApp suggests that "no media" other than voice messages will be downloaded. This is misleading, as static stickers are downloaded nonetheless (see Sect. 8.1), therefore, a user can prohibit the download of any media file except for static stickers.

Every downloaded media file is stored in its dedicated directory in the user-accessible media partition, for which we show an excerpt in Fig. 3, and is presented to the user outside of WhatsApp's user interface, for example, in the device gallery, except for stickers. This is due to the fact that only the `WhatsApp Stickers` directory is marked with a `.nomedia` file, which signals Android's media scanner to ignore this directory and its content [7]. As a result, a user is never confronted with the fact that stickers are stored on their device, in contrast to other media files.

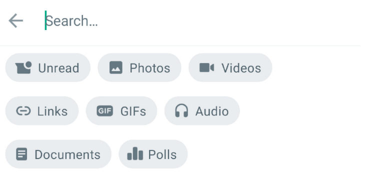


Fig. 4. WhatsApp's can search for, e.g. images and videos across all chats, but not for stickers.

However, stickers are also hidden from the user's perception in WhatsApp's user interface. For example, a user can get an overview of all photos, videos, and GIFs from all chats, which is impossible for stickers, as shown in Fig. 4. Therefore, apart from used and collected stickers, a user is never confronted with stickers that are stored on its device beyond the chat in which a sticker was received.

3 A Special Emphasis on Possession

The differences shown in sticker handling in contrast to the handling of other media files are a problem in light of the legal definition of possession. For example, the civil codes of European countries explicitly define possession as having *actual control* and sometimes even require specific *intent* [4]. On the contrary, common law legislation does not provide a precise definition of possession, although the concepts applied are similar [4,11]. The aspect of possession of files has already been argued in court proceedings, for example, when the only CSAM files found were in the browser cache [11,16].

In proceedings based on CSAM found in a browser cache, some defendants have admitted that they intentionally viewed CSAM on the Internet, but argued that they were unaware of the existence of a browser cache and never intended to possess these images. Consequently, they claimed that they had no control over these files and therefore did not possess them [11,16]. Although some courts have accepted this reasoning, others have contended that cached files merely serve as evidence of past possession while intentionally viewed CSAM was possessed in the form of the image displayed [11].

Consequently, Howard [11] aptly distinguished the two approaches followed by the courts and named them *Present Possession* approach and *Evidence Of* approach. The *Present Possession* approach assumes that the cached files are the possessed files, which can be circumvented by technical ignorance, e.g., the existence of the browser cache was unknown to the defendant. On the contrary, the *Evidence Of* approach expects the cached files to be the witness of a crime and, hence, cannot be circumvented by technical ignorance. However, it is hard to prove that an artifact effectively testifies that a crime was committed. For example, Horsman [9,10] showed that the existence of a file in a browser cache does not prove that it was actually displayed on the screen, let alone viewed by the user, and therefore has little evidentiary value.

In the rest of this paper, we follow the *Evidence of* approach and concentrate solely on technical facts that prove a defendant's capability to control a sticker and knowledge of its existence, summing up to *actual control*, and avoid jumping to conclusions or making assessments, which is the duty of the judge and jury [3].

4 Related Work

So far, no light has been shed on WhatsApp's sticker handling, although some specific WhatsApp functionalities have been studied by the community, such as the call signaling messages [13] or the security of group chats [20].

However, back in 2014 Anglano [1] established with his work a thorough understanding of WhatsApp's artifacts, which included not just the extraction of information but also correlation from several artifacts to reconstruct, e.g. deleted messages, but also the temporal context. Anglano focused on SQLite databases that were stored in the `/data/com.whatsapp/databases` directory of the data partition. Anglano identified, for example, the `msgstore.db` to contain exchanged messages. But, for the reconstruction of deleted information and the temporal context, Anglano incorporated also the log file of WhatsApp (i.e. `/data/com.whatsapp/files/Logs/whatsapp.log`). Fortunately, these artifacts are still relevant in newer versions of WhatsApp, which we show i.a. in Sect. 6.2.

Although the focus of Anglano [1] was not on media sharing, the exchange of multimedia files was briefly examined on the example of an image. However, at the time of the study, WhatsApp did not automatically download received images; instead, only thumbnails were displayed. As a result, identifying instances of incriminating images on a device always implied that users had manually downloaded and retained them. Thus, the question of possession of incriminating images did not arise. In addition, the study could not include an analysis of stickers as they had not yet been introduced at the time.

Furthermore, significant transformations have occurred in the realm of backups since the publication of Anglano [1]. Today, WhatsApp's backup strategy is using Google Drive by default, while additionally creating encrypted local copies of some files which were studied by Anglano [1]. However, back in 2014, the backups were encrypted with a universal encryption key that was publicly known for all users, whereas today these backups are encrypted using AES-256 and employ a unique key stored in Android's protected `data` partition [6]. Therefore, we focus on Google Drive backups which were not studied before, and only remark the potential existence of local encrypted backups for some databases (see Table 2, artifacts with ID 3-5).

5 Research Approach

We aim to find which artifacts effectively prove that a user had actual control over a sticker or distributed a sticker, therefore, we study the complete life cycle of stickers collected from peers which results in 40 scenarios, we divide into the areas of (I) reception, (II) interaction, (III) removal, and (IV) backups, respectively in the context of one-on-one and group chats. We now give a brief overview of our scenarios, the execution of the scenarios and our evaluation methodology. For a detailed and atomic description of our experiments, please refer to Table 4 in Appendix A.

