forensic sciences

MDPI

Article

Forensic Analysis of the Bumble Dating App for Android

António Barros [1,*,†], Rafaela Almeida [1,*,†], Tiézer Melo [1,*,†] and Miguel Frade [1,2]

[1] School of Technology and Management (ESTG), Polytechnic of Leiria, Morro do Lena—Alto do Vieiro, 2411-901 Leiria, Portugal; miguel.frade@ipleiria.pt
[2] Computer Science and Communication Research Centre (CIIC), Polytechnic of Leiria, Morro do Lena—Alto do Vieiro, 2411-901 Leiria, Portugal
* Correspondence: 2202271@my.ipleiria.pt (A.B.); 2200330@my.ipleiria.pt (R.A.); 2200175@my.ipleiria.pt (T.M.)
† These authors contributed equally to this work.

Abstract: Mobile applications that facilitate interaction between people have grown in popularity and, as a result, the number of e-dating applications have expanded. In these types of applications, there is usually a trade-off between user privacy and safety. On one hand, users want to keep their data as private as possible, on the other hand, user identification forces accountability, which, hopefully, will foster the development of responsible behaviors and minimize abuses. The Bumble e-dating app has been growing in popularity and differs from other apps by giving women the power to make the first contact after a match. Their main goal is to prevent women's harassment. In this work, we study the digital artifacts that can be found after the use of the Bumble app on Android devices. Despite applying many measures to ensure data protection, it was possible to obtain information that identifies users and exchanged messages. The data structure stored on the device is described, as well as the artifacts with forensic value for an investigation. Moreover, a script was created to parse and visualize the main forensic artifacts of the Bumble app.

Keywords: digital forensics; Android; e-dating; Bumble

Citation: Barros, A.; Almeida, R.; Melo, T.; Frade, M. Forensic Analysis of the Bumble Dating App for Android. *Forensic. Sci.* **2022**, *2*, 201–221. https://doi.org/10.3390/forensicsci2010016

Academic Editors: Mário Antunes and Patrício Domingues

Received: 29 December 2021
Accepted: 23 February 2022
Published: 27 February 2022

1. Introduction

Through technological evolution, smartphones have become mini-computers, possessing the ability to store and process data, perform tasks of high computational value, and access a large amount of information through Internet access [1]. Consequently, smartphones store a trove of user information, such as location data, captured images, videos and audio, and interactions in multiple social networks, from entertainment to e-dating. As such, these devices are an essential part of a digital forensic analysis process.

Bumble is an e-dating app founded in 2014 [2], available for both Android and iOS. At a time when the term feminism is gaining traction, the Bumble app (https://bumble.com/, accessed on 28 December 2021) proposes to empower women by making "(...) not only necessary, but acceptable for women to make the first move, shaking up outdated gender norms" [3]. The name indicates that the app spins around the queen bee, in this case represented by the app's female users. For this reason, the company describes itself as 100 percent feminist, encouraging equality and "reversing the heteronormative rules surrounding dating" [4]. Bumble currently has more than 10 million downloads on the Google Play Store (https://play.google.com/store/apps/details?id=com.bumble.app, accessed on 28 December 2021), where its average rating is 3.9 in the opinion of approximately 280 thousands users. In the App Store (https://apps.apple.com/us/app/bumble-dating-meet-people/id930441707, accessed on 28 December 2021), its rating is higher, being 4.2 out of one million reviews. Bumble has around 42 million active users globally, of which 1.2 million subscribe to the application's premium features [5]. The United States has the highest number of users, with around 5 million, only surpassed amongst dating apps by Tinder, with approximately 8 million users [6]. In February 2021,

Bumble stated that its market value was 13 billion dollars, an increase from the previous year when it was valued at around 8 billion [5].

The Bumble app allows the user to perform various operations, such as: creating a profile, exchanging messages, images, and videos, searching for filters, and integrating with other social networks like Facebook, Instagram, and Spotify. To benefit from the features that Bumble offers, one must first create an account in the application. This process allows the authentication to be done from two sources: a mobile phone number, or through a Facebook account. When creating a Bumble account with the Facebook credentials, the users grant permission to share their names and profile pictures. Also, users can choose what additional permissions are granted by Facebook by opting out of granting access to: the users' email address (if there is one associated with the Facebook account); date of birth; profile photos; gender; page likes, and current town/city.

Like other e-dating applications, Bumble allows the creation of a profile, where users add information about themselves, such as a short biography, their interests, height, weight, religion, photos, geographic location, and gender. Furthermore, it is possible to insert activities that the user performs, such as whether a person is a smoker, has a habit of drinking, practices sports, among others. However, there is no mandatory information other than a name, the date of birth, and a mobile number or a Facebook account. It is noteworthy that when creating the profile, the user is requested to verify their profile by sending a photo in a pose illustrated in the application. This procedure has the purpose of reducing the creation of fake profiles, because of the extra work it would take to find someone's else picture in a pose that is unknown before the verification. Yet, the verification process is not mandatory.

Once the account is created, the application starts to present a stream of profiles for the user to choose from by swiping right (indicating interest on the shown profile) or left (if not interested). When a match occurs, that is, two people swiping right on each other's profile, Bumble invites one of them to send a message within 24 h. If it does not occur, the match will expire, and, therefore, the users will not be able to interact again. If the match is between two individuals with opposite gender, the female user must start the conversation. If the interactions are between users of the same gender, both can initiate the conversation [7]. While communicating, the application allows users to exchange text messages, photos, audio recordings, and video calls. If the user is not interested in the shown profile, they can swipe left. Henceforth, the user will no longer see the profile previously displayed, and, subsequently, the application recommends another potentially compatible user. The application also has a nude recognition mechanism. Through artificial intelligence, it processes the exchanged pictures to identify if they contain sexual content, also known as "nudes" [8]. In our limited test, we sent five distinct sexual pictures and the application was able to identify all of them accurately. This functionality is designed to prevent the delivery of unsolicited nudes. If a nude is delivered, the application hides the image and informs the user that it is an erotic image, asking the user to approve, or not, its reception. If preferred, the user may not receive the image and report the sender.

Besides the e-dating mode, Bumble also offers two extra modes in the application: the Best Friends Forever (BFF) and business (identified as BIZZ). The BFF mode focuses on establishing a match to create a new friendship, while BIZZ mode aims at finding professional connections.

As Bumble is a global application, all data is sent to, stored, and processed in the United States of America, and the United Kingdom, regardless of the user's country of residence. The privacy policy continues explaining how personal data is handled [9]. Table 1 shows the information that might be collected about a user when creating a Bumble account. Additionally, the application may request the user's full name and address to share with third parties, to send merchandise and loyalty programs. Besides registering the user's email, when the customer support team is contacted, Bumble can store the users' Internet Protocol (IP) address to keep track of customer communications and complaints concerning other users. Bumble processes collected data, such as demographic information,

to target advertising through in-app advertising and, in addition, shares data with ad networks that host its ads. Bumble may also collect information about the device, such as its unique identifier, model, operating system, Media Access Control (MAC) address, and, if authorized, may access the user's contact list.

