electronics

Article

Forensic Analysis of TikTok Alternatives on Android and iOS Devices: Byte, Dubsmash, and Triller

Yansi Keim *, Shinelle Hutchinson *, Apoorva Shrivastava and Umit Karabiyik *

Department of Computer and Information Technology, Purdue University, West Lafayette, IN 47907, USA
* Correspondence: ykeim@purdue.edu (Y.K.); hutchi50@purdue.edu (S.H.); umit@purdue.edu (U.K.)

Abstract: TikTok has consistently been one of the most used mobile apps worldwide on any mobile operating system. However, despite people's enjoyment of using the application, there have been growing concerns about the application's origins and alleged privacy violations. These allegations have become such a big problem that the former President of the United States, Donald Trump, expressed a desire to ban the TikTok application from being offered on US application stores like Google's Play Store and Apple's App Store. This remark sent TikTok users into a frenzy to find alternatives before the ban took effect. To this end, several alternative applications for TikTok have surfaced and are already garnering millions of users. In this paper, we identified three popular alternatives to the TikTok application (Byte, Dubmash, and Triller) and forensically analyzed each on smartphones of Android version 8 and iOS version 13. We focused on identifying forensically relevant artifacts that may be helpful to investigators in the event of a criminal investigation, should these or similar apps fall under scrutiny. We used Magnet AXIOM Process and Cellebrite UFED 4PC for acquisition, and Magnet AXIOM Examine and DB Browser for SQLite for analysis and reading. The investigation resulted in successful extraction of expected yet unique data points, plain text sensitive data, directories and format. These results lead to a discussion about identifying and comparing these app's privacy concerns to that of TikTok, as formulated from the literature.

Keywords: Android forensics; Byte; forensic analysis; iOS forensics; mobile forensics; Dubsmash; social networking forensics; Triller

Citation: Keim, Y.; Hutchinson, S.; Shrivastava, A.; Karabiyik, U. Forensic Analysis of TikTok Alternatives on Android and iOS Devices: Byte, Dubsmash, and Triller. *Electronics* **2022**, *11*, 2972. https://doi.org/10.3390/electronics11182972

Academic Editor: Cheonshik Kim

Received: 18 August 2022
Accepted: 14 September 2022
Published: 19 September 2022

1. Introduction

In 2020, when news surfaced that the United States government planned to ban the popular Chinese application, TikTok, its users began considering other alternative video-sharing apps available on Apple's App Store and Google's Play Store. The TikTok app has served as a popular video-sharing platform with over 100 million active monthly users in America [1], and over 1 billion installations from the Google Play Store [2]. Popular features of the app include creating, sharing, and viewing content based on lip-syncing, dancing, comedy skits, and other physical activities [1]. Their in-app camera features support colored filters, stickers, makeup, and voice changers [3]. Nonetheless, being labeled a "national security concern" by multiple nations, including India and the United States, there was a brief moment of uncertainty regarding TikTok's future in America. Despite the huge number of followers and popularity, TikTok has recently lost many followers. One of the biggest losses of users was caused by the Indian government's ban on the app due to political tension between China and India. India's ban on TikTok opened the door for other countries to consider implementing similar decisions. The United States government has expressed interest in banning the app for national security, whereas Australia and the European Union [4] have called for investigations to be carried out.

Consequently, there has been a surge in apps with similar features. Some of the popular TikTok alternatives are (1) Byte—Video communities, (2) Dubsmash—Create & Watch Videos, and (3) Triller: Social Video Platform, henceforth referred to as Byte,

Dubsmash, and Triller, respectively. These apps have garnered a large following in recent times, and as of February 2022, Byte (now Clash) has 1 M+ downloads, Dubsmash (now acquired by Reddit) has 100 M+ downloads and 1 billion video views per month [5], and Triller has 10 M+ downloads on the Play Store. These video-sharing apps fall within the social media sphere and attract hundreds of millions of users who, on average, spend 2 h and 24 min per day on social networks [6].

However, not all time spent on video-sharing apps is done by benign users, as several sexual assaults and sex trafficking incidents took place in Bangladesh when the victims were lured with shooting for TikTok and Likee videos [7]. Furthermore, TikTok and other video-sharing apps have been used to share pornography and even became popular with less savory users, including child predators [8]. These incidents are extremely worrying, as 32.5% of TikTok's 100 million active US users are children between the ages of 10 and 19 [1]. It has also been observed that videos with tags like ACAB, Transgender, and Gay have been shadow-banned [9] in many different countries and languages. This serves as an issue because a great majority of the users on TikTok are part of Generation Z, the most progressive [10] and outspoken generation. Most of these political discussions are done on social media platforms. If the topics that most of Generation Z discuss are shadow-banned, they will seek another platform to continue the discussion. If TikTok continues to shadow-ban relevant political tags, Generation Z may abandon the app in favor of another app with less censorship. Many entrepreneurs have realized this and created similar apps in hopes of replacing TikTok.

As such, various forensic investigations involving the TikTok app are present on various platforms [11–14]. Despite the recent surge in the installation and use of TikTok alternatives such as Byte, Dubsmash, and Triller, these newly popular apps are still not investigated. These growing statistics certainly raise concerns among mobile forensics researchers. These three apps may not require money when being downloaded and used, but the users might be paying with a loss of privacy, in both a physical and data aspect. To the best of our knowledge, no current research focuses on the forensic analysis of these apps on any mobile platform. Hence, we aim to fill this gap and identify the security and privacy issues within these apps on Android and iOS platforms. Relevant artifacts recovered during the examination are methodologically reported for future reference.

In this research, we forensically investigate the Byte, Dubsmash, and Triller mobile apps to answer the following research questions:

1. What forensically relevant data can be recovered from these apps?
2. What are some of the privacy concerns associated with these apps?
3. What sensitive user data do these apps collect?

Our contributions to this research are as follows:

- Conducted a forensic analysis of the Byte, Dubsmash, and Triller apps on Android 10 and iOS 13.3.1, which increases understanding and highlights what data these apps can access.
- Identified the locations of relevant artifacts pertaining to a mobile forensic investigation.
- Compared the privacy concerns of these apps with those of TikTok, previously identified in the literature.
- Increased understanding of how Magnet AXIOM mobile forensic investigation tool can be applied to moderately-known apps.
- Provided a forensic investigative framework for investigators who may encounter these or similar apps in future cases.

The remainder of this paper is organized as follows. Section 2 provides background information on Mobile Forensics of social networking and video sharing applications. In Section 3, we present the detailed methodology we followed to complete this study and present our findings in Section 4. In Section 5, we discuss the relevance of our findings and conclude our work in Section 6.

2. Literature Review

The mobile apps (Byte, Dubsmash and Triller) we investigate in this research follow a similar structure to that of TikTok. Therefore, it is essential to understand the forensics research previously done on TikTok. However, due to the limited search results, we extended our review to those apps that are popular in use for their similar video-sharing and messaging features. In the following paragraphs, we indicate their findings, followed by the gap or the scope of future research in these articles.

In [11], the authors examined artifacts left on Android devices by the TikTok app, which could be recovered and analyzed in the event of a forensic investigation. In particular, the authors were interested in discovering how to acquire user data, the content of those data, which accounts the user followed, which accounts followed them back, timestamps associated with these events, and with whom the user communicated. The method used to conduct this experiment started with first installing TikTok on a rooted Nox Player running Android 5.1.1. The data were extracted using the Android Debugging Bridge (ADB), and the files were read using the DB SQLite Browser. The results yielded a valid methodology for recovering TikTok artifacts that investigators could use to gather evidence. Future work should address this process on iOS mobile devices, as there may be differences in the data storage format and artifact locations.

A similar investigation on the TikTok Android app has been done by Domingues et al. [12], who conducted a postmortem forensic analysis following a similar methodology that details databases and XML files containing relevant artifacts. Their research complements existing works by extending it to more diverse environments, as they verify the validity of formerly established results across various TikTok app versions on different Android OS versions. The different environments tested did not show significant differences in the artifacts collected, except for the video cache, where the latest version was encoded differently and had an additional directory. The study yielded valuable data such as the accounts with which the user interacted, the messages exchanged, and so on. However, much of TikTok's inner data is kept on the cloud, which could not be accessed under the scope of this postmortem forensic analysis.

In a Chrome-based TikTok application running on Windows 10, Pandela T. and Riadi I. in [15] discuss the data population and recovery of artifacts generated via this web browser. They used popular tools like FTK Imager, Browser History Capture/Viewer, and Video Cache Viewer to assist their findings. While 80% of data was successfully recovered, such as text, caption content, the usernames of the suspect and victim, the profile photos of the suspect and victim, video photo thumbnail, and the source link from Tiktok the suspect accessed, they failed at recovering videos.

Like TikTok, many video-sharing apps also include a direct messaging feature; one such example is the Kik Messenger app. The forensic analysis of the Kik Messenger app on iOS devices done by Ovens et al. [16] aimed to find where Kik artifacts are stored on iOS devices and document them. The authors also had a secondary goal: interpret the artifacts to answer the questions typically asked during a forensic investigation. The analysis was conducted on three iPad devices and began by factory resetting and jailbreaking the iPads. The recovered artifacts were then queried to clarify what had been created or edited. Finally, the artifacts were manually analyzed. The results showed a range of recoverable artifacts, such as deleted messages and who communicated with whom. This app, in particular, showed little effort to verify the users' identities, which, while convenient for the privacy and anonymity of the users, may prove problematic for investigators.

Similarly, Jadhav Bhatt et al. [17] conducted a network forensic analysis of iOS social networking and messaging apps to understand the types of user data that the apps were sending and to study the runtime behavior of these apps, drawing attention to the lesser known security flaws of many of the apps studied. The methodology involves using Charles proxy and Wireshark on a Mac workstation to capture the communication between an iPad running iOS 11.2.6 and the network and analyzing the traffic on a Windows workstation. The results show that only a small percent of the apps studied encrypt their data. In contrast,

the others captured a lot of sensitive information like unencrypted geo-coordinates, text messages, URLs retrieving server-side contents, multimedia content (such as profile images of users), device information, calling information, email IDs/passwords, social networking credentials, and information shared to third-party domains. While their work covers iOS forensic analysis in depth, the experiments only involved free apps, and further work should be extended to paid apps.

User privacy is a critical concern that must be addressed by mobile app developers, especially when these apps are targeted for use by children who may not be aware of the risks associated with transmitting sensitive information online. Basu et al. [18] address the drawbacks of existing methods to detect the compliance of apps with privacy regulations by proposing a hardware performance counter (HPC) based model titled Children's Online Privacy Protection Act (COPPA) with a compliance detection accuracy rate greater than 99%. The methodology for HPC-based analysis involves creating an app corpus of both compliant and non-compliant apps, collecting their HPC data on the smartphone, using unsupervised machine learning to create labels for sample data followed by supervised learning to create a COPPA classifier, and storing the model in the smartphone and running test apps to predict COPPA compliance. The model was also tested on apps that aren't required to be COPPA compliant to detect transmission of advertising ID, Android ID, and device description, which were all found to be transmitted by FaceApp, TikTok, Facebook Lite, Uber, Zillow, and LinkedIn. However, this study does not analyze iOS apps, and the smartphone processor used is also old, so scalability to modern processors has not been verified.

Salamh et al. [19] investigated the privacy and security concerns in over 27 Android and iOS applications, including TikTok. The authors highlight that plain text messages sent via the app, the in-app user's uploaded videos, and usage activity history can all be recovered from Android and iOS platforms. We expect similar findings during our investigation as we follow a similar forensic methodology as [19]. Hutchinson et al. [20] also focused on the privacy and security concerns within Android and iOS apps, particularly dating apps. These authors found that one out of five apps investigated leaked all relevant user data, including the app user's email, phone number, GPS location, and sent and received messages. The authors note that the varying levels of data leakage present in these dating apps may allow forensic investigators to prove or disprove the app users' alibis.