5.1 Scenarios

Generally, our scenarios are created with the idea of comparing the resulting artifacts against each other, to determine their meaningfulness. For example, in the reception area, we have the scenario `RECEIVE-OFF-NODISPLAY` (I.4) and its direct counterpart `RECEIVE-OFF-DISPLAY` (I.3). In Fig. 5, we show the flow

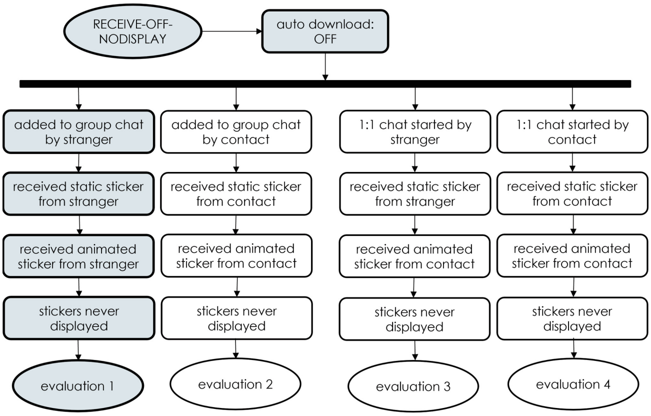

RECEIVE-OFF-NODISPLAY
auto download: OFF
added to group chat by stranger
added to group chat by contact
1:1 chat started by stranger
1:1 chat started by contact
received static sticker from stranger
received static sticker from contact
received static sticker from stranger
received static sticker from contact
received animated sticker from stranger
received animated sticker from contact
received animated sticker from contact
received animated sticker from contact
stickers never displayed
stickers never displayed
stickers never displayed
stickers never displayed
evaluation 1
evaluation 2
evaluation 3
evaluation 4

Fig. 5. Flow diagram of the `RECEIVE-OFF-NODISPLAY` scenario.

diagram of the scenario `RECEIVE-OFF-NODISPLAY` (I.4) which combines every possible characteristic of receiving a sticker, resulting in four subscenarios. The colored boxes highlight a subscenario that represents a case where a user disabled auto-download, was added to a group chat by a stranger, and received an animated and a static sticker that were never displayed. In contrast, the `RECEIVE-OFF-DISPLAY` scenario is identical except that the stickers were displayed because the user opened the chat after receiving them. These complementary scenarios allow us to determine whether there are artifacts that reliably indicate that a sticker was displayed.

5.2 Execution

In order to have a convenient full access to the file system, we use an emulated[1] Google Pixel 6 Pro smartphone, running Android 13, to execute each scenario. We installed and registered WhatsApp in version 2.23.7.76 with the official package installer[2]. Furthermore, we validate the results of the emulated device by executing key scenarios on a physical Samsung Galaxy S9+ (Android 9 with WhatsApp v2.22.23.84). To study whether different operating system versions and app versions have an impact on the artifacts, we executed some scenarios on an emulated Google Pixel 3a (Android 10 with WhatsApp v2.23.7.76). Please refer to Table 4 in the Appendix A for a detailed mapping of the devices to scenarios. We also provide the extracted data upon request with the assurance that the phone numbers and accounts involved will not be disclosed.

5.3 Evaluation

For our evaluation we consider artifacts stored in the WhatsApp directories in the media and the data partition, we acquired logically. Therefore, the recovery

[1] Using: Android Studio Electric Eel — 2022.1.1 Patch 2.

[2] https://www.whatsapp.com/android/?lang=en.

of deleted files, which requires a physical extraction, is beyond the scope of this work. Furthermore, we analyzed databases, by commonly connecting and querying them, therefore, we did not consider deleted records that may be recovered by analyzing their WAL files, which, however, is a discipline of its own [17–19].

Nevertheless, we acquired more data than is feasible for a manual analysis. For each scenario, more than 200 files have been extracted from WhatsApp's directories, including up to twelve databases. Two of these databases, namely `wa.db` and `msgstore.db`, which were already presented by Anglano [1], contained three tables in 2014, respectively. Now, `wa.db` and `msgstore.db` have 33 and 164 tables, respectively. Therefore, we applied a systematic and reproducible preprocessing step prior to the analysis phase to reduce and structure the data corpora to a manageable amount for review.

We identify relevant artifacts, by applying a recursive backward search, as proposed by Klier et al. [15], which starts from the known metadata of a file, and searches within artifacts for appearances of those metadata and consequently collects further metadata to search for, until no new metadata can be found. In this case, we start the search with the sticker's filename and its SHA-256 file hash in Base64 representation, based on the findings of Anglano [1]. The collected findings are saved to a JSON file, which allows a straightforward comparison of complementing scenarios and a starting point for an in-depth analysis.

Finally, in Table 1 and Table 2, we show all relevant artifacts we identified with our approach for sticker handling, including a self-assigned ID for further reference. Please note that square brackets are used to represent a naming scheme (for an example, see Sect. 6.1).

Table 1. Relevant artifacts of the media partition, located in `/media/...` (Android 9/10) or `/media/Android/media/com.whatsapp/...` (Android 13).

ID	Filename	Path	Type	Content
1	`STK-[YYYYMMDD]-WA[NNNN].webp`	`WhatsApp/Media/WhatsApp Stickers/`	WEBP	actual sticker file
2	`[Base64].thumb.webp`	`WhatsApp/.StickerThumbs/`	WEBP	animated sticker file preview

Table 2. Relevant artifacts of the data partition, for all Android versions located in `/data/data/com.whatsapp/...`.

ID	Filename	Path	Type	Content
3	`wa.db`	`databases/`	SQLite	contact & profile records
4	`msgstore.db`	`databases/`	SQLite	communication records
5	`stickers.db`	`databases/`	SQLite	data for sticker menu
6	`media.db`	`databases/`	SQLite	media download records
7	`whatsapp.log`	`files/Logs/`	text file	main log of WhatsApp
8	`whatsapp-[YYYY-MM-DD].log.gz`	`files/Logs/`	GNU zip	former versions of log
9	`[SHA256 in Base64].webp`	`files/Stickers/`	WEBP	copy of sticker file
10	`[SHA256 in Base64][RESOLUTION].0`	`cache/webp_static_cache/`	PNG	preview of sticker
11	`[SHA256 in Base64].tmp[RESOLUTION].0`	`cache/webp_static_cache/`	PNG	preview of sticker

6 Fundamental Results

We will now explain our fundamental results before addressing the legal questions at hand in the subsequent sections. Most importantly, each downloaded sticker file (ID `1`, in Table 1) is stored in the directory `/WhatsApp/Media /WhatsApp Stickers` which is either located in the root directory (Android 9/10) or in `/Android/media/com.whatsapp` (Android 13) of the media partition.