Table 1. Information that might be collected about a user when creating a Bumble account.

Name	Username
Email address	Mobile number
Gender identity	Date of birth
Sexual preference	Photographs
Location	Login information for social media accounts (e.g., Facebook, Instagram, Spotify)

Bumble offers a free tier with a limited amount of profiles a user can view per day. The optional premium account offers additional features, such as Spotlight, SuperSwipe, Bumble Boost, and Bumble Premium. The Spotlight allows the user's profile to be viewed by more people instantly for 30 min, while the SuperSwipe notifies a potential match that the user is confidently interested in them. The Bumble Boost feature includes the possibility to backtrack (swipe right on a profile which the user swiped left previously), extend the time to answer on current matches, unlimited swipes, one Spotlight, and five SuperSwipes per week. The Bumble Premium service has all the Bumble Boost features and access to unlimited advanced filters, the user's admirers, travel mode, the rematch with expired connections, and incognito mode (not available on the Bumble's Web version).

The Android version of Bumble is updated regularly, about once a week. The available versions of the application are compatible with Android 5.0 and upper. So far, Bumble for Android's updates do not contain information that notifies the user of the modifications made in the latest updates. Nevertheless, the updates for the iOS app describe the changes made, which mainly consist of bug fixes. Bumble does not allow the usage of some of the prior versions, pushing the users to update it regularly to keep using the application. Our tests showed that it is only possible to use up to the 10th older version before the user is required to update. The version analyzed in this work refers to 5.250.1 released on 13 December 2021 (https://www.apkmirror.com/apk/bumble-holding-limited/bumble-date-meet-friends-network/bumble-date-meet-friends-network-5-250-1-release/, accessed on 28 December 2021).

Despite app developers' best efforts to protect users' information, e-dating services are not immune to cybercrimes, such as extortion, romance scams, and identity theft. Generally, attackers use e-dating applications to create a relationship with users, misleading them into sending money, personal, and financial information. According to the Internet Crime Complaint Center's (IC3) 2020 Internet Crime Report, extortion, identity fraud, and romance scams crimes were, respectively, the third, fifth, and eighth most commonly reported cybercrimes globally, creating a totaled victim loss superior to 900 million US dollars [10]. Therefore, identifying and analyzing e-dating applications' forensic artifacts can help to discover how a crime was conducted and uncover information that might lead to its perpetrators. There are several studies about e-dating applications, such as Tinder and Badoo [11–14], but, to the best of our knowledge, there is no in-depth analysis about Bumble in the digital forensics domain. Hence, the contributions of this work are: (1) a thorough study of the Bumble data stored in an Android device, and the analysis of its artifacts, (2) the diagram of the most forensic valuable database schema, and (3) a parsing script to present the most relevant forensic artifacts in a human-readable format.

This document is structured in the following way: Section 2 presents a literature review of works about mobile digital forensics analysis in general and e-dating apps. Then, Section 3 displays the testing methodology and tools used to conduct our study. The obtained results are presented and discussed in Section 4, followed by the description

of the developed parsing script in Section 5. Finally, the conclusions and future work are presented in Section 6.

2. Literature Review

The analysis of mobile applications and devices has been the subject of several literature reviews over the last few years. The forensic interest is due to the amount of information regarding the status of devices and data about the user, mobile systems, and applications use and storage [15]. Data about users' previous locations can be obtained from applications' digital artifacts, both on Android and iOS systems. This information contributes to an investigation with data relating to preceding suspect locations [16]. Another type of forensic artifact is the identification of files present in the cache, which are automatically generated by the applications and may contain important information about the user [17]. One type of application that can create relevant artifacts for forensic investigations are e-dating apps. E-dating applications might provide multiple artifacts, such as the Global Positioning System (GPS) location, telephone contacts, email addresses, and messages exchanged between users [14,18]. Additionally, Lcdi and Lcdi [19] performed a static analysis on an older version of Bumble for Android, where it was possible to identify images stored locally on the device [19].

Hayes and Snow [11] analyzed three e-dating apps (Tinder, Bumble, and Grindr) and identified that these were collecting and sharing personal information about their users that did not match the information stated in the privacy policy [11]. Additionally, the authors identified communication protocols that posed potential risks to the security of users' data. Knox et al. [20] performed an analysis of the Happn e-dating app for Android and iOS systems to identify artifacts that could be exploited by a malicious agent to gain access to confidential data of its users [20]. Furthermore, Kim et al. [12] analyzed five e-dating apps (Tinder, Amanda, Noondate, Glam, and DangYeonsi). Through traffic analysis and reverse engineering techniques they found that sensitive user data could be exploited by malicious agents [12]. Another study by Shetty et al. [13] carried out an analysis to assess the possibility of executing Man-In-The-Middle (MITM) attacks on seven e-dating apps (Tinder, Happen, Badoo, MeetMe, Skout, Lovoo, Coffe Meets Bagel, Chrome for Android, and Facebook). The authors identified that these were vulnerable to this type of attack, allowing access to the application's user data [13]. Also, a study from Farnden et al. [14] presents forensic techniques used to identify and retrieve data from the following e-dating apps: Badoo, Grindr, Skout, Tinder, Jaumo, Meet Me, FullCircle, and MuiMeet [14].

The literature review by Phan et al. [21] presents a different perspective, identifying the physiological impacts that e-dating apps can have on users. The authors also point out the risks related to digital security, possible crimes, and the digital artifacts of these apps that can be used to solve crimes [21].

Several e-dating applications have been studied to identify digital artifacts and determine which have forensic value. Nonetheless, to the best of our knowledge, the Bumble app has not yet been a target of an in-depth analysis in the digital forensic domain.

3. Methodology and Tools

The next subsections describe the methodology and tools used for the analysis of the Bumble mobile application.

3.1. Methodology

A forensic analysis of an application can be performed in a virtual environment, or a physical device. The use of a virtual device has several advantages: it is easily accessible; permits a quick reset of the system state in case of any difficulties; it enables the creation of copies of system versions created by the user, allowing tests to be run for each type of interaction or configuration change in the app or system. However, these advantages come at the cost of performance (all operations are slower than on a real Android device) and

practicality (Android was designed with a touch interface, and its use with mouse clicks can be cumbersome). For our tests, both alternatives were employed, as shown in Table 2.

Table 2. Testing devices.

Device	Android Version	Type	Details
AOSP API 30	Android 11.0	Virtual	With root permissions
AOSP API 28	Android 9.0	Virtual	With root permissions
Asus Zenfone 3 Max	Android 7.0	Real	No root permissions
Samsung A40	Android 10.0	Real	With root permissions

For the device virtualization process, we used the Android Virtual Device (AVD) Manager functionality is included in Android Studio. It allows for the creation and execution of the Android system emulator, enabling the creation of AVDs, which simulate the physical devices and where the Android system was configured to run the Bumble app. A set of operations were performed on all devices to simulate the actions made by a user when running the app, such as swipes, clicks, and message exchanges.