Video sharing is a form of social media, and both Android and iOS platforms provide multiple apps to facilitate this capability. The Snapchat app is a prime example of that. This app promises its users that after their pictures, messages, or videos expire, they are deleted. Alyahya et al. [21] aimed to understand what exactly can be recovered from deleted and expired "snaps." The experiment was carried out with a test account in which researchers conducted average Snapchat activities such as sending and receiving snaps, deleting snaps, and posting/deleting from a story. Magnet AXIOM Process was used to create a physical image of the phone's memory, and AXIOM Examine and Autopsy were used to analyze the recovered image. The results showed varying levels of recoverable artifacts depending on the forensic tool used. It is noteworthy to focus on using two popular tools to understand the difference in reported results, and to observe and document that different programs and methods may yield different and sometimes incomplete results.

Literature so far shows some gaps in technology that we overcome in this research. In [11], employing ADB as a primary file system extraction method has limitations. It only extends to logical extraction. Additionally, the study is limited to recoverable evidence on Android. We addressed these caveats via Axiom Examine for Android as well as iOS. During the data population, we deliberately deleted some data to verify the extraction of deleted content. This deleted content was successfully recovered and is discussed in the analysis section. In [12], one of the novelties is the contribution of the TikTok.py python module for Autopsy, which is yet to be peer-reviewed. We surpass this limitation via a well-established and peer-reviewed tool, Magnet Examine. This helped us to dig more than just XML files, as it yielded user login information, profile information, posting activity,

chat information, and app usage information, in addition to databases. Again, this is recovered from both Android and iOS devices. In ref. [15], 20% of the data failure occurs due to the non-recovery of posted videos. One highlight of our research is recovering the posted videos as well as deleted videos.

3. Methodology

The authors in this study conducted the investigation based on National Institute of Standards and Technology (NIST) standards for mobile device investigations and implemented the forensic procedures involved [22]. This study used two smartphone devices: an iPhone 7 (A1660) running iOS 13.3.1 and a Samsung Galaxy S7 (G930U) smartphone running Android 8. The choice of these devices was purely out of convenience, although we ensured that both devices were running the newer operating systems. These devices have also been shown to be jailbroken or rooted without significant challenges. Jailbreaking/rooting mobile devices allows us to access the user data stored on the device. However, this process tends to alter the device's state, potentially losing relevant data. As such, forensic investigators should consider only jailbreaking/rooting devices when necessary.

The forensic software tools used for the acquisition of these devices included Cellebrite UFED 4PC (Version 7.42.0.82) and Magnet AXIOM Process (Version 4.7.0.22371). The examination and analysis of the resultant forensic images were done using Magnet AXIOM Examine (Version 4.9.1.23338-1). Cellebrite and Magnet are both widely used and accepted by digital forensic investigators and courts of law. NIST guidelines, despite being written in 2014, maintain a computer forensics tools catalog that is periodically updated. The Computer Forensics Tool Testing Program (CFTT) at NIST monitors these updates while focusing on advances in forensic investigation software and ensuring their reliability [23]. Having verified Magnet and Cellebrite under the mobile forensics tools catalog, the authors are confident about the integrity of the results produced via these tools. To aid in future research, we have summarized the forensic processes and software involved in this investigation in Figure 1.

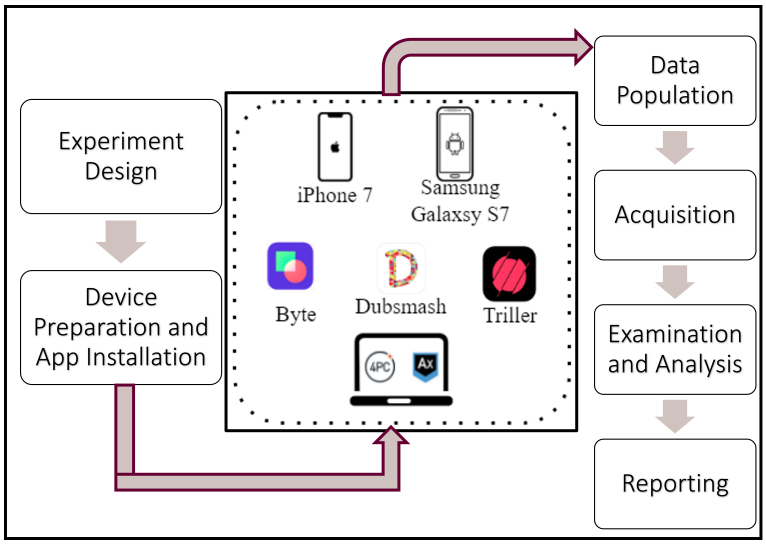

Figure 1. Methodology used in this research.

3.1. Data Population

We created two accounts, one Apple account and one Google account, to prepare the data population. The smartphones were then populated as follows:

1. Factory reset the iPhone 7.
2. Rooted the Samsung Galaxy S7.
3. Signed into the new Apple and Google accounts on the iPhone 7 and Samsung Galaxy S7, respectively.

4. Populated each device with data, as necessary, following the NIST guidelines [22]. Downloaded and installed the three apps (Byte, Dubsmash, and Triller) from the App Store and the Google Play Store on the respective handsets. The app versions are given in Table 1. For each app, we did the following:
   (a) Created two accounts through the app, one used on the iPhone 7 *(focyber21)* and the other on the Samsung Galaxy S7 *(focyber86)*.
   (b) Used each smartphone to interact with the app and each other.
5. Performed a full acquisition of both devices.
6. Examined both forensic images using AXIOM Examine.

Table 1. Version numbers of the applications investigated.

Application	Version (February 2021)	
	Android	iOS
Byte	1.2.43	0.6.0
Dubsmash	5.19.0	5.20.0
Triller	v19.1b8	22

Social Media App Interactions

This section accounts for the population of the app for which the authors interacted with all free features of the app. We used each app (Dubsmash, Triller, and Byte) to do the following activities:

- Create and set up an account on each app.
- Setting up the bio on each app.
- Create a video and post it on the app's wall.
  - Video #1: Create a video, save it in the phone's gallery, and post it on the app's wall.
  - Video #2: Create and post a self-destruction video (with timer) and post it on the app's wall.
  - Video #3: Create and post a video and delete it from the app's wall.
  - Video #4: Create, but never post, a video, and instead save it in the app's draft storage.
- Search for keywords using the application's search feature.
- Chatted through the chat feature of the apps.
  - Saved a draft message while chatting.
  - Exchanged GIFs, emojis, voice notes, photos, and videos.
- Interaction with other accounts on the apps.
  - Follow other people's accounts and unfollow some of them.
  - Follow other's people's accounts and block some of them.
  - Press the like button on some videos of the people you are following.
  - Press the share button on some videos of the people you are following.
  - Like some hashtags trending on the app (not necessarily the trending one though).
  - Download (or export) some of the videos of the people you are following in the phone's local storage.

3.2. Acquisition

The Samsung Galaxy S7 was acquired using Magnet AXIOM Process while the iPhone 7 was acquired using Cellebrite UFED 4PC (Version 7.42.0.82).

3.3. Analysis

For the purpose of examination and analysis, the resultant forensic images of both the devices, that is, the Samsung Galaxy S7 and iPhone 7, were treated with Magnet AXIOM

Examine (Version 4.9.1.23338-1). The resultant Samsung S7 forensic image had a file size of 29.7 GB, and Magnet AXIOM acquired 108, 345 artifacts. For iPhone's forensic size, the file size was 8.68 GB, and Magnet AXIOM acquired 498, 965 artifacts. No SIM cards were added to either phone, and thus all the communication was purely Internet (WiFi) based. Other software tools were also utilized to make the examination and analysis of the images more efficient. These are listed in Table 2.

Table 2. Summary of software tools used during the analysis phase.

Tool	Version (February 2021)	Purpose
Magnet AXIOM Examine	4.9.1.23338-1	Viewing artifacts from forensic image
DB Browser for SQLite	3.12.2	Viewing database files
DCode	5.1	Converting timestamps

4. Findings

In this section, we present relevant findings that can help investigators more efficiently conduct analyses using these and similar apps. We also inform any of these app users about privacy leaks that may affect them. Since we found many accounts in these apps, we take it upon ourselves to protect the privacy of these app users. To that end, we have redacted any Personally Identifiable Information (PII) that is not associated with our test accounts.

In the following subsections, we present the artifacts we recovered, discussing them under the five categories described below:

- **Login Information**: artifacts in this category include the app user's email address, password, and login tokens.
- **Profile Information**: artifacts in this category include the app user's username, user ID, profile picture, bio, and birth date.
- **Posting Activity**: these artifacts include timestamps, posted videos, video likes and comments, deleted videos, account follow, unfollow, blocking activity, and search queries.
- **Chat Information**: artifacts in this category include messages sent/received through the app.
- **App Usage Information**: artifacts in this category include information on how the user used the app, including session information and devices used.

For the Android investigation of three apps (Byte, Triller, and Dubsmash), we looked into Magnet AXIOM's *samsung SM-G930U Full Image-SDA.raw* file, which consisted of 23 partitions. In the initial investigation, we identified the partition of forensic interest, that is, Partition #23 (sized 24.59 GB). Similarly, for the iPhone investigation of the apps, Cellebrite UFED 4PC acquired and created a *FullFileSystem.1.dar*, which was then investigated using Magnet AXIOM. The search on this image was conducted with the help of keywords *focyber21* and *sweetfire21*, resulting in 925 and 143 matches, respectively.

4.1. Byte Forensic Artifacts

This section represents Byte-related findings such as what paths to look for specific artifacts, a directory tree of a few important sub-directories, and division of artifacts into two sub-sections (based on OS): Android and iOS.

4.1.1. Byte Android Artifacts

Upon analyzing the acquired phone to investigate the Byte app, we observed that Android stores all Byte-related files under the base path shown in Table 3. Therefore, all artifacts listed in Table 4, as well as the artifacts discussed in the following paragraphs, start at these base locations.

Table 3. Byte app package paths.

OS	Package Path
Android	`samsungSM-G930UFullImage-SDA.raw-Partition23(EXT-family,24.59GB)\data\co.byte`
iOS	`FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\23B078E4-6E9E-44E0-AEA9-E3B859503D4F`

Login and Profile Information

Starting from setting up the account on the device, the first artifact in the line of evidence is the registration date. The authors used email *focyber86@gmail.com* for registration. As verified during the investigation, the account was created on 2 April 2021. Byte app stores all the timestamps in an Epoch time format; therefore, the Epoch converter website [24] was used to view these Epoch times in a more human-readable format.

When an account is registered on Byte, the app stores all account-related information such as username, a URL link to the profile picture, registration date, profile bio, and the user's date of birth (DOB) in the *Account* table within the *\databases\byte.db* database (see Figure 2). We could not recover another user's DOB besides its column being present in the table (empty for others). However, the *focyber86* DOB was still recovered in plain text. This table also contains every user's account metrics such as the following count, block count, unread conversations count, and isEmployee (whether the user is Byte's employee or not), among others. Based on our interactions with the app, the values present in this table for our account are found to be consistent with our actions on the app.