6.1 Sticker Files

Listing 6.1. The structure of a static sticker file with optional ICC profile and alpha channel information.

```
Chunk    |   Length |    Offset | Payload (excerpt)
 RIFF    |    21346 |         0 | WEBP
   VP8X |       10 |        12 | 8........
   ICCP |      536 |        30 | .........0..mntrRGB XYZ ...
   ALPH |     7164 |       574 | .!.m.F...._8I.BD.'.I. ...o>
   VP8  |    13218 |      7746 | p....*....>1..D"!..yu. ....
   EXIF |      374 |     20972 | II*.......AW..‘.......{"sti
```

The filenames of the sticker files adhere to the scheme: `STK-[YYYYMMDD] -WA[NNNN].webp`, whereas the date (`YYYYMMDD`) reflects the day of reception and is based on the device time, just like the file system timestamp. In addition, `NNNN` is a counter that starts at 0 and increments by 1 with each sticker received that day. However, each sticker, as identified by its SHA256 hash, is only saved once, regardless of the time it was received, how many times, or in which chats. However, this is only true when WhatsApp correctly references all stored stickers; for exceptions, see Sect. 8.1. Nevertheless, the filename reflects the circumstances of the very first time that sticker was received.

The stickers adhere to the WebP standard [8], which is a container format for media content and is based on the Resource Interchange File Format (RIFF). In Listing 6.1 we show the structure of a static sticker file. The `VP8X` chunk indicates which features are present in the given file, such as alpha channel information. Next, follow optional chunks for an ICC profile (indicated by `ICCP`) or the referenced alpha channel information (`ALPH`). Then, a `VP8` chunk follows which contains the actual image data to be displayed. On the contrary, animated stickers contain an additional chunk with animation information (`ANIM`) and carry the media content in multiple frames (each indicated by `ANMF`). Afterwards, optional metadata follows, whereas each examined sticker file concludes with a chunk of Exif information (`EXIF`).

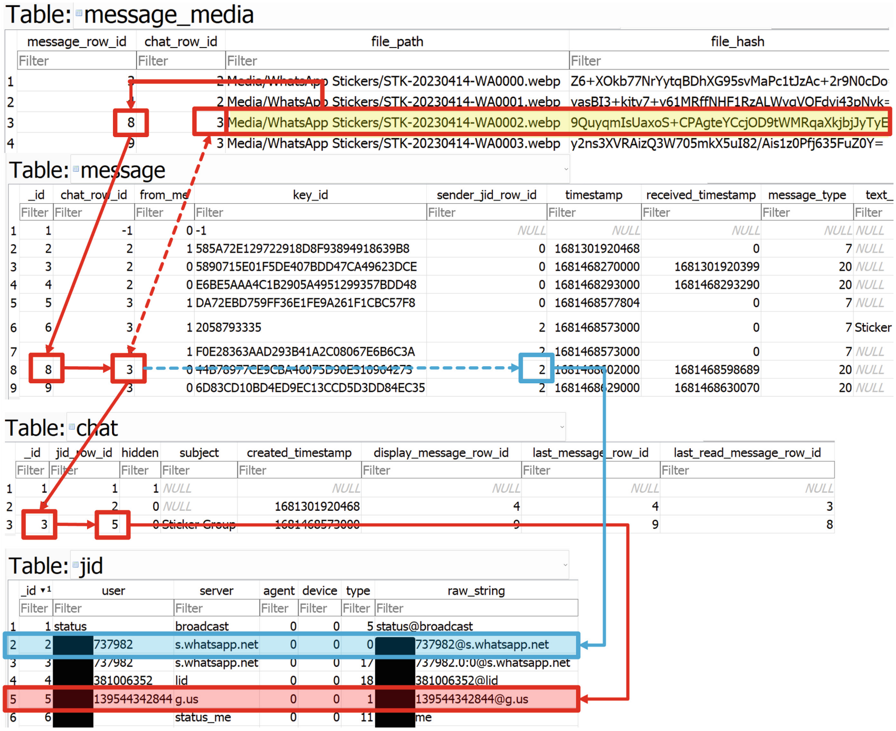


Table: message_media

	message_row_id	chat_row_id	file_path	file_hash
1	3	2	Media/WhatsApp Stickers/STK-20230414-WA0000.webp	Z6+XOkb77NrYytqBDhXG95svMaPc1tJzAc+2r9N0cDo
2	4	2	Media/WhatsApp Stickers/STK-20230414-WA0001.webp	vasBI3+kitv7+v61MRffNHF1RzALWvqVOFdvi43pNvk=
3	8	3	Media/WhatsApp Stickers/STK-20230414-WA0002.webp	9QuyqmIsUaxoS+CPAgteYCcjOD9tWMRqaXkjbjJyTyE
4	9	3	Media/WhatsApp Stickers/STK-20230414-WA0003.webp	y2ns3XVRAizQ3W705mkX5uI82/Ais1z0Pfj635FuZ0Y=