The Bumble app, version 5.250.1, was installed on several instances of the Android Open Source Project (AOSP) API 30 device, and three user accounts were created. Then, each of the three user profiles were configured so that each user could find the other ones. This way, our tests could be conducted without interfering with user accounts unaware of our tests. Afterward, a match between our accounts was done by swiping right on each other profiles. Then it was possible to maximize the number of interactions, including sending messages, audio, images, and video calls.

Bumble was also installed on the AOSP API 28 device (Android 9) since Frida is not yet available for Android 11. Therefore, we had to install an older version of an AVD that would support the execution of `arm64` instructions (Bumble is only available for this type of processor) on top of the Intel `x86` architecture used by the AVD itself. Only AVDs with Android version 9 (besides version 11) contain a mechanism that translates `arm64` instructions to `x86` in runtime.

Physical devices (Samsung A40 and Asus Zenfone) were also used in our tests because of the ease of interaction with the Bumble app. Additionally, one account was upgraded to a premium account for a week on the Samsung A40 to enable the identification of artifacts that could be exclusive to this version.

3.2. Tools

Table 3 lists all the used tools and their respective versions. The selected tools are either open source or free. Only HxD and Packet Capture (for Android) are not open source. Both of them are proprietary, but free, and were chosen for being easy to use. The following paragraphs describe their purpose and usage.

Android Studio (https://developer.android.com/studio, accessed on 28 December 2021) is an Integrated Development Environment (IDE) to develop Android applications, consisting of a code editor and other advanced development tools. Through its features, it is possible to execute virtual Android systems—Android Virtual Devices (AVD)—and the recompilation of the code to change the behavior of the application. Thus, AVD was used to perform the virtualization of Android devices to be applied in the forensic investigation environment.

The Android Debug Bridge (https://developer.android.com/studio/command-line/adb, accessed on 28 December 2021) (ADB) is a command-line tool to carry out the communication between a computer and the Android device, to execute commands, and interact with the Android OS. This tool was used to install Android Application Packages (APKs), to navigate in the Android directory structure, to run tools like Frida, and to extract all the data produced by the Bumble app to later be analyzed.

Table 3. Tools used to analyze the Bumble app for Android.

Tool Name	Version
Android Studio	2020.3.1
Android Virtual Devices (AVD) manager	30.4.5.0
Android Debug Bridge (ADB)	1.0.41
SchemaCrawler	16.16.4
dbdiagram.io	(online tool)
DB Browser	3.12.2
HxD	2.5.0.0
Autopsy Forensic Browser	4.18.0
Frida	14.2.14
Packet Capture	1.7.2
OpenSSL	1.1.1
Mobile Security Framework (MobSF)	3.4.3

The SchemaCrawler (https://www.schemacrawler.com/, accessed on 28 December 2021) is a tool to automate the database schema design process. This open-source tool aims to discover and draw the schema of databases, allowing the search for objects within the schema and illustrating its result in a text format [22]. Furthermore, SchemaCrawler generates diagrams of these schematics through Graphviz (open-source graphics visualization software). This tool was used to help understand the structure of the databases created by Bumble. However, not all tables contain references regarding their foreign keys and relationships, and therefore, automated tools, like SchemaCrawler, are not able to make a correct representation of the database schema. Subsequently, the dbdiagram.io (https://dbdiagram.io/home, accessed on 28 December 2021) web application was used to edit and redefine these schemas. This tool allows to draw entity-relationship diagrams by writing code.

The DB Browser for SQLite (https://sqlitebrowser.org/, accessed on 28 December 2021) is an open-source tool for viewing, creating, and editing SQLite compatible database files. This tool was used to visualize the contents of the SQLite database files of the Bumble application for Android.

The HxD (https://mh-nexus.de/en/hxd/, accessed on 28 December 2021) tool is a hexadecimal viewer and editor for the Windows operating system. This tool opens and edits files in hexadecimal format and displays and changes the memory used by running processes. This tool was used to visualize the content of the Bumble binary files.

The Autopsy Forensic Browser (https://www.autopsy.com/, accessed on 28 December 2021) is an open-source tool that simplifies the digital forensic analysis of many files and file systems. The graphical interface provided by the tool allows the user to easily obtain and visualize forensic artifacts. As such, Autopsy was mainly utilized to help view the content of Bumble files and search for keywords.

Frida (https://frida.re/, accessed on 28 December 2021) is a free toolkit used by security researchers for reverse-engineering purposes and to inject code into GNU/Linux, macOS, Windows, Android, iOS, and QNX applications (native apps). Frida is based on a client-server architecture, which means it is necessary to install it both on the host operating system and the device (physical or virtualized). The tool allows the observation and manipulation of software while running on a device. Frida has various use cases in which the most common include bypassing in-app protections, such as certificate pinning, or Android root detection. In our tests, the Frida server is executed on the Android phone, where root permissions are required, and the client runs on a computer. This tool was used, in conjunction with Packect Capture to observe the network traffic of the Bumble app.

The Packet Capture (https://play.google.com/store/apps/details?id=app.greyshirts.sslcapture, accessed on 28 December 2021) is an Android app that acts like a proxy and sniffer, capable of capturing network traffic and deciphering Secure Sockets Layer (SSL)/Transport Layer Security (TLS) packets, employing a man-in-the-middle technique. This

app allowed the capture of packets sent and received by the Bumble app and to see its content.

OpenSSL (https://www.openssl.org/, accessed on 28 December 2021) is an open-source library that is used in tasks related to cryptography and secure communication. In this work, OpenSSL was used to generate a digital certificate for the Packet Capture software to intercept the TLS encrypted traffic sent by Bumble app.

Finally, the Mobile Security Framework (MobSF) (https://github.com/MobSF, accessed on 28 December 2021) is an automated software that performs static analysis of mobile applications, both Android and iOS. It performs malware analysis and contains a security assessment framework to assist in performing a static analysis of an application. To obtain a full Bumble analysis, this tool was used in the static analysis, capturing the relevant data in a report format.

4. Results

After conducting several interaction tests with our accounts, we extracted the data from both the public folder (`/sdcard/Android/data/com.bumble.app`) and private folder (`/data/data/com.bumble.app`). The Bumble public folder contains two folders: `cache` and `files`, both without any content.