Communication between Byte Accounts

Two Byte test accounts were created for this research, *focyber86* to be used on the Android device and *focyber21* to be used on the iOS device. After setting up the account on the app, *focyber86* interacted with the app and other users, including our iOS account user. Conversations or chats between the Android user and the iOS user were recovered in plain text from the *DbChatMessage* table within the *byte.db* database (see Figure 3). For example, the message *"Audio video photo messages working sweetfire21"* was exchanged between the two users and recovered. Each user is assigned a unique userID (or authorID) and conversationID to keep track of different interactions, both shared in Table 4. The chat feature allows users to share hashtags, URLs, and also tag other people. However, Byte chat does not support sending images, videos, or voice messages in the chat.

id	username	displayName	avatarURL	bio	birthday	registrationDate	badges	conversationID	isRegistered
AZDEW665SJF7ZHTVJW7TXRXGWE	byte	byte	https://e6k9t9a9.stackpathcdn.com/...	say something nice		1568319559000	["beta"]	NULL	1
6VNXFEKD3NCKNFXCIJBNFXNFA4			https://e6k9t9a9.stackpathcdn.com/...			1596886419000	[]	NULL	1
HDP7EZMRNVBFLM4RTW2WYG5KOI	focyber86	focyber86	https://e6k9t9a9.stackpathcdn.com/...	hi, I'm sweetfire21 byte account bio	1994/01/01	1605725268000	[]	NULL	1

followerCount	followingCount	loopCount	loopsConsumedCount	isFollowed	isFollowing	isSubscribed	isBlocked	isDeleted	foregroundColor	backgroundColor	isFollowingFeedPreferred
0	0	0	0	0	1	0	0	0	-3352855	-16777216	0
0	0	0	0	0	0	0	0	0	-3408904	-12039068	0
2	4	92	132	0	0	0	0	0	-1906177	-16733454	0

shouldShowCommunityPicker	unreadConversationsCount	unreadConversationsTimestamp	preferences	publicLikesFeed	isEmployee
1	0	0	NULL	0	0
1	0	0	NULL	0	0
0	1	1612539723122927	{"publicLikesFeed":false}	0	0

Figure 2. The contents of byte.db\Account from Byte Android investigation demonstrating personal user data (id, birthday, registration date, count of followers, message ID) details as shown in DB Browser for SQLite.

	conversationId	id	body	created	authorID
	Filter	Filter	Filter	Filter	Filter
1	YOV2WIXTYJETBDEHP6A2GEGYUE	8	{"type":"text","text":"https://byte.co/b/48UMhnHz2XG","mentions":[],"hashtags":[],"urls":[]}	1612476937000	D3323I2GZFBR5NEQHSKLAWGVAY
2	YOV2WIXTYJETBDEHP6A2GEGYUE	7	{"type":"text","text":"https://byte.co/b/JSBkLSkw3eX","mentions":[],"hashtags":[],"urls":[]}	1612476274000	HDP7EZMRNVBFLM4RTW2WYG5KOI
3	YOV2WIXTYJETBDEHP6A2GEGYUE	6	{"type":"text","text":"it's simple messaging, we don't have the capability to send images, videos or audio messages. ","mentions":...	1612469500000	HDP7EZMRNVBFLM4RTW2WYG5KOI
4	YOV2WIXTYJETBDEHP6A2GEGYUE	5	{"type":"text","text":"🤣🤣🤣","mentions":[],"hashtags":[],"urls":[]}	1612469329000	HDP7EZMRNVBFLM4RTW2WYG5KOI
5	YOV2WIXTYJETBDEHP6A2GEGYUE	4	{"type":"text","text":"👀☺️🙇","mentions":[],"hashtags":[],"urls":[]}	1612469295000	D3323I2GZFBR5NEQHSKLAWGVAY
6	YOV2WIXTYJETBDEHP6A2GEGYUE	3	{"type":"text","text":"hey it's me android sweetfire21","mentions":[],"hashtags":[],"urls":[]}	1612469273000	HDP7EZMRNVBFLM4RTW2WYG5KOI
7	YOV2WIXTYJETBDEHP6A2GEGYUE	2	{"type":"text","text":"Hey sweetfire21","mentions":[],"hashtags":[],"urls":[]}	1612469250000	D3323I2GZFBR5NEQHSKLAWGVAY
8	YOV2WIXTYJETBDEHP6A2GEGYUE	1	{"type":"text","text":"This is your first time messaging. Be nice, and follow the Terms of Service.","mentions":[],"hashtags":[],"urls":[]}	1612469234000	AZDEW665SJF7ZHTVJW7TXRXGWE

Figure 3. Conversation between Byte Android accounts from byte.db\DbChatMessage as shown in DB Browser for SQLite.

Posting Activity

An important part of the population was testing how the Byte app handles the storage and recovery of videos. Multiple videos were created and modified for public viewing and later modified to private. Certain videos were created, but were set on auto-destruction mode after 15 min. Therefore, we tested four different video features as defined in Section 3.1: Data Population. Interestingly, any public video can be recovered from at least four different places associated with that video. For example, we created a video and published it on the Byte wall. During recovery, we found the storage path and three unique cloud-based URLs that could be pasted on any web browser to access it anytime as long as the cloud still holds it. These locations and URLs are (1) the phone's gallery storage path, (2) a cloud-based URL link, (3) a different cloud URL link for additional watermarking on the video (watermarked during population), and (4) a Byte shareable link that can be used on any web browser to open the video from the URL. The table *Post* from the database file *byte.db* extensively stores multiple attributes for each video posted on the app's wall. These attributes include a unique ID for the post, authorID or userID, caption used on the video, date it was created on, like count, liked by the user, rebyte (equivalent to reshare feature) by user, shareURL (Byte based URL of the video), thumbnail source URL, video source URL, animated thumbnail, watermarked video URL, comment count, comments made on the video, and hashtags (see Figure 4). We recovered all the other three videos created during the population with all sets of unique URLs as described above. The location and file name of these videos are listed in Table 4. Byte stores these videos in *.0* file format and hence after exporting these artifacts, we converted from *.0* to *.mp4* format using the command-prompt command, *ren *.0 *.mp4*, where ren is rename.

id	authorID	author	caption	date	likeCount	likedByMe	bytedByI	shareURL	thumbSrc	videoSrc
TSJMG...	64A776A...	{"id":"64A...	megan. ily but.... what is th...	1612483221000	27	0	0	https://byte.c...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...
CPIBPT...	FRSGM4E...	{"id":"FRS...	The directors cut is savage	1609943045000	4078	0	0	https://byte.c...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...
UGWN...	HDP7EZ...	{"id":"HDP...	#sweetfire21 #androids7	1612470355000	2	0	0	https://byte.c...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...

trackInfo	media	animatedThumbnail	watermarkedVideo	loopCount	commentCount	comments
{"videoTL":15.066667,"audioTL...	{"source":{"id":"source","url":"https:...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...	131	1	[{"id":"TSJMGDOYQFAJDF3ILXEGW2...
{"videoTL":9.7,"audioTL":...	{"source":{"id":"source","url":"https:...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...	35634	155	[{"id":"CPIBPTSYS5FW7K4AYHMM7A...
{"videoTL":15.531267,"audioTL...	{"source":{"id":"source","url":"https:...	https://e6k9t9a9.stackpathcdn.co...	https://e6k9t9a9.stackpathcdn.com/...	41	2	[{"id":"UGWNBXAHB5F4PH5LCRUHI...

commentCursor	permissions	selfDestructDate	allowRemix	soundParentID	soundTitle	soundArtworkSrc	soundCanUpdate	gridLabel	community	communityId
NULL	[]	NULL	1	NULL	body club remix	https://e6k9t9a9.stackpathcdn.co...	0	NULL	{"id":"trash","slug":"trash","descripti...	trash
CZPI76JPB2FFQAI	[]	NULL	1	NULL	original sound	https://e6k9t9a9.stackpathcdn.co...	0	NULL	{"id":"happyholidays","slug":"happyh...	happyholidays
NULL	[]	NULL	1	NULL	original sound	https://e6k9t9a9.stackpathcdn.co...	0	NULL	NULL	NULL

mentions	hashtags	urls	stubId
[]	[]	[]	NULL
[]	[]	[]	NULL
[]	[{"text":"#sweetfire21","tag":"sweet...	[]	NULL

Figure 4. Byte's Android video-related attributes stored in byte.db\Post recovered from as shown in DB Browser SQLite.

App Usage Activity

The authors followed the population protocol described in Section 3.1 for the app usage. Thus, we were interested in recovering the likes, comments, followed accounts, unfollowed accounts, blocked accounts, followed hashtags, etc. These activities were easily recoverable from byte.db\ActivityEntry as shown in Figure 5. In general, the *byte.db* database contained 27 tables to store the data collected as part of the application's functionality. For example, there were tables named Account, AccountListMembership, ActivityEntry, CommentListEntry, DbChatMessage, DbChatTyper, DbConversation, DbConversationMember, DbCounter, Post, PostFeedMembership, android_metadata. For future investigations, the database file *byte.db* could be tracked first, as it contains the most relevant forensic artifacts from the Byte app on Android. Figure 6 can be referred to get a general overview of the file structure consisting of forensically important directories.

id	date	type	subType	body	authorID	author ▾1
UGWNBXAHB5F4PH5LCRUHIHLOLU-...	1612470800000	"comment"	NULL	{"id":"UGWNBXAHB5F4PH5LCRUHIH...	D3323I2GZFBR5NEQHSKLAWGVAY	{"id":"D3323I2GZFBR5NEQHSKLAW...
UGWNBXAHB5F4PH5LCRUHIHLOLU-...	1612470771000	"like"	NULL	{"authorID":"D3323I2GZFBR5NEQHS...	D3323I2GZFBR5NEQHSKLAWGVAY	{"id":"D3323I2GZFBR5NEQHSKLAW...
D3323I2GZFBR5NEQHSKLAWGVAY-...	1612469218000	"follow"	NULL	{"followerAccountID":"D3323I2GZFB...	D3323I2GZFBR5NEQHSKLAWGVAY	{"id":"D3323I2GZFBR5NEQHSKLAW...
ITW4SZPWCFGCDGRRMEUIHUTGLA-...	1605725756000	"follow"	NULL	{"followerAccountID":"ITW4SZPWCF...	ITW4SZPWCFGCDGRRMEUIHUTGLA	{"id":"ITW4SZPWCFGCDGRRMEUIHU...
UGWNBXAHB5F4PH5LCRUHIHLOLU-...	1612473117000	"like"	NULL	{"authorID":"MIXC57BGL5GOLA6FVQ...	MIXC57BGL5GOLA6FVQIVAKUWWE	{"id":"MIXC57BGL5GOLA6FVQIVAKU...

Figure 5. The contents of byte.db\ActivityEntry from Byte Android investigation demonstrating as shown in DB Browser for SQLite.

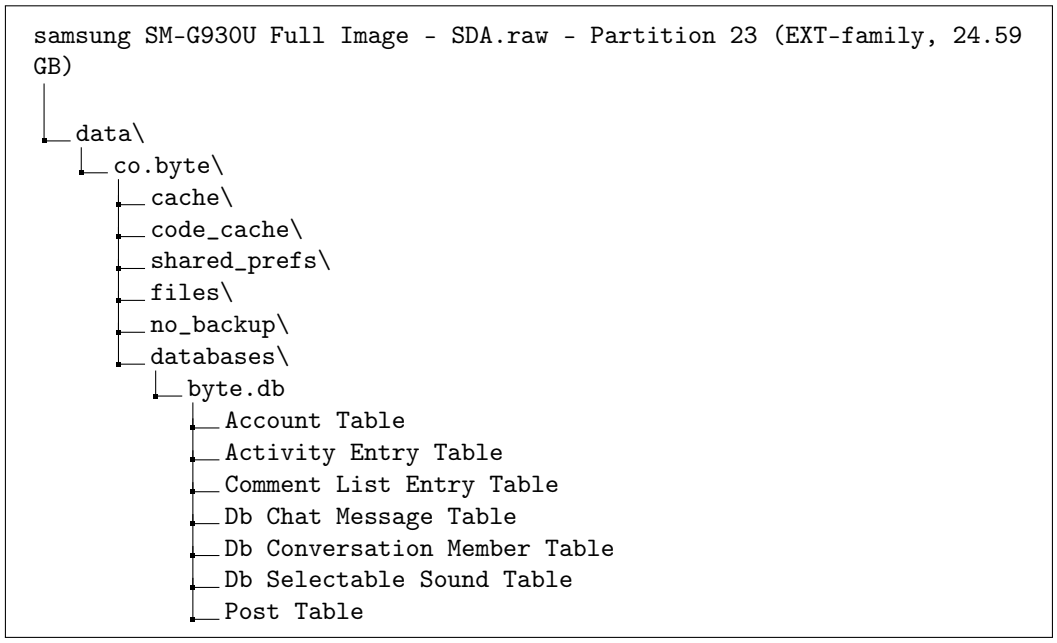

Figure 6. Byte Android file structure as captured in Magnet AXIOM Examine.