Table: message

	_id	chat_row_id	from_me	key_id	sender_jid_row_id	timestamp	received_timestamp	message_type	text_
1	1	-1	0	-1	NULL	NULL	NULL	NULL	NULL
2	2	2	1	585A72E129722918D8F93894918639B8	0	1681301920468	0	7	NULL
3	3	2	0	5890715E01F5DE407BDD47CA49623DCE	0	1681468270000	1681301920399	20	NULL
4	4	2	0	E6BE5AAA4C1B2905A4951299357BDD48	0	1681468293000	1681468293290	20	NULL
5	5	3	1	DA72EBD759FF36E1FE9A261F1CBC57F8	0	1681468577804	0	7	NULL
6	6	3	1	2058793335	2	1681468573000	0	7	Sticker
7	7	3	1	F0E28363AAD293B41A2C08067E6B6C3A	2	1681468573000	0	7	NULL
8	8	3	0	44B78977CE9CBA46075D90E310964273	2	1681468602000	1681468598689	20	NULL
9	9	3	0	6D83CD10BD4ED9EC13CCD5D3DD84EC35	2	1681468629000	1681468630070	20	NULL

Table: chat

	_id	jid_row_id	hidden	subject	created_timestamp	display_message_row_id	last_message_row_id	last_read_message_row_id
1	1	1	1	NULL	NULL	NULL	NULL	NULL
2	2	2	0	NULL	1681301920468	4	4	3
3	3	5	0	Sticker Group	1681468573000	9	9	8

Table: jid

	_id ▾1	user	server	agent	device	type	raw_string
1	1	status	broadcast	0	0	5	status@broadcast
2	2	[redacted]737982	s.whatsapp.net	0	0	0	[redacted]737982@s.whatsapp.net
3	3	[redacted]737982	s.whatsapp.net	0	0	17	[redacted]737982.0:0@s.whatsapp.net
4	4	[redacted]381006352	lid	0	0	18	[redacted]381006352@lid
5	5	[redacted]139544342844	g.us	0	0	1	[redacted]139544342844@g.us
6	6	[redacted]	status_me	0	0	11	[redacted]me

Fig. 6. Excerpts of `msgstore.db`, markings show the reconstruction of a sticker origin by the example of a group chat. (Color figure online)

6.2 Tracing the Sender

WhatsApp records communication in the `msgstore.db` (see `ID 4` in Table 2), including receiving and sending of stickers. In Fig. 6 we trace a sticker, as identified by its file hash and file path (highlighted yellow) in the `message_media` table, back to the originating chat (red markings). First of all, the column `message_row_id` references an entry in the `message` table which, in turn references an entry in the `chat` table. The `chat` table is also referenced by the `message_media` table, hence, this is redundant information. The `chat` table, most importantly, references an entry in the `jid` table by a `jid_row_id`, which finally points to the chat in which the sticker was received, in this case, this is a group with the identifier `******139544342844`.

In contrast, if the sticker had been received in a one-on-one chat, we would discover a user identifier and thus the sender. However, in this case we only identified a group. Now, to identify the concrete sender within the group, we need further information from the `message` table (blue markings), namely, the `sender_jid_row_id`, which again references the `jid` table. But this time the referenced entry contains the user identifier, which is actually the registered phone number (`******737982`), of the user who sent the sticker to the group, hence the sender is identified. In a one-on-one chat, the `sender_jid_row_id` is `0`, hence, invalid.

Furthermore, if the user of the device under investigation is the sender of the sticker in question, the value of the `from_me` column in the `message` table would be `1` while the `chat_row_id` would identify the recipient, again either a specific user or a group.

6.3 Timestamps Set by WhatsApp

WhatsApp uses several timestamps in its databases, in the `whatsapp.log` and for its file names (see Sect. 6.1). We found that the timestamps in the `whatsapp.log` reflect the local device time without stating the local timezone used and are in human readable format (similar ISO 8601). Furthermore, we can confirm that timestamps which indicate by their name to come from the server, e.g. `receipt_server_timestamp`, indeed reflect the server time in our experiments. On the contrary, any other timestamp reflects the local device time in `UTC+0`, which is consistent with the findings of Anglano [1].

6.4 Hints to the Origin

Information of a stickers origin and creation may be found in the embedded metadata, for example, in the `EXIF` tag (see Listing 6.1).

Each of the stickers studied contained a JSON in its `EXIF` tag, an example is shown in Listing 6.2 which is not altered by distribution. The embedded information is partly used to retrieve more stickers from the same source, e.g. by the `sticker-pack-id` or to manage stickers within WhatsApp, e.g. the `emojis` key is used to organize stickers by mood. In this case, the `sticker-pack-publisher` and the `android-app-store-link` point to the sticker maker that was used to create the sticker. The `sticker-pack-name` was assigned by us, in contrast, the `sticker-pack-id` was assigned by the sticker maker app used. However, while every sticker has such an embedded JSON, the actual available information differs tremendously, depending on a sticker's origin.

Listing 6.2. Exif metadata of a sticker.

```
{"sticker-pack-id": "stickerwhatsapp.com.stickers.stickercontentprovider afylruu",
"sticker-pack-name": "RECEIVE",
"sticker-pack-publisher": "Sticker Make for Wha[sic]",
"android-app-store-link": "https://play.google.com/store/apps/details?id=stickerwhatsapp.com.
    ↪ stickers",
"emojis": [[...]]}
```

6.5 Google Drive Backups

A Google Drive backup restores sticker files that were referenced in a chat and stored on the device at the time of backup, as well as data from the `/data/com.whatsapp/` directory, including databases, log files, and even cache files. Therefore, all of our findings can be applied to data restored from a Google

Drive back up, as even the original filenames of the stickers are recovered, and hence the filename still represents the circumstances of the very first reception before the backup. This means that it is possible that the filename of a sticker points to a date on which the device at hand was not yet in use. Consequently, a device can contain evidence of crimes committed with another device.