The most valuable data for the forensic analyst resides in the application's private directory: `/data/data/com.bumble.app`. To read the contents of this directory, the device (physical or emulated) must be rooted. To quickly acquire data while testing, a bash script was developed that automates the acquisition operations of the private directory. Bumble's private directory hierarchy is illustrated in Figure 1.

```
com.bumble.app/
├── app__animations/
│   └── bumble_relationship_profile_builder_start/
├── app_sslcache/
├── app_textures/
├── app_webview/
│   └── Default/
├── cache/
│   ├── decorator_tmp/
│   ├── downloader/
│   ├── paywall_disk_cache/
│   ├── sentry-buffered-events/
│   └── WebView/
├── code_cache/
├── databases/
├── files/
│   ├── AFRequestCache/
│   └── splitcompat/
├── no_backup/
├── oat/
│   └── arm64/
└── shared_prefs/
```

Figure 1. Bumble's private directory hierarchy.

When Bumble's app is uninstalled, all folders and files created by the app are deleted, which is the normal behaviour of the Android operating system. Then, if the app is reinstalled, and the same account is configured, all previous data is restored from Bumble's servers. However, if the Bumble account is deleted, no data will be restored, even if the user repeats its previous account credentials.

In the following paragraphs, the directories that were considered most relevant at a forensic level are addressed, namely: `databases`, `shared_prefs`, `files`, and `cache`. The remaining directories are not addressed, as they did not reveal any data with forensic value, or the content could not be identified due to unknown encoding.

4.1. Databases

Bumble contains five databases, four of which are in the `databases` directory (`ChatComDatabase`, `com.google.android.datatransport.events`, `google_app_measurement.db` and `lexems.db`). The `Cookies` database is present in the `app_webview/Default` directory, while `androidx.work.workdb` is in the `no_backup` directory. Following, the tables and fields that we consider most relevant from a forensic perspective are discussed. The timestamps in databases are in an Epoch format (representation of time in operating systems, such as Unix, Linux, and Android, represented by the number of seconds elapsed since 0:00 of 1 January 1970).

The `ChatComDatabase` database contains the most extensive number of tables and mostly stores information about the user's messages and conversations. The 18 tables relating to this database are listed in Table 4, and a summary of the database schema is illustrated in Figure 2.

Table 4. List of tables present in `ChatComDatabase`.

Tables	
`android_metadata`	`conversation_info`
`gif`	`group_chat_preload_queue`
`group_chat_sync_state`	`live_location_sessions`
`message`	`message_read_info`
`offline_message_read_info`	`search_fts`
`search_fts_content`	`search_fts_docsize`
`search_fts_segdir`	`search_fts_segments`
`search_fts_stat`	`sending_info`
`sqlite_sequence`	`url_preview`

A conversation contains a set of messages between two users of the application. The table that holds information about conversations is named `conversation_info`, and each table entry (row) represents a user conversation. The table fields identified as having forensic value are illustrated in Table 5.

Table 5. Attributes with forensic value on the `conversation_info` table.

Attribute	Type	Description
`user_id`	text	User identifier
`gender`	integer	Gender
`user_name`	text	Username
`user_image_url`	text	User profile image URL
`user_deleted`	boolean	0—user not deleted, 1—user deleted
`age`	integer	Age
`game_mode`	integer	Application Mode: 0—Bumble Date, 1—Bumble Friends, 5—Bumble Bizz
`encrypted_id`	text	Encrypted user identifier
`user_photos`	text	User profile photos
`photo_id`	text	User profile photo identifier
`common_interest_count`	integer	Number of common interests

The tables that store information about messages are designated as `offline_message_read_info`, `message_read_info`, and `message`. Contextually, messages refer to any data sent and received, such as text, images, gifs, and audio. The main fields of the `message` table with forensic value are described in Table 6. Each entry in the table represents a message sent or received in one of the user's conversations. It is noteworthy to mention that even if a user blocks another user, the messages exchanged will still be available in this table.

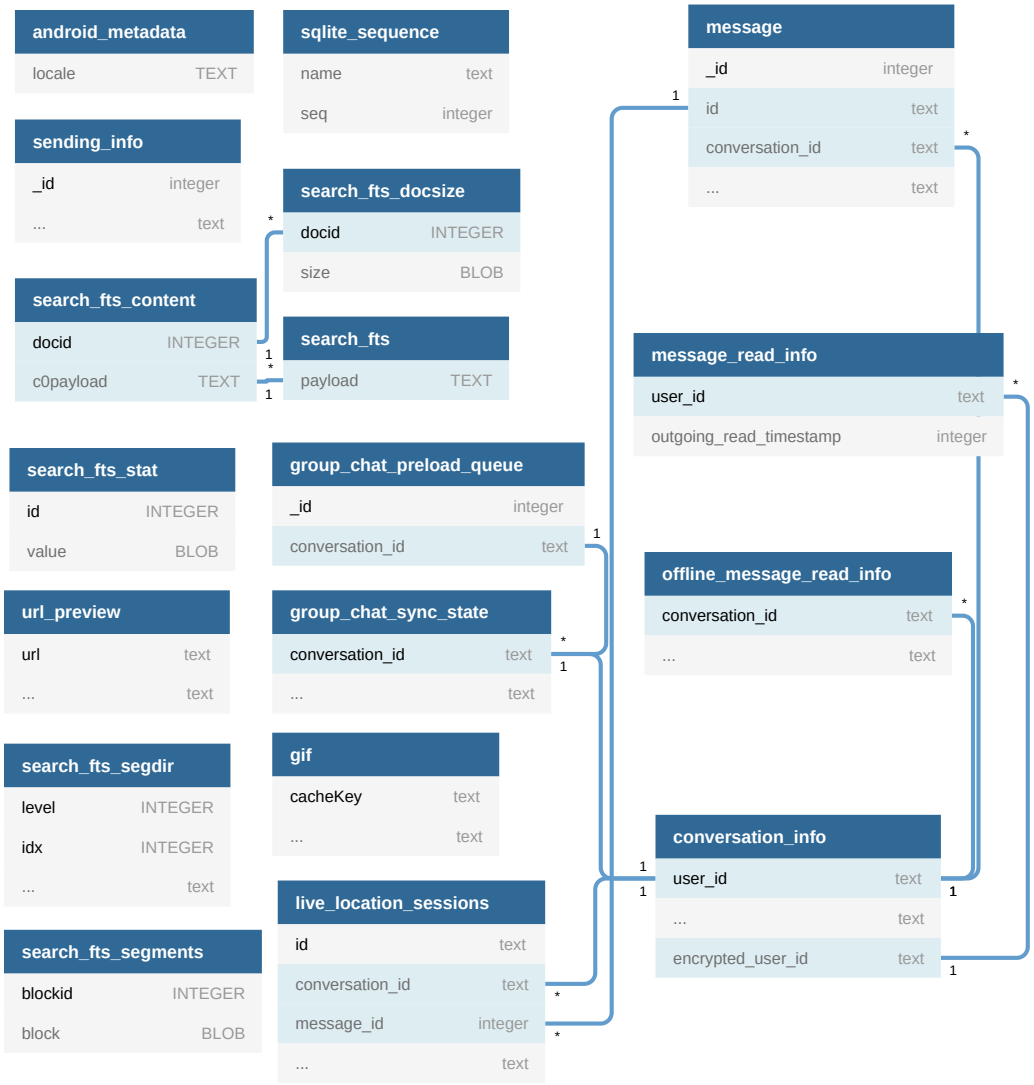

Figure 2. Summary of `ChatComDatabase` database schema. Its complete structure is available from the following link: https://github.com/rafsalmeida/Bumble_Database_Schema, (accessed on 28 December 2021).