4.1.2. Byte iOS Artifacts

Most of the videos were stored in the *mp4* format in the videos folder. The video folder could be found in the caches folder. As mentioned before, the path for the Byte iPhone-related files is listed in Table 3. *FullFileSystem.dar* contained 11 folders and 2 files. Most of the Byte-related artifacts were stored in a folder called 23B078E4-6E9E-44E0-AEA9-E3B859503D4F. In 23B078E4-6E9E-44E0-AEA9-E3B859503D4F, there was another folder called co.byte.video. When selected, the first time a Byte interaction took place was verified.

Viewing certain iOS artifacts in Magnet AXIOM is difficult, so we used a DB SQLite viewer application. To do this, we recovered a database called *cache.db* (from within the folder *co.byte.video*) as an artifact and opened it in the application. The birthdate was registered as 14/02/1995 (see Figure 7). The cloud URL for the profile picture is stored as https://e6k9t9a9.stackpathcdn.com/avatars/ZVGDQH5CD5GRNKXBXZUIRRFFNM.jpg. The Android byte account *focyber86* and the iOS account *focyber21* communicated during the research period, and these conversations could easily be traced back. Their conversation was also examined using DB Browser for SQLite. Each message was given an id (for example, 1, 2, 3, etc.) as shown in Figure 3. The highest number is the most recent text, and the smallest is the oldest text. The two accounts exchanged texts, emojis, and mentions with each other. As for videos, they were located in the ...*Library**Caches**videos* under the home directory from Table 3.

Table 4. Byte Android and iPhone artifacts.

S.No.	Artifact Type	Android Location	Artifact Data Value/File Name
1	Android Account username	*\databases\byte.db\Account	focyber86
2	iPhone's Account username	*\databases\byte.db\Account	focyber21
3	Account avatarURL	*\databases\byte.db\Account	https://e6k9t9a9.stackpathcdn.com/avatars/P2GJL2KURBFGPMAI74VRWXLTYU.jpg
4	Account Registration Date and Time	*\databases\byte.db\Account	1605725268000 (Epoch Time)
5	Account Bio (Profile)	*\databases\byte.db\Account	*hi, i'm sweetfire21 byte account info*
6	Account Other Info	*\databases\byte.db\Account	date of birth, conversation ID, follower count, following count, Block Count, unread Conversation Count, isEmployee
7	UserID (or authorID) of focyber86	*\databases\byte.db\Account	
8	Conversation ID with focyber21	*\databases\byte.db\Account	YOV2WIXTYJETBDEHP6A2GEGYUE
9	Video #1 (recovered from Phone's gallery)	*\cache\v	sweetfire21 android first video 02/04 hashtag: #sweetfire21 #androids7 Filename:UGWNBXAHB5F4PH5LCRUHI -HLOLU.0
10	Video #1 (recovered from cloud I)	*\byte.db\Post\videoSrc	https://e6k9t9a9.stackpathcdn.com/videos/LCH4M4F7DFGM3CG7HLM4CBFVBQ-h264.mp4
11	Video #1 (recovered from cloud II)	*\byte.db\Post\cloudwatermarkedVideo	https://e6k9t9a9.stackpathcdn.com/videos/LCH4M4F7DFGM3CG7HLM4CBFVBQ-watermarked.mp4
12	Video #1 (Byte shareable URL)	*byte.db\Post\shareURL	https://byte.co/b/JSBkLSkw3eX
13	Video #2 (Stored Locally)	*\cache\v	6FGXFSTCABCX5NMAGBS4VVH42Y.0
14	Video #3 (Deleted Video)	*\cache\v	ITVIXVIX5BH4LBFFBABNHC53M4.0
15	Video #4 (Self-destruction Video)	*\cache\v	JJKHFJE42RF7FJC3YBEMAC2XDI.0
16	Exchanged messages in chat	*\databases\byte.db\DbChatmessage	hey it's me android sweetfire21
17	Posting Activity	*\databases\byte.db\ActivityEntry *Partition23\data\co.byte\...	#likes, #follow, #comments
S.No.	**Artifact Type**	**iOS Location**	**Artifact Data Value**
1	iPhone's Account username	*\Cache.db\cfurl\cache\receiver\data	focyber21
2	Account avatarURL	*\Cache.db\cfurl\cache\receiver\data	https://e6k9t9a9.stackpathcdn.com/avatars/ZVGDQH5CD5GRNKXBXZUIRRFFNM.jpg
3	Account Registration Date and Time	*\Cache.db\cfurl\cache\receiver\data	1612468233 (Epoch Time)
4	Video #1	*\Library\Caches\videos	CNQSSNIEJBABLJZLXXY4-6LJROM-progressive.mp4
5	Video #2	*\Library\Caches\videos	Y324SVBX3ZFYBAWZUTKQM7I6XU-progressive.mp4
6	Saved Video (made by someone else)	*\Documents\videos\export\bytebyKiera.Edwards.mp4	CNQSSNIEJBABLJZLXXY46LJROM -progressive.mp4
7	Deleted Video	*\Library\Caches\co.byte.video\Cache.db	captioned: "Gonna delete this one sweetfire21
8	Blocked Account	*\Library\Caches\co.byte.video\Cache.db	Blocked account display name: kaylanatsumi
9	Exchanged Messages In Chat	*\Library\Caches\co.byte.video\Cache.db *\...23B078E4-6E9E-44E0-AEA9-E3B859503D4F\...	Conversation ID: YOV2WIXTYJ ETBDEHP6A2GEGYUE


```
{"data":{"id":"D3323I2GZFBR5NEQHSKLAWGVAY","isRegistered":true,"isChannel":false,
"isSuspended":false,"isDeactivated":false,"isDeleted":false,"registrationDate":
1612468233,"username":"focyber21","avatarURL":"
https://e6k9t9a9.stackpathcdn.com/avatars/ZVGDQH5CD5GRNKXBXZUIRRFFNM.jpg",
"backgroundColor":"#000000","foregroundColor":"#CCD6E9","followerCount":1,
"followingCount":5,"loopCount":59,"loopsConsumedCount":303,"preferences":{"showFollows":
false,"hasSeenCommunityPicker":true,"publicLikesFeed":false},"isFollowingFeedPreferred":
false,"shouldShowCommunityPicker":false,"publicLikesFeed":false,"tests":{"newPostObject"
:true},"birthday":"1995/02/14","unreadConversationsCount":1,
"unreadConversationsTimestamp":1612476867856149},"success":1}
```


Figure 7. focyber21 information recovered from Byte app as shown in DB Browser for SQLite.

4.2. Dubsmash Forensic Artifacts

This section represents Dubsmash-related findings such as what paths to look for specific artifacts, a directory tree of a few important sub-directories, and division of artifacts into two sub-sections (based on OS): Android and iOS.

4.2.1. Dubsmash Android Artifacts

Upon analyzing the acquired Android phone for investigating the Dubsmash app, we observed that Android stores all Dubsmash-related files under the path shown in Table 5. Therefore, all the recovered artifacts listed in Table 6 start with this base location.

Regarding the recovery of any login information, we were able to recover the app user's username (*focyber86*), an authentication token with corresponding expiration timestamp (1612554866), and a refresh token from the `\shared\prefs\com.mobilemotion.dubsmash.a.xml` file. The authentication token may be useful for forensic investigators should they need a method of signing into the Dubsmash account. However, going off the token expiration timestamp, the token was set to expire after 23 h of signing into the app.

Table 5. Dubsmash app package paths.

OS	Package Path
Android	`samsungSM-G930UFullImage-SDA.raw-Partition23\\(EXT-family,24.59GB)\data\com.mobilemotion.dubsmash`
iOS	`FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\4C82EECD-902E-4393-A7F2-2E32BF63E0A7`

Table 6. Dubsmash Android and iPhone artifacts.

S.No.	Artifact Type	Android Location	Artifact Data Value
1	Account username	`databases\dubsmash_database.db`	focyber86
2	Account email address	`databases\dubsmash_database.db`	forcyber86@gmail.com
3	User's uuid	`databases\dubsmash_database.db`	8dba1d56d73f4914958448dd207694c8
4	Account Profile Picture URL	`databases\dubsmash_database.db`	https://d2nr8mwohhwyyc.cloudfront.net/profile_pictures/b83d41d0-8147-4914-851d-6ded3e6ce6ba.jpeg
5	Account Registration Date	`databases\dubsmash_database.db`	04-Feb-21 2:54:24 PM
6	Account Bio (Profile)	`databases\dubsmash_database.db`	hi, I'm sweetfire21 bio for dubsmash
7	Account Activity	`databases\dubsmash_database.db`	number of posts, number of private posts, number of videos, first and last name (if entered)
8	iPhone's Account username	`cache\http_cache\887dcbd43636363ab5103158710b63d7.1`	focyber21
9	iPhone Account uuid	`cache\http_cache\887dcbd43636363ab5103158710b63d7.1`	2df00e8b2b5344d59470078b662509ac
10	iPhone Account Registration Date	`cache\http_cache\887dcbd43636363ab5103158710b63d7.1`	04-Feb-21 4:28:51 PM
11	Conversation uuid with focyber21	`cache\http_cache\e49c8c935df4d90bf243ed9f3f8601b6.1`	1a05ed42368d4194a0c16a255bdee8f3
12	Exchanged Messages in chat	`cache\http_cache\e49c8c935df4d90bf243ed9f3f8601b6.1`	see Figure 12
13	Search Queries	`cache\http_cache\5345cde3b206ce1ad55bd06b773e1236.1,`	
14	Notifications	`cache\http_cache\f5420f989fe442f13b6fcf3591924432.1`	see Figure 9
15	Video #1 (app's storage)	`files\recording_cache\video_upload\408ca681-6e36-422b-bbf8-46872e3a49ab_overlayed.mp4`	
16	Video #1 (Byte shareable URL)	`cache\http_cache\758d2aa5e2169d4ba9ef136ed5b664e7.1`	
17	iPhone Account Posted Videos (URL)	`cache\http_cache\7472eeb55afa96525691d944b77f32fd.1`	https://d28cdhge7i53l0.cloudfront.net/dubs/459e71c2-c908-4896-ae25-c3aa08be192b.mp4 (accessed on 6 July 2021)
18	iPhone Account Posted Videos (app's storage)	`files\download_dir\6\25.0.1612475155868.v3.exo`	
S.No.	**Artifact Type**	**iOS Location**	**Artifact Data Value**
1	Account username	`Library\Preferences\com.mobilemotion.dubsmash.plist`	focyber21
2	Account email address	`Library\Preferences\com.mobilemotion.dubsmash.plist`	focyber21@gmail.com
3	User's birthday	`Library\Preferences\com.mobilemotion.dubsmash.plist`	1995-02-14
4	User's uuid	`Library\Preferences\com.mobilemotion.dubsmash.plistt`	2df00e8b2b5344d59470078b662509ac
5	Account Profile Picture URL	`Library\Preferences\com.mobilemotion.dubsmash.plist`	
6	Account Registration Date	`Library\Preferences\com.mobilemotion.dubsmash.plist`	2021-02-04T21:28:51.247255+00:00
7	Account Bio (Profile)	`Library\Preferences\com.mobilemotion.dubsmash.plist`	*This serves as my bio sweetfire21*
8	Account Activity	`Library\ApplicationSupport\Google\Measurement\google-app-measurement.sql`	

The `\databases\dubsmash_database.db` file, *users* table holds a wealth of information relating to the app user's profile, including their Dubsmash uuid, username, email, display name, first and last name (if entered), a URL link to their profile picture, the date joined, number of posts and followers, and bio (see Figure 8 for a reduced listing of these artifacts).

uuid	username	email	displayName	avatar	dateJoined	numPosts	numFollows	numVideos
8dba1d56d...	focyber86	focyber86@gmail.com	focyber86	https://...	2021-02-04T19:54:...	2	1	2

Figure 8. Reduced profile information for the Dubsmash Android account.

Strikingly, we recovered a plethora of artifacts related to the app user's posting activity. Most of these artifacts are recovered from the *\cache* folder and include *.m4a* files containing the audio the app user recorded/created for their video post, URL links to the *.mp4* videos the user and other users posted, thumbnails of videos including thumbnails from the videos the app user posted as well as from the iOS user, listings of the video recommendations made to the app user, the app user's profile information similar to that found in the *users* table above, and a listing of results for searches the user made.