7 Evidence for Distribution

The distribution of a sticker by the user of the device under investigation can be proven with the msgstore.db database (ID 4, in Table 2), as shown in Sect. 6.2. In summary, the complete communication, including the time of distribution and the recipients, can be reconstructed, under the premise that the message_media table has a record for the searched sticker. However, these records are only available if the respective message or chat have not been deleted after the distribution. In this case, the distribution can be proven by resorting to WhatsApp's records of recently used stickers (see Fig. 2).

The recently sent stickers are recorded in the recent_stickers table of the stickers.db (artifact ID 5, in Table 2) in which a sticker can be identified by its file hash, referred to as plaintext_hash here. Furthermore, the timestamp of the last distribution is available in the last_sticker_sent_ts column, and the entry_weight column is actually a counter for executed send operations. However, a user can easily initiate the removal of these entries by deleting a sticker from the recent stickers menu (see Fig. 2). Fortunately, every insertion and removal from the recent_stickers table is recorded in whatsapp.log, as shown in Listing 7.1. Therefore, while the recipients of a sticker remain unclear, some distributions can be proven.

Listing 7.1. Excerpt of whatsapp.log that records the sending or forwarding of a sticker.

```
2023-04-12 12:29:41.552 [...] RecentStickers/addEntry/adding entry:
     ↪ WeightedRecentStickerIdentifier{stickerIdentifier=RecentStickerIdentifier{fileHash='
     ↪ ELkg[...]’, imageHash=’sqrI[...]’, sticker=Sticker{[...]}, weight=1.0}
[...]
2023-04-18 09:13:32.573 [...] RecentStickers/removeEntry/removing entry:
     ↪ RecentStickerIdentifier{fileHash=’ELkg[...]’, imageHash=’sqrI[...]’, sticker=Sticker
     ↪ {[...]}, lastStickerSentTs=1681302581878,[...]}
```

8 Evidence for Possession

While the evidence for distribution is unambiguous, the determination if a sticker in the WhatsApp Stickers directory is possessed by a defendant is complex; hence, we now discuss the aspects of control and knowledge of existence, identified as the main characteristics of possession in Sect. 3, in detail. While intention is also an integral component in some jurisdictions and must be considered in a prosecution, we will not discuss its aspects in this paper, as it is hardly technically.

8.1 Control

Cache Behavior. Although sticker files are stored in the media partition, they exhibit cache-like behavior. For example, stickers are automatically stored on the device upon receipt and are automatically deleted from the file system when they are no longer referenced in a chat. Additionally, a user has little option to affect this behavior. To be more precise, static stickers are downloaded even when the automatic media download has been disabled while the user cannot object to the automatic removal in any way, for example, when the sticker was part of a disappearing message. Consequently, sticker files are temporary and only reflect the state of the sticker within WhatsApp.

This is also reflected by the fact that stickers are downloaded and stored only once, regardless of how often they have been received. Therefore, storing the sticker files locally improves the user experience, e.g. due to offline displaying capabilities, reduced loading times and reduced data usage; hence, improve the apps performance which is the typical aim of caching. Furthermore, there is no indication that WhatsApp wants a user to handle sticker files, as they are treated differently than other media file types (see Sect. 2). To summarize, sticker files are temporary, improve the performance and are solely managed by WhatsApp, which is typical behavior for an application cache.

File System Access. Due to the full read and write privileges on the media partition, a user is capable to exercise full control over a sticker file in the file system once downloaded and before deleted. Most importantly, stickers deleted by the user in the file system are not automatically recovered by WhatsApp, not even when a backup is restored[3]. However, only technically skilled users can exercise this control over sticker files by manually browsing the file system with a file manager, as the respective storage area is not presented to the user, unlike, for example, the `WhatsApp Images` directory (see Sect. 2.2).

Control from WhatsApp. In turn, from within WhatsApp, every user can exercise three types of control over a sticker. First, a user can delete sticker files indirectly, yet reliably, from the `WhatsApp Stickers` directory[4] by any kind of removing operation offered, such as removing the received sticker from the chat, clearing the entire chat history, or blocking the contact. Second, a user is capable to redistribute stickers, as shown in Sect. 2. Third, users can control whether a sticker is added to their sticker menu by marking it as a favorite. These WhatsApp-specific options enable every user to control stickers to some extent.

[3] The deleted sticker in the respective chat is replaced by a button that allows the user to re-download a missing sticker.

[4] Under the condition that the sticker is not used in another chat.

Uninstalling and Migrating WhatsApp. In general, the files within the WhatsApp Stickers directory are in sync with the status of a sticker in WhatsApp. However, this linkage dissolves when WhatsApp is uninstalled on Android 9 and Android 10[5], while, the sticker files remain on the device. In this case, the only way a non-technically savvy user can exhibit control of a sticker diminishes. Even when WhatsApp is reinstalled and recovered from a backup the link to the stored sticker files is not recovered, in contrary, they are simply re-downloaded which leads unusually to several instances of identical stickers. The newly downloaded stickers again can be controlled from within WhatsApp, but not the old instances.

Although this effect appeared in our scenarios only on Android 9 and Android 10, sticker files on newer operating system may also be affected, as the device may have been upgraded from an older Android version which would migrate the sticker files to the newer operating system without restoring the ability to control these files.

Assessing Control for Prosecution. To sum up, although the sticker files are saved in a user-accessible storage area, there are arguments that contradict the assumption that the sticker files in the WhatsApp Stickers directory are under the control of the user. Therefore, the ascertainment that a sticker file is stored on a device is not sufficient and a digital forensic examination should bring those sticker files to light, which are accessible from within WhatsApp, and hence, evidently controllable by a user.

8.2 Knowledge of Existence

First of all, WhatsApp downloads sticker files to the WhatsApp Stickers directory, regardless of whether the chat has been opened by the user. Since the user is not confronted with sticker files in any other context, as shown in Sect. 2.2, an investigator cannot assume that a user knows that a sticker exists even when it is referenced in WhatsApp's user interface. Therefore, an examination must further verify the evidence of knowledge.