The `payload` field is represented through a JavaScript Object Notation (JSON) object, which has different characteristics depending on the information present in the message. In addition to the information already available in the database, sending images, audio, and GIFs provide additional information. When sending a picture, data such as height, width, id, expiration timestamp, and if the image is masked (`is_masked`) are stored. The image itself is stored online, on Bumble's servers, and only the URL and its metadata are stored on the local database. When an image is marked as masked, it means that it is blurred by displaying it at a markedly lower resolution to the point that its content is not recognizable. Furthermore, an additional field is saved: `is_lewd_photo`. This field is a boolean, which displays as true if the image content is classified as sexual and false otherwise. Usually, whenever a sexual image is sent, it is camouflaged and identified as sexual, allowing the user to choose whether to view it. Sending an audio message stores its id, expiration timestamp, duration, and waveform. The latter represents, through a numerical vector, the change in amplitude over time. When sending a GIF, the application stores two pieces of information, these being its Uniform Resource Locator (URL) and its provider.

Table 6. Attributes with forensic value on the message table.

Attribute	Type	Description
id	text	Message identifier
conversation_id	text	Conversation identifier
sender_id	text	Encrypted sender identifier
recipient_id	text	Encrypted recipient identifier
created_timestamp	integer	Message creation identifier
modified_timestamp	integer	Message modification identifier
is_masked	integer	0—image not masked, 1—masked image
payload	text	Message payload
is_reply_allowed	boolean	0—unable to reply to the message, 1—able to reply to the message
is_fowarded	boolean	0—message to fowarded, 1—fowarded message
is_forwarded_allowed	boolean	0—not possible to forward a message, 1—possible to forward a message
is_incoming	boolean	0—message sent, 1—message received
payload_type	text	Payload type: text, gif, image, question_game, video_call, audio and offensive

Sending images, audios, and GIFs only stores their URL on the local database. However, only GIFs can be visualized outside the Bumble's app, such as in a browser. An attempt to view the exchanged private media always returns the HTTP error "403 Forbidden". Nevertheless, the user's profile photos specified in `user_image_url` and `user_photos` fields in the `conversation_info` table can be viewed in a browser until their URL expires.

In addition to the aforementioned tables, the database contains a virtual table created using the Full Term Search (FTS4) extension. This is an extension that helps the creation of virtual tables with a built-in full-text index. Through this method, it is possible to manually create a virtual table that maps a JSON document schema [23]. This table is named `search_fts` and includes a payload field that contains the text messages sent in multiple conversations, however, it does not include information on images, audios, and GIFs sent.

The `google_app_measurement.db` database, despite the tests carried out, was exclusively populated with relevant data in one table named `apps`. The table stores the application version in the field `app_version`. It could be useful to determine the last version the user executed.

4.2. XML Files

The Bumble application uses XML (Extensible Markup Language) files to configure and store information and application features. These files are located in the `shared_prefs` subdirectory. This directory had a total of 36 XML files, 10 of which had a forensic value and are listed in Table 7.

Table 7. XML files with forensic value.

Files	
appsflyer-data.xml	ServerCommunicationPreferences.xml
BumbleAppPreferences.xml	com.facebook.AccessTokenManager.SharedPreferences.xml
DeviceUtil.xml	RatingRulesCriteriaParams.xml
BumbleAppPreferences.xml	com.facebook.login.AuthorizationClient.WebViewAuthHandler.TOKEN_STORE_KEY.xml
HotLexemPrefs.xml	VOTING_QUOTA.xml

The contents of the files are briefly presented below, as well as a consideration of why each was classified as being of forensic interest. Note that the files `appsflyer-data.xml` and `com.google.android.gms.measurement.prefs.xml` have already been addressed in other literature [24] and refer to functions not related to the operation of the application itself.

The file `appsflyer-data.xml` is related to the AppsFlyer Software Development Kit to collect statistical information for advertisers and advertising campaigns. It displays the information related to the IMEI (International Mobile Equipment Identity) of the device on which the application is running. The IMEI-related fields can be useful to identify the phone which executed the application [24]. However, in all our tests the IMEI values were set to `false`, meaning they were not collected. The file also presents some information related to timestamps in the `appsFlyerFirstInstall` field, referring to when the app was first installed.

The `com.google.android.gms.measurement.prefs.xml` file is a collection of Google APIs that contains classes to configure Firebase Analytics core services to support functionalities across multiple devices [25]. The file is of forensic interest due to some information related to timestamps when in the `first_open_time` field, referring to when the app was first opened.

The `BumbleAppPreferences.xml` file shows, in the attribute `MyCurrentUserStateKEY_GAME_MODE`, the information of the current App mode selected by the user. When *dating mode* is active, the stored value is `GAME_MODE_REGULAR`. This option can also have the values `GAME_MODE_BFF`, for those who select the *looking for friends* option, and `GAME_MODE_BIZZ`, which corresponds to the users that selected the *business connections* option.

In the file `com.facebook.AccessTokenManager.SharedPreferences.xml` it is possible to identify a considerable amount of information regarding the Facebook account linked to the application. The identifiers referring to the Facebook token stand out, bringing a set of information, such as the registration or token code, the permissions within Facebook assigned to this token (which can be accessed on Facebook when using this token), the application that is using it (code `428250913904849` refers to Bumble), the account code to use, and the token and its expiration date. In the same file, it is possible to identify the name of the Facebook account to which the token is linked, and which is related to the id. A content sample of this file is shown in Listing 1.

The file `com.facebook.login.AuthorizationClient.WebViewAuthHandler.TOKEN_STORE_KEY.xml` contains the Facebook authentication token to be used by the application. This token allows the application to access the user's Facebook account information, which was previously linked to Bumble's account. The value of `expires_at` (line 7 of Listing 1) as the value `9223372036854775807`, which corresponds to the maximum value of the 32 bits signed integer (0x 7F FF FF FF FF FF FF FF). This suggests that the token never expires.

The `DeviceUtil.xml` file presents some information that identifies the device on which the application is installed. In this sense, the `DeviceId` attribute is present. Also, there is the attribute `FirstLaunch`, which shows if it is the first execution of the application on the device. The third attribute of interest is `DeviceIdStored`, however it was not possible to identify the information referring to it.

In the `HotLexemPrefs.xml` file, the version of the application running on the device can be found. As previously informed, the version of the application used is 5.250.1. However, it can also be referenced by the number 2618, as it is present on websites that download APK files. Also, it was possible to identify the APK's creation date as well as the installation date.

The `RatingRulesCriteriaParams.xml` file appears to be related to the number of times the application has been executed and the number of conversations with other users.