Specifically, the `cache\http_cache\f5420f989fe442f13b6fcf\3591924432.1` file holds all the notifications that the app showed the user. Most notably, this data includes the *notification_type* (your_video_is_popular, new_dub_mention, you_were_in_a_duet, new_video_comment, and video_liked) and the *payload*, which holds the content of the notification including the uuid and username of the person who generated the notification (*focyber21*), the uuid of the post, and the uuid of the post creator (*focyber86*). Figure 9 shows the *payload* section of a notification that was generated when the iOS account user commented on the Android app user's posted video: (1) timestamp of when the comment was made, (2) notification type: *new_video_comment*, (3) uuid of the iOS user, (4) uuid, thumbnail URL, and uuid of the video the iOS user commented on, (5) plain text comment the iOS user left and the uuid of that comment, and (6) the username of the account that commented.

The `\cache\http_cache\c0fedc9a276e5eb8266af2966af6ee16.1` JSON file contained all the comments the iOS user left and the comments the Android app user left on one of the app user's posts. Figure 10 shows the comment of the iOS user and the subsequent reply of the Android user. Similarly, the app user commented on the iOS user's video post, and this artifact was recovered from the JSON file located at `\cache\http_cache\58055e6c452b9ba2adddac83f8fc6a61.1`.


```
"created_at":"2021-02-04T21:41:12.000000+00:00",  1
"group_count":null,
"notification_type":"new_video_comment",  2
"payload":
"{\"user_uuid\": \"2df00e8b2b5344d59470078b662509ac\",  3
\"source_object_type\": \"video\",
\"source_object_uuid\": \"2613e8a015af4d2db6678a3ebd3f2117\",
\"original_object\": {\"thumbnail_url\":                                  4
\"https://d28cdhge7i5310.cloudfront.net/dubs/thumbnail/a754f76d-
bed1-4d4b-ac2f-79da2b75b6ea.jpeg\",
\"uuid\": \"2613e8a015af4d2db6678a3ebd3f2117\"},
\"text\": \"Good colours sweetfire21\",
\"uuid\": \"fe4d37356b8748f486e83517a8245b01\",  5
\"username\": \"focyber21\"}",  6
```


Figure 9. Dubsmash notification generated and shown to the Android user when the iOS user left a comment.

The *\files\download_dir* folder holds multiple folders containing *.exo* video files the app user may have come across while using the app and the videos the Android user and the iOS user posted. A *.mp4* version of the video the Android user posted was also found within the *\files\recording_cache* folder.

The *\cache\http_cache* folder holds numerous JSON files containing information about videos posted by different users. We were able to recover a URL link to the posted video, a URL link to the audio from the video, video title, video creation date, the number of likes and views the video has, and whether the app user liked the video (see Figure 11). The URL links were still live some four months later. Each video and the original sound were accompanied by the details of the content creator, which includes the creator's date joined, uuid, username, display name, and a URL link to the creator's profile picture.

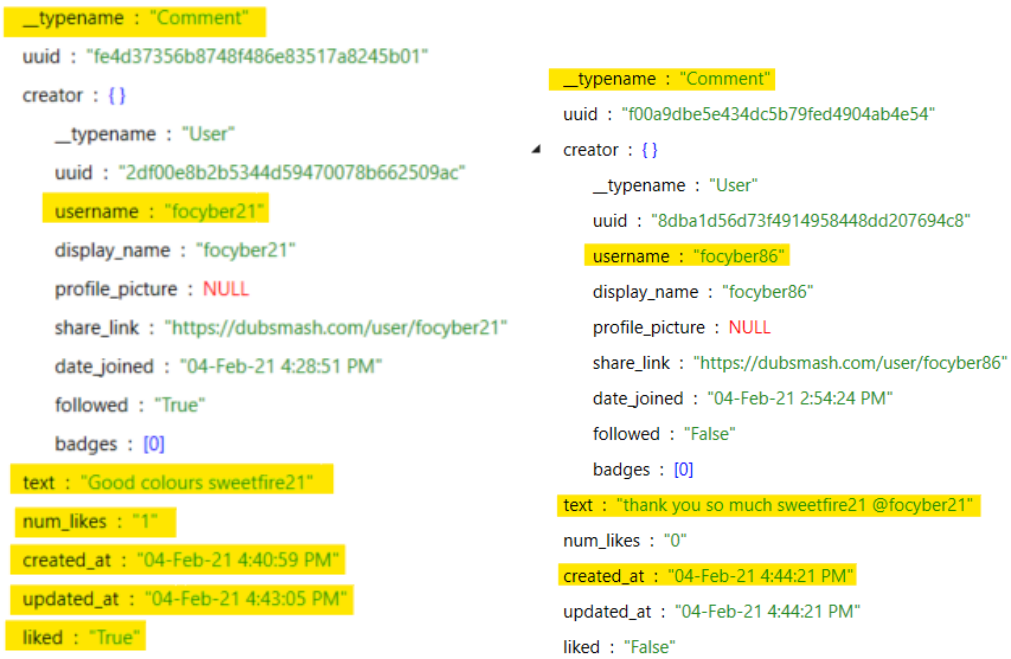

Figure 10. Comment left by iOS user (**left**) and Android user's reply (**right**), as recovered from the Dubsmash app on Android.

Figure 11. Details of the Dubsmash video the iOS user posted as recovered from the *7472eeb55afa96525691d944b77f32fd.1* file on Android.

Regarding the chat messaging activity, we were able to get a full record of the sent and received messages the Android user had with the iOS account from the `\cache\http_cache\e49c8c935df4d90bf243ed9f3f8601b6.1` JSON file. A complete reconstruction of the chat activity between our two test accounts is given in Figure 12. The timestamps are in GMT format, local to the device. To get the corresponding time when we performed the activity, simply subtract five hours from the given timestamp shown in Figure 12. Each record includes the *created_at* timestamp of when the message was sent, the *uuid*, *username*, *display_name*, *profile_picture* URL, and *date_join* of the message author, and the *body* or content of the message being sent/received. The progression of the conversation between our two test accounts is as follows:

1. The iOS account user (*focyber21*) sent a message to the Android user (*focyber86*) saying *"Hey this is a chat sweetfire21"*
2. The Android user then replied *"hi sweetfire21 dubsmash chat reply"*.
3. The iOS user replied saying *"Can't send anything other than text sweetfire21"*.
4. The iOS user sent an emoji that was not rendered by Magnet AXIOM.
5. The Android user replied with an emoji.

Within this same *http_cache* folder, we were also able to recover the first message the app user received from the iOS account user (see Figure 13) within the `b9d37e98893f0ae03c074a4c4d31f8ff.1` JSON file. However, this artifact is labeled *most_recent_message* and corresponds to the message received at #1 in Figure 12. It should be noted that the artifact shown in Figure 13 has its *is_read* value set to *False* (in red), while the artifact shown in #1 of Figure 12, has its value *is_read* set to *True*.

Figure 12. Full chat history between the Android and iOS accounts, as recovered from the Dubsmash app on Android.

Figure 13. First chat the Android account received from the iOS account, as recovered from the Dubsmash app on Android.

Details of how the user used the app were limited. We were only able to recover the number of sessions the user had from the `user_sessions` table within the `databases\dubsmash_database` database file. The model of the phone being used (*samsung_SM-G930U*) was recovered from the `databases\google_app_measurement_local.db-journal` file. Figure 14 is a general overview of file structure consisting of forensically important directories.

```
samsung SM-G930U Full Image - SDA.raw - Partition 23 (EXT-family, 24.59 GB)
└── data\
    └── com.mobilemotion.dubsmash\
        ├── cache\
        │   ├── .m4a files
        │   ├── .mp4 files
        │   └── http_cache\
        ├── code_cache\
        ├── shared_prefs\
        │   └── com.mobilemotion.dubsmash.a.xml
        ├── files\
        │   ├── download_dir\
        │   └── com.mobilemotion.dubsmash_preferences.xml
        ├── no_backup\
        └── databases\
            ├── google_app_measurement_local.db-journal
            └── dubsmash_database.db
                └── users table
```

Figure 14. Dubsmash Android file structure as captured in Magnet AXIOM Examine.

4.2.2. Dubsmash iOS Artifacts

Upon analyzing the acquired Apple phone for investigating the Dubsmash app, we observed that Apple iOS stores all forensically relevant Dubsmash artifacts under the path shown in Table 5. Therefore, all recovered iOS artifacts listed in Table 6 and below start with this base location. The package name for the Dubsmash app on iOS is the same as on Android (*com.mobilemotion.dubsmash*).

Regarding the recovery of any login information, we were able to recover the app user's uuid (*2df00e8b2b5344d59470078b662509ac*), an authentication token with corresponding expiration timestamp (*634253331*), and a refresh token from the `Library\Preferences\com.mobilemotion.dubsmash.plist` file. The authentication token may be useful for forensic investigators should they need a method of signing into the Dubsmash account. However, going off the token expiration timestamp, the token was set to expire after 24 h of signing into the app.

The profile information of the user of the app was also recovered from the `com.mobilemotion.dubsmash.plist` file. This information includes the app user's uuid, username, email address, and date of birth, a URL link to the user's profile picture, the bio the user set on their profile, and a timestamp of when the app user joined Dubsmash (see Figure 15).

{"username":"focyber21","emails":["focyber21@gmail.com"],
"birthday":"1995-02-14","language":"en",
"profilePicture":"https:\/\/d2nr8mwohhwyyc.cloudfront.net\/profile_pictures\/6db4c858-1bb8-4699-
"country":"","displayName":"focyber21","badges":[],"uuid":"2df00e8b2b5344d59470078b662509ac","
followsCount":1,"allowVideoDownload":true,"phones":[],"numInvites":0,"numPublicPostPlays":94,
"bio":"This serves as my bio sweetfire21","followingsCount":4,
"culturalSelections":[{"countryName":"United States","languageName":"English","code":"en_US"}],
"avatarURL":"https:\/\/d2nr8mwohhwyyc.cloudfront.net\/profile_pictures\/6db4c858-1bb8-4699-8ce
"numPosts":2,"numPrivatePosts":0,"shareLink":"https:\/\/dubsmash.com\/user\/focyber21",
"hasInviteBadge":false,"dateJoined":"2021-02-04T21:28:51.247255+00:00"}

Figure 15. Profile details for Dubsmash user recovered from iOS.

Regarding the posting activity, we were able to recover *.mp4* video files of videos the app user created and came across while using the app, including the videos the Android user posted. These videos were recovered from the `Library\Caches` folder. We also recovered *.jpg* files of frames from those *.mp4* videos, within the `Library\Caches\com.onevcat.Kingfisher.ImageCache.default` folder. However, although the actual video files were recovered, the files are not associated with forensically relevant metadata, such as creator details or created timestamps, as in the Android image.

Our analysis of the Dubsmash app on iOS indicates that the app records pertinent logs of how the user used the app as the *events* table within the `Library\ApplicationSupport\Google\Measurement\google-app-measurement.sql` database holds logs of various activities the app user performed. This table holds the activity name, an epoch timestamp of the last time the activity occurred, and the lifetime count of the number of times the activity has been performed. Figure 16 shows a subset of this data, including when the user created their account (*1612474131.96488*). Not pictured, we also obtained the number of accounts the user followed and unfollowed (but not blocked), the number of posts the user created, the number of posts deleted, and the number of comments the user made, along with many other activities. Similar information was also recovered from the `Library\Preferences\com.google.gmp.measurement.plist` file (see Figure 17).