Distributing, Quoting and Starring. The best evidence for knowledge is the proof that a user actively engaged with a sticker, for example, by distributing, quoting, or starring. Therefore, the evidence of distribution that we presented in Sect. 7 can also be used to prove knowledge. However, stickers that were quoted, which is not a distribution, are handled slightly differently. To be more precise, they are recorded in message_quoted_media table of the msgstore.db instead of the message_media table while the rest of the communication can be reconstructed, as demonstrated in Sect. 7 and Fig. 6.

Additionally, the msgstore.db database (ID 4, in Table 2) records which sticker messages have been marked as favorites by setting the starred column

[5] WhatsApp on Android 13 asks the user if "keep app data" is desired. Irrespective of the users' choice on Android 13 the user stays in control of the sticker files.

of the `message` table to 1. Furthermore, the `stickers.db` database (ID 5, in Table 2) records favorite stickers in the `starred_stickers` table. Finally, every sticker that was sent, forwarded or marked as favorite is additionally stored in the `/data/com.whatsapp/files/verbStickers/` directory of the data partition, by the name of their file hash, e.g. `a0FxYpH-avyuN7RqWB+Zdz7Kd6DhyNWbc++cVy7xeoE=.webp` (ID 9, in Table 2). This is not the case for stickers with which the user has not been interacting with; hence, this also proves that the user was aware of the existence of a sticker.

Manual Download. Furthermore, a manual download of a sticker is also strong evidence that a user must know of the existence of a sticker. Fortunately, WhatsApp logs media downloads with their respective settings in the `whatsapp.log` log file (artifact ID 7, in Table 2), as shown in Listing 8.1 for a manually downloaded sticker. A manual download is indicated by `autoDownload=0`, `mode=manual` and `MediaDownloadManager/ start manual download`, consequently, automatically downloaded stickers are logged with `autoDownload=1`, `mode=auto` and `MediaDownloadManager/ queueDownload auto download`. Therefore, the log records the actual mode that was used to download a sticker and not the state of the auto-download setting; hence, actually proves a manual download.

Listing 8.1. Excerpt of whatsapp.log that indicates an automatic download of a sticker as identified by its hash.

```
2023-04-18 11:41:02.691 [...] MediaDownload/initialized;mediaHash=fSxk[...] autoDownload=0
[...]
2023-04-18 11:41:02.694 [...] MediaDownloadManager/start manual download [...], message.
      ↪ mediaHash=fSxk[...]
[...]
2023-04-18 11:41:03.004 [...] MediaDownload/updateMessageAfterDownload/mediaHash=fSxk[...]
      ↪ url=https://157.240.223.60/[...]&mode=manual status=success
```

Evidence for Display. While the display of a sticker may indicate knowledge, it is rather weak evidence, as a user must not see everything that was displayed, e.g. when scrolling to the end of a conversation rapidly. However, the display can be proven, as WhatsApp creates thumbnails of stickers when displaying them in a chat or in the notification bar. These thumbnails are stored in the data partition in the `/data/com.whatsapp/cache/webp_static_cache` directory (ID 10 and 11 in Table 2). The thumbnail file name contains the resolution (i.e. 64x64px) and the SHA256 hash of the actual sticker encoded in Base64, e.g. $\backslash \texttt{Z6} + \texttt{X0kb77NrYytqBDhXG95svMaPc1tJzAc} + \texttt{2r9N0cDo} = \texttt{.tmp\_64\_64.0}$.
The existence of a thumbnail with a resolution of `64x64px` indicates that the sticker was shown in the notification bar whereas a thumbnail with a resolution of `438x438px` was displayed in a chat.

9 Summary

We strongly advise that a prosecution should not be based exclusively on the existence of an incriminating sticker file in the WhatsApp Stickers directory, as this neither proves knowledge nor control. Therefore, in Table 3 we sum up our findings for each relevant artifact with respect to the evidence available under a given premise and, hence, open up the opportunity to prosecute stickers profoundly.

However, all artifacts that can prove distribution, knowledge, or control are stored in Android's data partition, which may not be acquirable in an investigation. This issue can at least partially be circumvented with a live examination, as the contents of the msgstore.db database and the stickers.db database, are reflected in WhatsApp's user interface.

Table 3. Summary of artifacts, incl. ID and evidence contained for Distribution, Knowledge and Control.

ID	Name	Premise	Evidence of...		
			Dis.	Knowl.	Cont.
1	STK-[YYYYMMDD]-WA[NNNN].webp	-	×	×	×
2	[Base64].thumb.webp	-	×	×	×
3	wa.db	-	×	×	×
4	msgstore.db: message_media	hash & path identical	×	×	✓
4	msgstore.db: message_media	hash & path identical, from_me=1	✓	✓	✓
4	msgstore.db: message_quoted_media	hash & path identical	×	✓	✓
4	msgstore.db: message	starred=1	×	✓	✓
5	stickers.db: recent_stickers	hash identical	✓	✓	✓
5	stickers.db: starred_stickers	hash identical	×	✓	✓
6	media.db	-	×	×	×
7	whatsapp.log	hash in RecentStickers/[add|remove]Entry	✓	✓	✓
8	whatsapp-[YYYY-MM-DD].log.gz	hash in RecentStickers/[add|remove]Entry	✓	✓	✓
9	[SHA256 in Base64].webp	hash in file name	×	✓	✓
10	[SHA256 in Base64].=.tmp_64_64.0	hash in file name	×	see Sect. 8.2	✓
10	[SHA256 in Base64]_438_438.0	hash in file name	×	see Sect. 8.2	✓

10 Conclusion

WhatsApp and stickers are a widespread and popular way to communicate. However, they can be used to distribute incriminated files and, hence, are in the focus of LEAs. Our study shows that a user can have stickers on its device without knowing or wanting to. For example, as a member of a innocuous group which is muted and rarely read a user has no option to prevent the unaware and automatic storage of a sticker on its device. Therefore, the mere existence of an incriminated sticker does not satisfy the prerequisites for possession. However, in such cases we show that deleting the sticker from the chat and from the file system is sufficient to effectively extirpate the incriminated content.