The `ServerCommunicationPreferences.xml` file shows the address of the hosts and possible ports considered to be secure, and which should be used in the application's communication with the server. It is noteworthy that the file does not present information regarding the protocol to be used by the application in establishing communication

with the server (TCP or UDP). The mentioned addresses and the respective ports are `bmaeu.bumble.com:80`, and `bmaeu.bumble.com:443`.

Listing 1. Sample content of `com.facebook.AccessTokenManager.SharedPreferences.xml` file (redacted).

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="com.facebook.AccessTokenManager.CachedAccessToken">
{
"&quot;version&quot;":1,
"&quot;token&quot;":&quot;EAAGFfeZBZAGNEBABwgxlQ8sgKZC4udy9zCkw4RV9gg4olAQg0ZCmUQ22ijaQQfZBv
↪ ZBjsRvlbNXbCngTXAKBv8HQhN9FrnsKzxRsT9ry5ImeGeZBo1sz7JWRLatqKronVzKfLI49ZBi1qRq7dNaI4zvuV
↪ Uv7SZAoV0IOrNCZBmx2cZA9ZCVYHeDSPvZAD7fxe1RlzoyWDb4YJBkJDAIVoxBYUwuhz3T1DKLp9NmQHOP0lpyrM
↪ TngBCZAzcqAvp&quot;,
"&quot;expires_at&quot;":9223372036854775807,
"&quot;permissions&quot;":[
"&quot;user_photos&quot;",
"&quot;user_birthday&quot;",
"&quot;user_likes&quot;",
"&quot;public_profile&quot;",
"&quot;user_gender&quot;",
"&quot;user_location&quot;",
"&quot;user_friends&quot;",
"&quot;email&quot;"],
"&quot;declined_permissions&quot;":[],
"&quot;expired_permissions&quot;":[],
"&quot;last_refresh&quot;":1639938823171,
"&quot;source&quot;":"&quot;CHROME_CUSTOM_TAB&quot;",
"&quot;application_id&quot;":&quot;428250913904849&quot;,
"&quot;user_id&quot;":&quot;120xxxxxxxxxxx&quot;,
"&quot;data_access_expiration_time&quot;":1647714825000,
"&quot;graph_domain&quot;":"&quot;facebook&quot;"
}
</string>
<string name="com.facebook.ProfileManager.CachedProfile">
{
"&quot;id&quot;":&quot;120xxxxxxxxxxx&quot;,
"&quot;first_name&quot;":"&quot;First Name&quot;",
"&quot;middle_name&quot;":"&quot;&quot;",
"&quot;last_name&quot;":"&quot;Last Name&quot;",
"&quot;name&quot;":"&quot;First Name Last Name&quot;",
"&quot;link_uri&quot;":"&quot;&quot;"
}
</string>
</map>
```

The `VOTING_QUOTA.xml` file presents the information corresponding to how many swipes the user can perform on the same day. This file can be considered as a possible indicator that the user has a premium account, as the value corresponding to the `KEY_YES_VOTING_QUOTA` field is much higher in the premium than that of the free version of the application. The premium version has a value in the billions whereas in the free version the value starts with 80. The value of the free version can increase if the user stays more than one day without swiping.

The `c2V0dGluZ3M=` file stands out because its name stands for *settings* after being base64 decoded. In the analysis performed it was not possible to have full access to the content of this file, as it was not possible to identify its encoding or if it was encrypted. With the use of UTF-8 encoding, it is possible to access part of the content, including some information about the user, which is presented in Table 8. The premium version of Bumble contains additional information that is marked with (*) in the aforementioned table. Figure 3 shows a capture of the partial mobile number using the HxD hexadecimal editor.

```
00000B10 08 01 12 0C 50 68 6F 6E 65 20 6E 75 6D 62 65 72  ….Phone number
00000B20 1A 05 41 64 64 65 64 22 0D 2B 33 35 31 39 31 37  ..Added"+351917
00000B30 2A 2A 2A 2A 38 32 28 01 30 01 40 00 50 02 62 09 ****82(.0.@.P.b.
00000B40 39 31 37 2A 2A 2A 2A 38 32 6A 03 33 35 31 82 01 917****82j.351,.
```

Figure 3. Phone number capture using the HxD software.

Table 8. List of user information that can be retrieved from the settings file, the symbol (*) marks information only available in the Premium accounts.

User Information
Location, specifically the region
Username
Birthday
Age
User verification status
User ID
Email associated with the account
Instagram link associated
Link to user profile picture
Partial mobile number, which only exists if a mobile number is associated
User's preferred language
(*) Premium expiration time
(*) Travel location, when travel mode is enabled

It is possible to conclude that the file `files/c2V0dGluZ3M=` contains information relating to user data. It is important to note that the location is not reliable, as it can be manually entered by the user. The travel mode present in the premium version does not change the data referring to the user's location but adds the data related to the travel mode's destination location. In our tests we noted that a user can enter the desired location on travel mode without any restriction or verification.

The `files` folder contains the `PersistedInstallation` file, whose contents (in JSON format) are shown in Listing 2. This file contains relevant information about user authentication in the application and the communication with Bumble servers. It is noteworthy that the token is renewed every seven days. In the same folder, there is a file named `c2Vzc2lvbklk`, which represents the `sessionId` in base64 encoding. However, no data with forensic value was identified.

Listing 2. Sample content of the file `PersistedInstallation`.