Other user attributes were also recovered from the *user_attributes* table within the same *google-app-measurement.sql* database. Notably, we can obtain the type of internet connection being used, the app version, and the age of the Dubsmash account (see Figure 18). The first time the app user opened the app was also recovered from the `com.google.gmp.measurement.plist` file.

	name	lifetime_count	urrent_bundle_coun	last_fire_timestamp
	Filter	Filter	Filter	Filter
1	_f	1	1	1612473915.78995
2	user_registration	1	1	1612474131.96488
3	hashtag_subscribe	3	3	1612474267.87552
4	contacts_accept	1	1	1612474286.05175
5	push_accept	1	1	1612474364.26407
6	follow	5	5	1612474421.75618
7	unfollow	1	1	1612474461.3568
8	sound_add	1	1	1612475177.85515
9	notification_interaction	2	2	1612475230.09096
10	sound_tap_on_post	1	1	1612475238.79209
11	explore_content_tap	1	1	1612475393.16228
12	profile_detail_tap	4	4	1612475456.35851
13	sound_upload	1	1	1612475546.4452
14	list_item_tap	6	6	1612475594.17718
15	profile_pic_selected	1	1	1612475636.8267

Figure 16. Events corresponding to user actions recovered from the Dubsmash iOS app.

[0] **preferredMuteState** = False
[1] **perSessionScrollHintDisplayed** = False
[2] **timestampLastLike** = 1612476223
[3] **shouldAutoplay** = True
[4] **currentSessionID** = 3803E76B-9B67-4EA4-8DCE-2380D0BCDA13
[5] **perUserSessionNumbers** Save bytes
[6] **urlCachedTime** Save bytes
[7] **timestampLastPost** = 1612475888
[8] **IBGSignupDate** = 04-Feb-21 9:25:16 PM
[9] **pushAuthorizationGranted** = True
[10] **userUuid** = 2df00e8b2b5344d59470078b662509ac
[11] **perUserPreviousSessionCloseDate** Save bytes
[12] **currentJWT** Save bytes
[13] **videosSeen** Save bytes
Hex preview : 0x5B 0x22 0x32 0x64 0x66 0x30 0x30 0x65 0x38 0x62 0x32
ASCII preview : ["2df00e8b2b5344d59470078b662509ac#video#309243
[14] **currentActiveSessionSegmentStartDate** = 04-Feb-21 11:05:47 PM
[15] **pushToken** = 2952DAD87FB45D54D2028BD2EC1D0DFD51BECED3F247
[16] **lastServerTimeDifference** = -1
[17] **user** Save bytes
Hex preview : 0x7B 0x22 0x75 0x73 0x65 0x72 0x6E 0x61 0x6D 0x65 0x22
ASCII preview : {"username":"focyber21","emails":["focyber21@gmail ...
[18] **appVersionNumber** = 5.20.0
[19] **versionSpecificTaskRecords2** Save bytes
[20] **userEmail** = focyber21@gmail.com
[21] **timestampLastLoginUserUpdate** = 04-Feb-21 10:00:12 PM
[22] **username** = focyber21
[23] **userProfileImage** Save bytes
[24] **numberOfPosts** = 3
[25] **CAMUserPreferenceDidMigrate** = True
[26] **totalActiveSessionTimeInMs** = 92782
[27] **currentSessionStartDate** = 04-Feb-21 11:00:30 PM
[28] **initialActivityFeedTab** = 0
[29] **numberOfLikes** = 4
[30] **installID** = 523C9CC7-3640-4178-A31F-AAF2894F11D6
[31] **snsEndpointArn** = arn:aws:sns:eu-west-1:417173853888:endpoint/APN
[32] **IBGAppVersionHistory**
[0] 5.20.0 (45431)
[33] **soundsHaveMovedTutorialSeenCount** Save bytes

Figure 17. Dubsmash user profile and usage details recovered from iOS.

name ▲1	set_timestamp	value	origin
Filter	Filter	Filter	Filter
username	1612479630.98369	focyber21	app
push_access_given	1612479630.98384	true	app
num_posts	1612479630.98382	2	app
num_follows	1612479630.9838	1	app
num_followings	1612479630.98381	4	app
internet_connection_type	1612479630.98379	wifi	app
build_number	1612479630.98375	45431	app
app_version	1612479630.59151	5.20.0	app
account_age	1612479630.9846	0	app

Figure 18. User profile attributes as recovered from the Dubsmash iOS app.

4.3. Triller Forensic Artifacts

This section represents Triller-related findings such as what paths to look for specific artifacts, a directory tree of a few important sub-directories, and division of artifacts into two sub-sections (based on OS): Android and iOS.

4.3.1. Triller Android Artifacts

When analyzing the Android phone to investigate the Triller app, all Triller-related files on the Android device were stored under the path shown in Table 7. Therefore, all the recovered artifacts listed in Table 8 start with this base location.

Table 7. Triller app package paths.

OS	Package Path
Android	`app\co.triller.droid-SRvrgEx7zBOjSRrwr8XxdQ==\base.apk`
iOS*	`FullFileSystem.1.dar\private\var\mobile\Containers\Data\Application\123AF72F-5457-4018-8A51-74675C385DF0`
iOS**	`FullFileSystem.1.dar\private\var\Containers\Shared\AppGroup\D24052E8-AAC9-4742-B73C-F481D51E0915`

Table 8. Triller Android and iPhone artifacts.

S.No.	Artifact Type	Android Location	Artifact Data Value
1	Account Username	\system_ce\0\accounts_ce.db	focyber86
2	Account Email Address	\system_ce\0\accounts_ce.db	focyber86@gmail.com
3	UserID for focyber86	\system_ce\0\accounts_ce.db	601c505b569aa78a64c6dcfb
4	Account Picture Profile Symbol	\system_ce\0\accounts_ce.db	profile picture symbol: "P"
5	Account Registration date	\system_ce\0\accounts_ce.db	02/04/2021 10:18:40 PM
6	Account's Bio (Profile)	\system_ce\0\accounts_ce.db	hi, i'm sweetfire21 triller bio account
7	Account Other Information	\system_ce\0\accounts_ce.db	liked videos, followers, blocked, and following count
8	Android account username	\system_ce\0\accounts_ce.db	Focyber86
9	Android Triller App ID	SqualkDatabaseV4.sqlite\json_messages_table2	7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app
10	Android account registration date	\system_ce\0\accounts_ce.db	01/26/2021 7:07:50PM
11	Message ID with Focyber21	data\co.triller.droid\databases \7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad_Squalk_7_DB	601c781b4c4f8618b7d70bc9
12	Exchanged messages in chat	data\co.triller.droid\databases \7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad_Squalk_7_DB	Text: Another sweetfire21 production...
13	Video #1 (Music Video)	data\co.triller.droid\cache\video_cache\1\518.0.1612478529621.v3.exo	Music video uploaded at 5:46 pm Triller
14	Video #2 (Sweetfire Private Video)	media\0\Android\data\co.triller.droid\files\SDK_TRILLER_FILES \7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad	This is a private video sweetfire21 Triller video created by @focyber86
15	Video #3 (Recovered from Cloud)	\files\projects\c3836662-9788-46f9-9646-5dd0f4e81de4\takes \1612479717050\clips\1612479717050\video.mp4	video.mp4
S.No.	**Artifact Type**	**iOS Location**	**Artifact Data Value**
1	iPhone's Account username	**\Library\Preferences\group.com.triller.projectx.plist	focyber21
2	Android's Account username	**\Library\Preferences\group.com.triller.projectx.plist	focyber86
3	Account's Avatar URL	*\Library\Caches\com.triller.projectx\Cache.db	https://uploads.cdn.triller.co/v1/avatars/416923600/1612478290_avatar.jpg
4	Account Registration Date and Time	*\Library\Caches\com.triller.projectx\ Cache.db\cfurl_cache_receiver_data	04/02/2021 at 1612477130329 (5:18:50.329 PM)
5	Account Bio (Profile)	*\Library\Caches\com.triller.projectx\Cache.db	This is my sweetfire21 triller bio
6	Chats with focyber86 (Location 1)	*\Library\Caches\com.triller.projectx\ Cache.db\cfurl_cache_receiver_data	
7	Chats with focyber86 (Location 2)	**user_6c4e41f1-e4d3-4303-ad59-443c6694f27e\ SqualkDatabaseV4.sqlite\json_messages_table2	
8	UserID of focyber21 (iPhone user)	-Same as above -	601c72ca5cb7c3b307bcbee1
9	UserID of focyber86 (Android user)	-Same as above -	601c505b569aa78a64c6dcfb
10	Triller ID of focyber86 (Android user)	-Same as above -	7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app
11	Triller ID of focyber21 (iPhone user)	-Same as above -	6c4e41f1-e4d3-4303-ad59-443c6694f27e@triller.app
12	Draft Message	*\Library\Preferences\group.com.triller.projectx.plist	
13	Video shared over chat	** \user_6c4e41f1-e4d3-4303-ad59-443c6694f27e\files\videos	601c7ac2ab747f6af021552d_VIDEO_20210204_175244.mp4
14	Audio shared over chat	** \user_6c4e41f1-e4d3-4303-ad59-443c6694f27e\files\audios\	audio_gNdECAW7_DTS_04-02-2021_DTE.wav
15	Image shared over chat	** \user_6c4e41f1-e4d3-4303-ad59-443c6694f27e\files\images\	601c7916d58e35b03d4c7089_IMAGE_20210204_174542.jpg
16	Video #4(Private Video)	*\Library\Caches\com.triller.projectx\Cache.db	This is a private video sweetfire21 Triller video created by @focyber21

*\..\123AF72F-5457-4018-8A51-74675C385DF0\. **\..\D24052E8-AAC9-4742-B73C-F481D51E0915\.

The first line of evidence recovered is the registered user, in our case *focyber86* from the database file *accounts_ce.db*. The unique ID assigned to this user was *7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app*. The registration email *focyber86@gmail.com* was used during the population phase, also recovered from the same database. The account was verified to be created on 2 April 2021. Triller stores the timestamps in epoch time format, and to convert this, an epoch convert was used each time [24]. Additional account information such as username, profile picture, profile picture URL, registration date, bio, and date of birth were all found in one file. DB Browser for SQLite was used to better view and understand this database file. Additionally, similar to other apps, Triller also maintains a preferences file at `data\co.triller.droid\shared_prefs\main_preferences.xml` and `data\co.triller.droid\shared_prefs\co.triller.droid_preferences.xml` that contains information pertaining to the user account, bio, user preferences, profile picture URL, etc.

Once the account was set on Triller, *focyber86* was able to interact with other users on the platform. A profile was put together for this research, *focyber21* for the iPhone user of Triller. The files and messages shared during their conversations with Android users were recovered from the database as seen in Figure 19.

	message_id	roomID	status	json	created	type	text	deleted
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	601c781b4c4f86...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c781b4c4f8618b7d70bc9","attributes":{"trillerVideoUrlData":...	1612478491654	1	Another sweetfire21 production...	0
2	601c783708cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c783708cbd570267dda92","created":...	1612478519723	1	Hey	0
3	601c784008cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c784008cbd570267dda93","created":...	1612478528871	1	Sweetfire21 nice	0
4	601c785e7dd9bf...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c785e7dd9bf19dbc97667","created":...	1612478558292	1	Sweetfire21 triller messaging feature	0
5	601c787e08cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"image\/jpeg","id":"601c787c19453b6baddac214","size...	1612478590156	2	Here's my beautiful face sweetfire21	0
6	601c79026a750a...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"image\/jpeg","id":"601c79024f4173163b908701","size...	1612478722846	2	...	0
7	601c7918f1183c...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"image\/jpeg","id":"601c7916d58e35b03d4c7089","size...	1612478744184	2		0
8	601c793708cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"audio\/wav","id":"601c79371453666a7254d164","size"...	1612478775831	2		0
9	601c794c909d21...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"video\/mp4","id":"601c794a5cb7c3b307bcbee6","size"...	1612478796771	2		0
10	601c7955909d21...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"audio\/wav","id":"601c795599dcc6861a2a89fb","size":...	1612478805789	2		0
11	601c7967909d21...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c7967909d21b2fe12ae63","created":...	1612478823174	1	Audio video photo messages working...	0
12	601c7973909d21...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"image\/jpeg","id":"601c79728e36866f68126b51","size...	1612478835761	2		0
13	601c799108cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"video\/mp4","id":"601c79908e36866f68126b53","size"...	1612478865224	2	Sweetfire21 video of keyboard 😂	0
14	601c79a0909d21...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c79a0909d21b2fe12ae65","created":...	1612478880307	1	Just photos, video, audio, emojis ...	0
15	601c79da08cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"audio\/wav","id":"601c79da8bc4176b2622d759","size"...	1612478938602	2		0
16	601c79fe08cbd5...	1-601c505b569aa78...	0	{"currentProgress":0,"_id":"601c79fe08cbd570267dda99","created":...	1612478974305	1	Can't download vn sweetfire21	0
17	601c7ac6b354f9...	1-601c505b569aa78...	0	{"currentProgress":0,"file":{"file":{"mimeType":"video\/mp4","id":"601c7ac2ab747f6af021552d","size":...	1612479174207	2		0

Figure 19. The messages shared between focyber86 (Android user) and focyber21 (iPhone user) on triller.db as shown in the DB Browser for SQLite.