Furthermore, to hold offenders accountable, we identified evidence that proves distribution, knowledge, or control, even when the respective information has been removed from WhatsApp's user interface. However, our results show that a logical acquisition of an Android device is insufficient for prosecution in any case, whereas a live examination can be used in case access to the data partition is not available. On the other hand, we show that seizing a device is not effective to deny an offender access to the incriminated material as stickers can be immediately restored from a Google Drive backup, hence, the access to the backup must be prohibited, as well.

Overall, we showed that the concept of possession presents several intricate challenges that require careful examination during a digital investigation. While the possession of digital files when they are stored in an application's working directory has concerned courts for a long time, there is little digital forensic research on the topic, especially beyond browser caches. This is concerning, as smartphones and their apps are an important part of most peoples lives today, and, while it is the duty of the judge and the jury to evaluate if incriminated files are possessed, it is the duty of us to deliver the facts necessary to make the evaluation.

A Detailed Description of Executed Scenarios

Table 4. Scenarios executed for this paper.

ID	scenarios	chat types	contact status	auto downl.	sticker types	devices	actions
I.1	RECEIVE-ON-DISPLAY	1:1, group	contact, stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a, Galaxy S9+	sticker received, chat opened
I.2	RECEIVE-ON-NODISPLAY	1:1, group	contact, stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a, Galaxy S9+	sticker received, chat unopened
I.3	RECEIVE-OFF-DISPLAY	1:1, group	contact, stranger	OFF	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a	sticker received, chat opened
I.4	RECEIVE-OFF-NODISPLAY	1:1, group	contact, stranger	OFF	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a	sticker received, chat unopened
I.5	RECEIVE-TWICE	1:1, group	contact, stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	same sticker received twice in one chat
I.6	RECEIVE-TWICE-CROSSCHAT	1:1, group	contact, stranger	ON	3rd-party	(virt.) Pixel 6a Pro	same sticker received in one group chat and in a 1:1 chat
I.7	MARK-AS-READ-NODISPLAY	1:1	contact, stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker received automatically, chat unopened, chat marked as read
I.8	MARK-AS-READ-DISPLAY	1:1	contact, stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker received automatically, chat unopened, chat marked as read, chat opened
I.9	WRONG-DEVICE-TIME	1:1	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker received, system time set to past, sticker received, system time set to future, sticker received
II.1	INTERACT-REPLY	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	reply to received sticker
II.2	INTERACT-FAVORITE	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	mark received sticker as favorite

(*continued*)

Table 4. (*continued*)

II.3	INTERACT-FORWARD	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	forward received sticker
II.4	INTERACT-SEND	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	mark received sticker as favorite (prerequ. for sending of received stickers) than sent by receiver, than sent again (repeat 5 times)
III.1	DELETE-FROM-CHAT	1:1, group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a, Galaxy S9+	sticker received, sticker message deleted in chat
III.2	DELETE-CHAT	1:1, group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 3a, Galaxy S9+	sticker received, complete chat deleted
ID	scenarios	chat types	contact status	auto downl.	sticker types	devices	actions
III.3	CLEAR-CHAT	1:1, group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker received, chat cleared
III.4	DELETE-FROM-FS	1:1, group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker received, sticker deleted from file-system, +analysis of Google Drive BU (see IV.5)
III.5	DELETE-ONE-OF-TWO	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	same sticker received twice in one chat, one of two sticker messages deleted
III.6	DEL-ONE-OF-TWO-CROSS	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	same sticker received in one group chat and in one 1:1 chat, one of two sticker messages deleted
III.7	DELETE-BY-ADMIN	group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	sticker received, sticker deleted by admin for all group members

(*continued*)

Table 4. (*continued*)

III.8	DISAPPEARING	group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker sent to a group which messages will disappear after 24 hours
III.9	DISAPPEARING-AFTER	group	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro	sticker sent to a group which messages disappeared after 24 hours
III.10	DELETE-REPLY-CHAT	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	reply to received sticker, delete all traces of sticker from chat
III.11	DELETE-FAV-CHAT	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	mark received sticker as favorite, delete all traces of sticker from chat
III.12	DELETE-FWD-CHAT	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	forward received sticker, delete all traces of sticker from chat
III.13	DELETE-SEND-CHAT	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	mark received sticker as favorite (prerequ. for sending of received stickers) than sent by receiver, delete all traces of sticker from chat
III.14	DELETE-FAV-MENU	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	mark received sticker as favorite, delete from favorites menu
III.15	DELETE-FWD-MENU	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	forward received sticker, delete from recent menu
III.16	DELETE-SEND-MENU	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	mark received sticker as favorite (prerequ. for sending of received stickers) than sent by receiver, delete sticker from favorites and recent menu
III.17	DELETE-FAV-COMPLETE	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	mark received sticker as favorite, delete all traces of sticker from chat and favorites menu
III.18	DELETE-FWD-COMPLETE	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	forward received sticker, delete all traces of sticker from chat and recent menu

(*continued*)

Table 4. (*continued*)