```
{
"Fid":"fcNePFUzQ3Oku4JbWcYI6p",
"Status":3,
"AuthToken":"eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJhcHBJZCI6IjE6OTQyMDkzMTcxODU4OmFuZHJvaWQ
↪ Q6NDYzNGE1ZjFmNjA2NjhhOSIsImV4cCI6MTYzNDE1MDQ2MCwiZmlkIjoiZmNOZVBGVXpRM09rdTRKYldjWUk2cC
↪ IsInByb2plY3ROdW1iZXIiOjk0MjA5MzE3MTg1OH0.AB2LPV8wRQIhAPLBEEYX1IiD1TXjWBjvCUeDiGjYcXnz0A
↪ acVU5kKEKtAiACsa0hQWgldbfW-Cx6lf5xwIIozeudb3Ggn8MSRuaqaA",
"RefreshToken":"2_8FZY6MgwoWuxvXhuPgNtoQKZOiwd9S1FKBTbflrdHOdyh_xKLxUXciQSVyEcM78o",
"TokenCreationEpochInSecs":1633545661,
"ExpiresInSecs":604800
}
```

Inside the `Cache` folder, two directories are of forensic interest, these being: `downloader` and `decorator_tmp`. These contain images sent, received, and viewed while using the application.

The `decorator_tmp` directory contains images with a blurring filter applied by the application, while the `downloader` folder contains the same pictures but without the filter applied. The comparison between the same image in the two directories is shown in Figures 4 and 5.

Figure 4. Image from the directory `decorator_temp`.

Figure 5. Image from the directory `downloader`.

Although the images are present in the directories mentioned above, they are quickly eliminated by the application. For this reason, these folders end up not being reliable for the collection of all images sent by a user. In addition, it is not possible to correlate the images sent in the message table, from the `ChatComDatabase` database, with the images present in the directories. Images stored in both the `decorator_tmp` and `downloader` directories do not contain relevant metadata to identify them, such as their date and time of submission. It has been determined that the Bumble application makes changes to photos uploaded by users. When sending photos with Bumble's app all EXIF (Exchangeable Image File Format) metadata is eliminated. Therefore, it is not possible to obtain typical image metadata such as the location, camera model, etc. In our tests, we were not able to confirm if the metadata is eliminated on the app itself, or on the server-side, given that the app does not allow exporting pictures taken with the app—this seems to be a privacy feature.

In the `cache` folder, only the most recent images that the user viewed, sent, or received can be observed. However, the images obtained are not satisfactory to determine whether the user sent, received, or viewed them on a user's profile. Additionally, the directory `cache/recents/teleport_cache` contains the last location, as this file only exists when the location is changed.

Additionally, requests to the Bumble API were executed. When viewing the Bumble Web API calls, it was found that several data provided, such as `session_id` and `device_id`, are present in the files directory. For this reason, this data was used to try to gain access to the URLs described in the message table, replacing Bumble Web's `cookie_session` with the tokens provided by the application. However, it was not possible to obtain results

through the requests made to the API with the authentication data provided in the files. This result may be due to numerous factors such as the lack of parameters, additional authentication by the application, requests that do not respect the structure required by Bumble, among others, making it extremely difficult to access the URLs of the images present in the payload field.

Various databases and XML files held no forensic interest. The databases `com.google.android.datatransport.events`, `lexems`, `Cookies`, and `androidx.work.workdb` were discarded since neither had any information which might be of forensic interest nor were they populated with data. These databases are related to how data transport takes place, required information for the app to function, web application session cookies, and a Room database, respectively. Correspondingly, most of the XML files had no forensic value and, therefore, were not discussed in this work.

4.3. MobSF Analysis

By means of the MobSF tool we were able to identify several relevant pieces of information from a privacy point of view. The following subsections address our findings.

The MobSF highlighted eight permissions classified as dangerous that are displayed in Table 9.

The APKID Analysis section contains information about how the APK was built, the compiler used, packages, obfuscation techniques, and more. The report shows that anti-reverse engineering techniques were applied to the application's source code. In this sense, it should be noted that anti-emulation (identified in the report as anti-VM, or virtual machine, code) and code obfuscation techniques were applied.

The Network Security section addresses the security of the application to receive and send data. A vulnerability classified as high was identified, which indicated that the application is configured to allow the traffic of cleartext (unencrypted) information for API versions lower than 27. However, it was not possible to test this vulnerability, as the available APKs were not able to compile in this Android version.

The Manifest Analysis describes some essential information about the application for the Android build tools, about permissions used in the application, and so on. In this analysis, 14 high-level vulnerabilities were identified related to essential application information and permissions that could be exploited by malicious users or to get privileged information that could help forensic analysts.

Table 9. The application's permissions classified as dangerous.

Permission	Description
`android.permission.READ_PHONE_STATE`	ff Allows read-only access to phone state, including current network information, the status of all ongoing calls, and a list of other mobile phone accounts registered on the device;
`android.permission.ACCESS_COARSE_LOCATION`	Allows access to an approximate location;
`android.permission.ACCESS_FINE_LOCATION`	Allows access to a precise location;
`android.permission.CAMERA`	Allows access to mobile phone camera;
`android.permission.RECORD_AUDIO`	Allows the application to record audio;
`android.permission.WRITE_EXTERNAL_STORAGE`	Allows the application to write to external storage;
`android.permission.READ_EXTERNAL_STORAGE`	Allows the application to read from external storage;
`android.permission.GET_ACCOUNTS`	Allows access to the list of accounts in the Accounts Service, which contains the list of accounts that are associated with the user.

The Code Analysis section contains information regarding the vulnerabilities identified in the application and their respective classification according to CVSSv2 (Common Vulnerability Scoring System version 2) [26]. However, the vulnerabilities were not analyzed due to obfuscation techniques applied to the source code.

The NIAP Analysis data describes, oversees, and monitors the security of commercial Information Technology products. In this section, only one risk was identified, since the application uses cryptographic hashing services in disagreement with the FCS_COP.1.1(2) and due to the use of outdated and insecure cryptographic algorithms like RC2/RC4 and MD4/MD5.

The Domain Malware Check refers to malware that may be present in the application. In this section, it was identified that the domains used by the application are not listed as unsafe and are not related to suspicious or malicious records or activities. In this way, all domains related to the application are identified as safe.

Multiple URLs hardcoded in the application have been identified. Despite this, none were identified or characterized as suspicious. Also, six emails were identified by the tool, however, only two are valid: `android.support@bumble.com` and `android.issue@corp.badoo`.

Trackers are pieces of software that are used to record information about the user. The MobSF report highlights the following trackers: `AppsFlyer`, `FacebookAnalytics`, and `GoogleFirebaseAnalytics`. Reviewing the data found by MobSF, it concludes that the companies mentioned are utilized for statistical purposes. Because of this, the Bumble application may send data to obtain relevant information about its users, which could not be identified in the analysis performed.

The Hardcoded secrets section contains data relating to secrets, privileged information such as API keys, passwords, and other relevant data stored in the application. In this section, possible secrets were identified, however, only two were recognized as being valuable, the `google_api_key`, and the `google_crash_reporting_api_key`.

The information found in the static analysis is common to other applications and was the only data of interest besides the API keys. The access to the API key will be able to make undue calls to the application and allow access to sensitive information to unauthorized people. Due to this problem, customer trust can be diminished, in addition to causing financial losses for the organization. However, this might be more harmful to the app itself than its users.

4.4. Dynamic Analysis

Bumble requires the location service to be active to load its contents. Hence, we expected to find some GPS coordinates in the *post-mortem* analysis, but that was not the case. We hypothesized that such information would be transient and not stored in the mobile device. To test this hypothesis we performed a dynamic analysis as described in the following paragraphs.

Android implements several protective mechanisms to prevent impersonation attacks (also known as man-in-the-middle attacks). One of those mechanisms is HTTP Public Key Pinning (HPKP), also known as certificate pinning. The HPKP security mechanism is delivered via an HTTP header, which allows HTTPS websites to resist impersonation using fraudulent digital certificates. A server uses it to deliver to the client (e.g., web browser, or an app) a set of hashes of the public keys that must appear in the certificate chain of future connections to the same domain name [27]. Although this mechanism is a good protective measure for end users' privacy, it is also a challenge to overcome to do a forensic analysis. We followed the "Android Network Traffic Interception" tutorial available on Github [28], but with some modifications. This tutorial shows how to install a proxy and a new digital certificate into the Android device to intercept the traffic. However, we opted to use Packet Capture because it allows us to intercept the network packets inside the Android device itself. The Packet Capture also requires the installation of a Certificate Authority (CA) digital certificate bundle (both private and public keys) that we created by issuing the `openssl` commands present in Listing 3.

Listing 3. `openssl` commands to generate a CA digital certificate to be used by Packet Capture to intercept traffic.