Understanding how the Triller app handles videos and their storage is incredibly important. Therefore, multiple videos (as per Section 3.1) were created to observe their storage hierarchy. A music video, private video, and video recovered from the cloud were discovered during the investigation. We were able to recover information from all three of these videos. For example, one of the music video was recovered from `data\co.triller.droid\cache\video_cache\1\518.0.1612478529621.v3.exo`. This video file included the time it was posted to the wall, the description, and the creation date. The authors verified the authenticity of this video by matching its creation date and time of 02/04/2021 at 5:46pm. The private video that was discovered (denoted by *private : "True"*) the user created within *media\0\Android\data\co.triller.droid\files\SDK_TRILLER_FIL ES\7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad*. This file includes the video description, creation date, and username. There is also audio throughout the video that states that this is a private *sweetfire* video. The last video that could be recovered was from the cloud. This was not posted on the wall. The user had created this video within `files\projects\c3836662-9788-46f9-9646-5dd0f4e81de4\takes\1612479717050\clips\1612479717050\video.mp4`. The artifact values pertaining to Android Triller application videos that were created can be found in Table 8.

Overall, the Triller app appeared to have many more in-app features than the Byte and Dubsmash apps. One of those features was the capability to save draft messages and recover them later in the chats, just like slack or discord. We saved such draft messages

and successfully recovered them from `data\co.triller.droid\shared_prefs\c9c-2c7b-4d5a-b5a7-62e6e8796aad_prefs.xml`. Secondly, Triller creates a unique .str for the social media activities associated with a given registered user. For example, the file *MyFeed.str* stores every feed from the user's home page, user's followers at *userfollowersrecommend.str*, user's follow activity at *activity_you.str*, trending hashtags at *hashtagstrending.str* and many more. Figure 20 can be referred to for a general overview of the file structure consisting of forensically important directories.

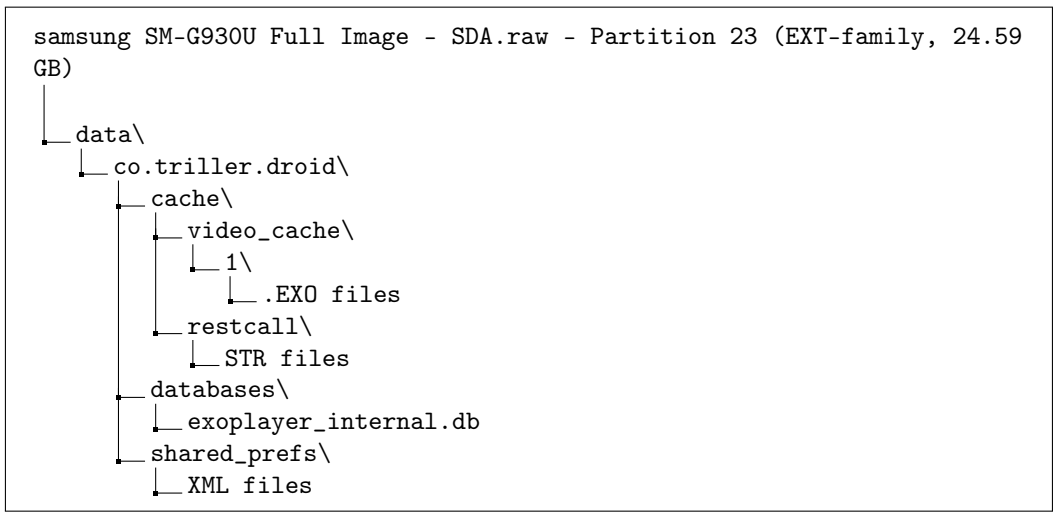

Figure 20. Triller Android file structure as captured in Magnet AXIOM Examine.

4.3.2. Triller iOS Artifacts

Upon investigating the Triller app on the iOS 13.3.1 device, it was observed that all the files related to Triller on the iPhone device are stored under the path shown in Table 7. Therefore, all recovered artifacts listed in Table 8 start at this base location.

Login and Profile Information

Starting from setting up the account on the device, the first artifact in the line of evidence is the registration date and time. The iPhone user account *focyber21* was created on 4 February 2021 at 1612468315 (14:51:55 (pm) and was recovered from multiple locations on Magnet AXIOM in the Triller folder.

An important aspect of the Triller app investigation was to find out the storage pertaining to chats, messages (draft and in-transit), photos, videos, and gifs thus communicated. We were interested in looking at whether these are stored in plain text or an encrypted format. The iPhone user *focyber21* interacted with the Android user *focyber86* and the communication was recovered in plain text in the two database files *Cache.db* and *SqualkDatabaseV4.sqlite* whose locations are listed in Table 8. For example, the text message *"Hey"* was exchanged between the two users at 1612478519723 (Thursday 4 February 2021 5:41:59.723 PM) as shown in Figure 21. Other important attributes collected were the chat creation time (epoch time), email (encrypted), password of the account (encrypted), Triller account ID, and avatar URL, among others. The Triller chat supported the exchange of photos, audio, video messages, and draft messages. Triller, among other apps investigated in this paper, is the only app found to be supporting saving draft message features. The recovered draft message is shown in Figure 22, and its path is listed in Table 8. The database file for each conversation with users stores that user's username, userID, chatID, messageID, mimeType(image/jpeg, video/mp4, audio/wav). The chatID was unique between two users, while messageID is a counter of messages exchanged.

All media that were exchanged during the chat conversation was recovered from `...\private\var\mobile\Containers\Shared\AppGroup\D24052E8-AAC9-4742-B73C-F481D51E0915\user\_6c4e41f1-e4d3-4303-ad59-443c6694f27e\files` folder. This *files* folder con-

tains four folders, one for each media type (audios, images, temp, videos) and within each of these folders, we recovered the voicenotes (*.wav* files), the pictures (*.jpg* files), and videos (*.mp4* and *.MOV* files) exchanged during the chat conversation. An example of each type is listed in Table 8.

{"code":1,"time":1612478520010,"data":{"list":[{"_id":"601c781be5db47377ccf1b34","chatId":"601c505b569aa78a64c6dcfb","userId":"601c72ca5cb7c3b307bcbee1","__v":0,
"chatType":1,"deleted":false,"lastMessage":{"messageId":"601c783708cbd570267dda92","message":"Hey","created":1612478519723,"type":1,"sentTo":[
"601c505b569aa78a64c6dcfb"]},"lastUpdate":1612478519732,"lastUpdateUser":{"muted":[],"trillerMuted":[],"blocked":[],"devices":[],"UUID":[{"UUID":
"9893F9F8-437A-4724-AD87-950B7DFCF55E","lastLogin":1612477700063,"blocked":null,"lastToken":"******","pushTokens":[
"d1e3672227b0f1cc842f13a4400300739ff13c606a1a924d010f5a4232d4c271"]}],"_id":"601c72ca5cb7c3b307bcbee1","name":"focyber21","sortName":"+70016124771290234","userid":
"6c4e41f1-e4d3-4303-ad59-443c6694f27e","created":1612477130329,"status":1,"organizationId":"5f23ff9be927763e3600fbe8","permission":1,"email":"+70016124771290234",
"__v":1,"device":"ios","tos":{"Result":"0","ResultTag":"CreateUserCompleted","ResultDescription":"Received verification code for new user.","User":{"UserName":
"6c4e41f1-e4d3-4303-ad59-443c6694f27e@triller.app","Password":"wOzAlKJEMj7I"},"tosusername":"6c4e41f1-e4d3-4303-ad59-443c6694f27e@triller.app","tospassword":
"wOzAlKJEMj7I"},"description":"","trillerLevel":"1","triller_avatar_url":"https://uploads.cdn.triller.co/v1/assets/chat-avatar-default.png",
"triller_thumb_avatar_url":"https://uploads.cdn.triller.co/v1/assets/chat-avatar-default.png","currentChatId":"1-601c505b569aa78a64c6dcfb-601c72ca5cb7c3b307bcbee1"
},"unreadCount":0,"lastUpdateMessageStatus":1612478517503,"lastUpdateUnreadCount":1612478518493,"user":{"groups":["5f23ff9be927763e3600fbea"],"trillerMuted":[],
"devices":[],"_id":"601c505b569aa78a64c6dcfb","name":"focyber86","sortName":"7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad","userid":"7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad",
"password":"******","created":1612468315205,"status":1,"organizationId":"5f23ff9be927763e3600fbe8","permission":1,"email":"+70016124683161881 6ez","tos":{"Result":"0"
,"ResultTag":"CreateUserCompleted","ResultDescription":"Received verification code for new user.","User":{"UserName":
"7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app","Password":"NKRZy1FGwQhs"},"tosusername":"7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app","tospassword":
"NKRZy1FGwQhs"},"__v":0,"description":"","triller_avatar_url":"https://uploads.cdn.triller.co/v1/avatars/416895195/1612477432_avatar_2021-02-04-17-23-12.jpg",
"triller_thumb_avatar_url":"https://uploads.cdn.triller.co/v1/avatars/416895195/1612477432_avatar_2021-02-04-17-23-12.jpg","currentChatId":"","onlineStatus":1}}],
"broadcastList":[]}}

Figure 21. The content of chat-related storage on Triller as shown in DB Browser for SQLite.

[2] **draft_601c505b569aa78a64c6dcfb_6c4e41f1-e4d3-4303-ad59-443c6694f27e** = This is a Draft sweetfire21 in triller

Figure 22. Draft chat message recovered from the iOS Triller app.

The user focyber21 followed four accounts, and the information was recovered from the path reported in Table 4. Each of the users followed has a unique ID assigned to them by Triller. For example, *7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad* is the userID and *7fc2dc9c-2c7b-4d5a-b5a7-62e6e8796aad@triller.app* is the userName of the Android user stored in the records. Interestingly, we were also able to recover information about the private video (denoted by *private : "True"*) the user created within the `Library\Caches\com.triller.projectx\fsCachedData\977F2E52-E579-444B-B311-95BACBB66C71` *.json* file. This file includes the video URL, description, creation date, and the video creator's details, including username and uuid. There are also headings for the video creator's email, date of birth, last seen, and location latitude and longitude. However, these fields all have a value of *NULL*. There is also a *name* heading, which has the value "*WiCys Forensics*".

5. Discussion

Going over the investigation, we found some similarities that are generally reported in an Android-based and iOS-based OS investigation. Here, we discuss some advantages and disadvantages posed by Byte, Dubsmash and Triller to mobile forensics.

- The previous literature on the analysis of apps (on Android and iOS) helped authors target specific data points, directories, and formats, thus saving crucial examination time. However, the analysis of these three apps still gave us expected yet unique data points, directories, and formats.
- Adoption of NIST guidelines for mobile forensics helped us formulate guidelines for these apps and provide an investigative framework for investigators who may encounter these or similar apps in the future.
- From the literature, we observed that Snapchat Forensics [21] also employs the usage of Magnet AXIOM for analysis. Similarly, our research provides an increased understanding of this tool and how it can be applied to moderately-known apps.
- The previous literature also shows the difficulty in recovering deleted content, especially videos, which is the unique selling point of these apps. By employing our proposed methodology, any researcher would be able to extract these data points that are otherwise reported under failure in data recovery.
- Some disadvantages are clearly present as well. At the time of beginning this research, the authors did not save the privacy policies present on the websites, as these are usually present in archives. However, in the surge of discussions around a potential TikTok ban, these apps drastically changed their policies without providing

access to their archive. Thus, comparing the websites and our found privacy policies proved difficult.