III.19	DELETE-SEND-COMPLETE	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	mark received sticker as favorite (prerequ. for sending of received stickers) than sent by receiver, delete all traces of sticker from chat, favorites and recent menu
III.20	BLOCK-CONTACT	1:1	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	sticker received, sender blocked
IV.1	UNINSTALL-KEEP	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	sticker received, WhatsApp uninstalled (check "keep app data")
IV.2	UNINSTALL-NOKEEP	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro	sticker received, WhatsApp uninstalled (default, "keep app data" unchecked)
IV.3	UNINSTALL	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 3a, Galaxy S9+	sticker received, WhatsApp uninstalled (no uninstall options available)
IV.4	REINSTALL-RECEIVE	1:1, group	stranger	ON	3rd-party	(virt.) Pixel 6a Pro, (virt.) Pixel 3a, Galaxy S9+	sticker received, WhatsApp uninstalled (each uninstall options available), WhatsApp reinstalled, sticker re-received
IV.5	DRIVE-BACKUP	1:1	stranger	ON	3rd-party	(virt.) Pixel 6a Pro, (virt.) Pixel 3a	sticker received, chat unopened, backed up to Google Drive, default uninstall and re-install of WhatsApp, Drive back up restored
IV.6	DRIVE-BU-DELETE-CHAT	1:1	stranger	ON	3rd-party	(virt.) Pixel 6a Pro, (virt.) Pixel 3a	sticker received, chat unopened, chat deleted, backed up to Google Drive, default uninstall and re-install of WhatsApp, Drive back up restored
IV.7	DRIVE-BU-RERECEIVE	1:1	stranger	ON	3rd-party, 1st-party anim.	(virt.) Pixel 6a Pro, (virt.) Pixel 3a	sticker received, chat unopened, backed up to Google Drive, default uninstall and re-install of WhatsApp, Drive back up restored, same sticker received again

References

1. Anglano, C.: Forensic analysis of whatsapp messenger on android smartphones. Digit. Investig. **11**(3), 201–213 (2014)
2. Baden-Württemberg, L.: Strafbare inhalte bei whatsapp und co. Technical report, Polizei Baden-Württemberg (2019)
3. Casey, E.: Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. Academic Press, Cambridge (2011)
4. Chang, Y.C.: Law and Economics of Possession. Cambridge University Press, Cambridge (2015)
5. Europol: Operation chemosh: how encrypted chat groups exchanged emoji "stickers" of child sexual abuse (2019)
6. Fayyad-Kazan, H., Kassem-Moussa, S., Hejase, H.J., Hejase, A.J.: Forensic analysis of whatsapp sqlite databases on the unrooted android phones. HighTech Innov. J. **3**(2), 175–195 (2022)
7. Google: Mediastore (2023). https://developer.android.com/reference/android/provider/MediaStore#MEDIA_IGNORE_FILENAME
8. Google: Webp container specification (2023). https://developers.google.com/speed/webp/docs/riff_container
9. Horsman, G.: I didn't see that! an examination of internet browser cache behaviour following website visits. Digit. Investig. **25**, 105–113 (2018)
10. Horsman, G.: Reconstructing streamed video content: a case study on youtube and facebook live stream content in the chrome web browser cache. Digit. Investig. **26**, S30–S37 (2018)
11. Howard, T.E.: Don't cache out your case: Prosecuting child pornograpy possession laws based on images located in temporary internet files. Berkeley Tech. LJ **19**, 1227 (2004)
12. International Centre for Missing & Exploited Children: Child sexual abuse material: Model legislation & global review (2018)
13. Karpisek, F., Baggili, I., Breitinger, F.: Whatsapp network forensics: decrypting and understanding the whatsapp call signaling messages. Digit. Investig. **15**, 110–118 (2015)
14. van Kesteren, M., van Eeten, M., van Wegberg, R.: CSAM data - factcheck of recent European commission statements (2023)
15. Klier, S., Varenkamp, J., Baier, H.: Back and forth – on automatic exposure of origin and dissemination of files on windows. Digit. Threats Res. Pract. (2023). https://doi.org/10.1145/3609232
16. Marin, G.: Possession of child pornography: should you be convicted when the computer cache does the saving for you. Fla. L. Rev. **60**, 1205 (2008)
17. Meng, C., Baier, H.: bring2lite: a structural concept and tool for forensic data analysis and recovery of deleted sqlite records. Digit. Investig. **29**, S31–S41 (2019)
18. Nemetz, S., Schmitt, S., Freiling, F.: A standardized corpus for sqlite database forensics. Digit. Investig. **24**, S121–S130 (2018)
19. Pawlaszczyk, D., Hummert, C.: Making the invisible visible-techniques for recovering deleted sqlite data records. Int. J. Cyber Forensics Adv. Threat Investig. **1**(1–3), 27–41 (2021)
20. Rösler, P., Mainka, C., Schwenk, J.: More is less: on the end-to-end security of group chats in signal, whatsapp, and threema. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 415–429. IEEE (2018)

21. Schmehl, K.: Whatsapp has become a hotbed for spreading nazi propaganda in Germany (2019). https://www.buzzfeednews.com/article/karstenschmehl/whatsapp-groups-nazi-symbol-stickers-germany. Accessed 28 May 2023
22. Statista: Most popular global mobile messenger apps as of January 2022, based on number of monthly active users (2022). https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/
23. Statista: Most popular social networks worldwide as of January 2023, ranked by number of monthly active users (2023). https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/
24. Sticker.ly: Sticker.ly - sticker maker (2023). https://play.google.com/store/apps/details?id=com.snowcorp.stickerly.android. Accessed 30 June 2023
25. TcitNews: Whatsapp expanding features with in-app sticker creation capability (2023). https://tcitnews.com/whatsapp-expanding-features-with-in-app-sticker-creation-capability/. Accessed 10 July 2023
26. WhatsApp: How to create stickers for whatsapp (2018). https://faq.whatsapp.com/1056840314992666/. Accessed 29 Mar 2023
27. WhatsApp: Introducing stickers (2018). https://blog.whatsapp.com/introducing-stickers?lang=en
28. YouTube: How to create your own whatsapp stickers with iphone — whatsapp sticker new update (2023). https://www.youtube.com/watch?v=0UG-JDt0-1o. Accessed 10 July 2023