5.1. Implications for Research

This research contributes to the academic literature by conducting an extensive forensic analysis of three of the most popular mobile app alternatives to TikTok: Byte, Dubsmash, and Triller. Our findings indicate that the recovered artifacts highlight the wealth of recoverable user data. Particularly, the recovery of the user's private messages and media, and the extensive logs pertaining to how the user used the apps, all lend to the need for better privacy protections to be built into these and similar applications. With the increasing popularity of alternatives to TikTok, app developers often neglect to put proper security measures in place before putting the app on the market. A similar hike is observed in the mobile forensics literature that continues to forensically investigate these apps while simultaneously providing a checklist to fellow investigators. To this end, authors from [12–14] identified different privacy issues around the use of smartphone apps. We took these privacy concerns and used them as a checklist to compare against existing privacy practices exercised by Byte, Dubsmash, and Triller.

The first line of action that the authors highly recommend for all three apps is to start encrypting (1) user's personally identifiable information, (2) user's video posting data, (3) user's audio notes, and (4) user's in-app activity, which is otherwise recovered extremely easily. Images shared within direct messages are sometimes highly personal and sensitive in nature and would often leave the user embarrassed if they became publicly available. In our investigation, we could access images (including profile pictures and shared images) available several months after the forensic extraction was completed and were easily accessible by any internet browser. Since the majority user base of these apps is teenagers, it can be assumed that they are not aware of how much data is being sent and stored and what their data is being used for. These users (children or adults) would be devastated to learn that their privately shared images and conversations were recovered by any malicious hacker they had not consented to. As such, the authors also recommend further research be done by both ethicist and behavioral researchers to explore how the sustained use of these video sharing apps affect their users in terms of their privacy awareness and behavioral changes.

5.2. Implications for Practice

Some artifacts collected by Byte, Dubsmash, and Triller are particularly interesting from the forensic investigator's point of view. Dubsmash stored the authentication token for the signed-in user; this may be useful for forensic investigators should they need a method of signing into Dubsmash. However, going off the token expiration timestamp, the token was set to expire after 23 h of signing into the app. We were able to recover URL links of user-posted videos even after four months from the *\cache\http_cache* under the Dubsmash Android directory. Other attributes included a URL link to the audio from the video, video title, video creation date, number of likes and views the video has, and whether the app user liked the video (see Figure 11). The URL links were still live some four months later. Each video and the original sound were accompanied by the content creator's details, including the creator's DOB, uuid, username, display name, and a URL link to the creator's profile picture. Also of note was the recovery of plain text chat messages, including media such as voice notes and pictures, that the user exchanged with other users. Though these artifacts are a goldmine for forensic investigators, they also violate these app users' privacy significantly. The authors further recommend that such video-sharing apps offer privacy-preserving default settings, such as deleting all messages within 24 h, for example.

Magnet AXIOM Examine recovered videos and thumbnail images in their original unencrypted format; this is insecure data storage, and any potential hacker is only a sophisticated software away from looking at these personal artifacts. The inclusion of

security-conscious coding practices would greatly benefit both these apps' users as well as these apps' success in building or maintaining a transparent relationship with their users.

6. Conclusions and Future Work

In this research, researchers forensically investigated three popular mobile app alternatives to TikTok, given its recent labeling as a "national security concern." Byte, Dubsmash, and Triller mobile apps are popular video-sharing apps available on the Google Play Store and the iOS App Store. The researchers used Android 10 and iOS 13.3.1 versions on Android (Rooted) and iPhone (Jailbroken) smartphones, these devices being chosen purely out of convenience. In the following paragraph, we answer the research questions posed in Section 1.

From the perspective of digital forensics, it was essential to root and jailbreak these smartphones to get access to private directories. In the Byte app, we recovered the majority of the data that was populated on both the Android and iOS phones for the purpose of this investigation. For example, Table 4 enumerates our recovered artifacts, including both Android and iPhone account username, the user's date of birth, date and time of registration, profile activities (follower count, block accounts count, number of conversations initiated by the said user), video posting activity, and chat messages all in plain text. Similarly, in the Dubsmash app, Table 6 enumerates the username of both the Android and iPhone account, email address, the unique UUID of the said user, date of birth, date and time of registration, profile activities (follower count, block accounts count, number of conversations initiated by the said user), video posting activity, search queries, and chat messages all in plain text. Additionally, the authors also recovered the type of internet connection being used, the app version, and the age of the Dubsmash account. Similarly, Table 8 enumerates Triller's artifacts for both Android and iPhone's account username, email address, said user's unique ID, profile picture, bio details, date of birth, date and time of registration, profile activities (follower count, block accounts count, number of conversations initiated by the said user), video posting activity, and chat messages all in plain text. Additionally, these apps underwent stark changes during the course of their research, ref. [3] indicated that in a merger between Reddit and Dubsmash, the latter app's feature will be adopted in the former, removing Dubsmash from the app market at all mobile platforms. A rival video-creation platform bought the Byte app and now operates under the name Clash [25].

Author Contributions: Conceptualization, Y.K. and S.H.; methodology, Y.K. and S.H.; software, U.K.; validation, Y.K. and S.H.; formal analysis, Y.K., S.H. and A.S.; investigation, Y.K., S.H. and A.S.; resources, U.K.; data curation, Y.K.; writing—original draft preparation, Y.K., S.H. and A.S.; writing—review and editing, Y.K. and S.H.; visualization, Y.K., S.H. and U.K.; supervision, U.K.; project administration, U.K.; funding acquisition, U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

XML	Extensible Markup Language
FTK	Forensic Toolkit
OS	Operating System
HPC	Hardware Performance counter
COPPA	Children's Online Privacy Protection Act
GPS	Global Positioning System
ADB	Android Debug Bridge

NIST	National Institute of Standards and Technology
UFED	Universal Forensics Extraction Device
CFTT	Computer Forensics Tool Testing
GIF	Graphics Interchange Format
SIM	Subscriber Identity Module
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
Plist	Property List
JPG	Joint Photographic Expert Group
SQL	Structured Query Language

References

1. TikTok Revenue and Usage Statistics. Available online: https://www.businessofapps.com/data/tik-tok-statistics/ (accessed on 4 May 2021).
2. TikTok—Apps on Google Play. Available online: https://play.google.com/store/apps/details?id=com.zhiliaoapp.musically&hl=en_US&gl=US (accessed on 10 April 2022).
3. Why Dubsmash Is Shutting Down in February 2022? Available online: https://www.makeuseof.com/why-dubsmash-is-shutting-down/ (accessed on 12 December 2021).
4. Dziedzic, S. Australian Intelligence Agencies Investigate Chinese-Owned TikTok over Security Concerns. Available online: https://www.abc.net.au/news/2020-08-02/tiktok-under-investigation-in-australia-over-privacy-concerns/12513466 (accessed on 15 May 2021) .
5. Constine, J. How Dubsmash Revived Itself as #2 to TikTok. Available online: https://techcrunch.com/2020/01/31/dubsmash-songs/ (accessed on 10 May 2021).
6. How Much Time Do People Spend on Social Media in 2021? Available online: https://techjury.net/blog/time-spent-on-social-media/#:~:text=Anaverageuserspent2,of40minperday (accessed on 15 May 2021).
7. Indiablooms. Ban Chinese Video-Sharing Apps Like Likee and Tik Tok: Bangladesh Security Force Chief: Indiablooms—First Portal on Digital News Management. Available online: https://www.indiablooms.com/world-details/SA/29714/ban-chinese-video-sharing-apps-like-likee-and-tik-tok-bangladesh-security-force-chief.html (accessed on 23 July 2021).
8. Suleman, M.; Soomro, T.R.; Ghazal, T.M.; Alshurideh, M. Combating Against Potentially Harmful Mobile Apps. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2021), Settat, Morocco, 28–30 June 2021; Hassanien, A.E., Haqiq, A., Tonellato, P.J., Bellatreche, L., Goundar, S., Azar, A.T., Sabir, E., Bouzidi, D., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2021; pp. 154–173.
9. Ryan, F.; Fritz, A.; Impiombato, D. TikTok and WeChat.: Curating and Controlling Global Information Flows. 2020. pp. 1–70. Available online: https://s3-ap-southeast-2.amazonaws.com/ad-aspi/2020-09/TikTok%20and%20WeChat.pdf?7BNJWaoHImPVE_6KKcBP1JRD5fRnAVTZ= (accessed on 23 July 2021).
10. Parker, K.; Igielnik, R. What We Know about Gen Z So Far. 2020. Available online: https://www.pewresearch.org/social-trends/2020/05/14/on-the-cusp-of-adulthood-and-facing-an-uncertain-future-what-we-know-about-gen-z-so-far-2/ (accessed on 23 July 2021).
11. Khoa, N.H.; Duy, P.T.; Do Hoang, H.; Pham, V.H. Forensic analysis of TikTok application to seek digital artifacts on Android smartphone. In Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, Vietnam, 14–15 October 2020; pp. 1–5.
12. Domingues, P.; Nogueira, R.; Francisco, J.C.; Frade, M. Post-mortem digital forensic artifacts of TikTok Android App. In Proceedings of the ACM International Conference Proceeding Series, Online, 24–29 June 2020. [CrossRef]
13. Domingues, P.; Nogueira, R.; Francisco, J.C.; Frade, M. Analyzing TikTok from a Digital Forensics Perspective. *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.* **2021**, *12*, 87–115.
14. Neyaz, A.; Kumar, A.; Krishnan, S.; Placker, J.; Liu, Q. Security, Privacy and Steganographic Analysis of FaceApp and TikTok. *Int. J. Comput. Sci. Secur.* **2020**, *14*, 38–59.
15. Pandela, T.; Riadi, I. Browser forensics on web-based tiktok applications. *Int. J. Comput. Appl.* **2020**, *175*, 47–52. [CrossRef]
16. Ovens, K.M.; Morison, G. Forensic analysis of kik messenger on ios devices. *Digit. Investig.* **2016**, *17*, 40–52. [CrossRef]
17. Jadhav Bhatt, A.; Gupta, C.; Mittal, S. Network Forensics Analysis of iOS Social Networking and Messaging Apps. In Proceedings of the 2018 11th International Conference on Contemporary Computing, IC3 2018, Noida, India, 2–4 August 2018. [CrossRef]
18. Basu, K.; Hussain, S.S.; Gupta, U.; Karri, R. COPPTCHA: COPPA Tracking by Checking Hardware-Level Activity. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3213–3226. [CrossRef]
19. Salamh, F.E.; Mirza, M.M.; Hutchinson, S.; Yoon, Y.H.; Karabiyik, U. What's on the Horizon? An In-Depth Forensic Analysis of Android and iOS Applications. *IEEE Access* **2021**, *9*, 99421–99454. [CrossRef]
20. Hutchinson, S.; Shantaram, N.; Karabiyik, U. Forensic analysis of dating applications on android and ios devices. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 10–13 November 2020; pp. 836–847.

21. Alyahya, T.; Kausar, F. Snapchat analysis to discover digital forensic artifacts on android smartphone. *Procedia Comput. Sci.* **2017**, *109*, 1035–1040. [CrossRef]
22. Ayers, R.; Brothers, S.; Jansen, W. Guidelines on mobile device forensics (draft). *NIST Spec. Publ.* **2013**, *800*, 101.
23. Computer Forensics Tool Testing Program (CFTT). Available online: https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt (accessed on 23 June 2021).
24. Epoch Converter. Available online: https://www.epochconverter.com/ (accessed on 23 July 2021).
25. One-time TikTok Rival Byte Relaunches as Clash, an App for Video Creators and Their Top Fans. Available online: https://techcrunch.com/2021/10/12/one-time-tiktok-rival-byte-relaunches-as-clash-an-app-for-video-creators-and-their-top-fans/ (accessed on 30 August 2021).