```
openssl req -x509 -newkey rsa:4096 -keyout key-private.pem -out key-public.pem -days 3650
openssl pkcs12 -export -out keyBundle.p12 -inkey key-private.pem -in key-public.pem -name
↪  "alias"
```

Once the certificate bundle is installed in Android, the traffic is intercepted when redirected by the operating system through a fake VPN service created by the Packet Capture. However, this is not enough, as the Bumble app still refuses to load any content due to the certificate pinning security mechanism. Therefore, we also had to resort to the Frida tool, which is able to intercept Bumble's calls to the Android API to overcome this security mechanism.

The data we intercepted includes user profile information, mobile device information (manufacturer, model, version, and device_id), and event timestamps in Unix Epoch format, as shown in Listing 4. We also found many packets corresponding to the images on other Bumble's users' profiles as presented in Figure 6. Although Bumble requires the location service to be active, we did not find any GPS coordinates being transmitted. However, we cannot assert that this information is never sent to Bumble's servers. First, the location information might be sent only sporadically and our tests might not have been long enough to find it. Second, the information might be encoded in an unknown way and, for that reason, it was not identified.

Listing 4. Sample of `json` encoded data sent to Bumble's servers.

```
{
"application": {
"brand": 4,
"layout": 1,
"platform": 3,
"app_version": "5.250.1",
"is_premium": false
},
"device": {
"manufacturer": "samsung",
"model": "SM-A405FN",
"os_version": "10",
"locale": "en_GB",
"device_id": "bab7fdb7ae71e6be"
},
"user": {
"user_id": 0,
"gender": 1,
"country_id": 48,
"age": 31,
"encrypted_user_id": "zAhMACTkzMjQyMjIxNQiHYnQ6AA..."
},
"session_id": "0d08e0a6-5a1b-19ed-8d64-a6dc6acfc753",
"events": [
{
"name": 703,
"body": {
"android_jinba": {
"value": 33,
"measurement": 46
}
},
"ts": 1643827853338,
"tracking": {
"screen_id": "29",
"screen_name": 11
}
}, ... ]
...
}
```

Figure 6. Example of traffic captured with the Packet Capture software showing an image transfer.

5. Bumble Parsing Script

To help digital practitioners to extract and analyze Bumble's forensic artifacts, we developed the script `Bumble-Extraction`. This script allows the visualization of users' conversations and exchanged messages, as displayed in Figure 7. The script is developed in Python, and to execute it, the forensic practitioner needs to insert the path to the Bumble's private directory, usually identified by `com.bumble.app`. Besides displaying the user messages, it also permits filtering by conversation and saving the results in Portable Document Format (PDF) format. The script is open-source and published on GitHub [29].

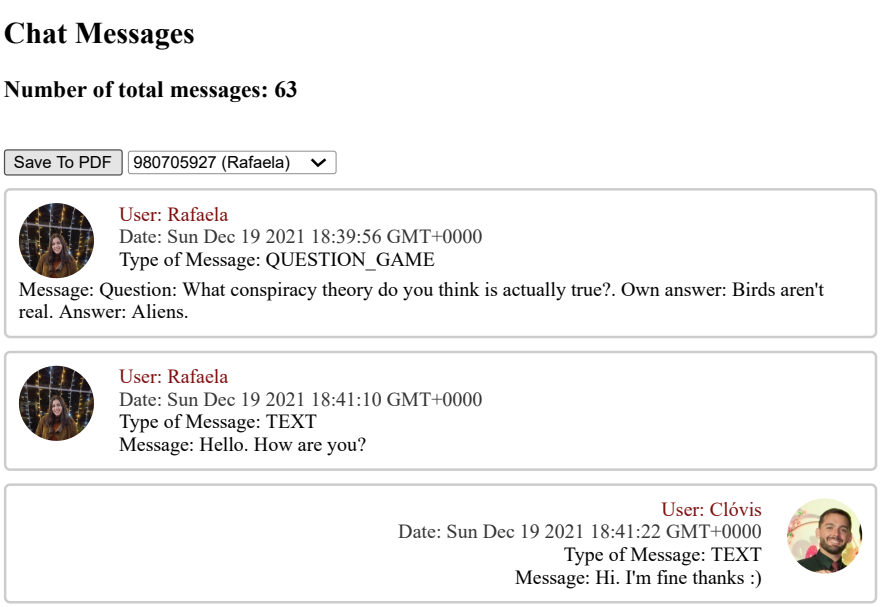

Figure 7. Sample of messages exchanged by the user.

6. Conclusions

Bumble is an e-dating application with many features and a complex internal structure. Thus, numerous tests were executed, using both virtual and physical devices, to cover as many use cases as possible. Afterward, an in-depth post-mortem analysis was done to determine what user data with forensic value is stored. Like other mobile apps, Bumble stores its data in public and private directories, hence `/sdcard/data/com.bumble.com` and `/data/data/com.bumble.com` directories were the primary focus of our analysis. While investigating the public directory, it was concluded that there was no relevant or forensically relevant data. However, the private directory was the focal point of analysis since it revealed the most significant results, highlighting several forensics artifacts. Bumble's private directory, in version 5.250.1, contains 6 SQLite3 databases, accounting for 40 tables, numerous cache files, and 36 XML files. These provided valuable information about a user, such as the messages exchanged with others, its matches, data about the linked Facebook account, and configuration settings. Yet, the files sent (images and audio recordings) were not obtainable since these are not stored locally. Nevertheless, it was possible to identify some of the pictures exchanged in personal conversations within cached files. Given its size, a forensic analyst with exclusive access to the private directory cannot accurately tell which were sent and received by the user. Additionally, the automated static analysis identified Bumble's required permissions, APKID, source code, secrets, application vulnerabilities, URLs, and IPs. To analyze the application's behavior in real-time, a dynamic analysis was performed with the help of tools to bypass the certificate pinning security mechanism, and intercept the network traffic on an Android device. The intercepted data included the user profile information, and the mobile device information (manufacturer, model, version, and device_id). Although Bumble requires the location service to be active, no GPS coordinates were found being transmitted. However, it is not possible to assert that this

information is never sent to Bumble's servers. Finally, a Python script was developed that enabled the visualization of messages exchanged between users through a web browser and its export to a PDF document.

As future work, it would be valuable to obtain the pictures and audio exchanged through the links available on `ChatComDatabase`'s message table. Eventually, it would be significant to investigate the application's vulnerabilities and determine if they can be useful to obtain more forensic relevant data. Additionally, it would be valuable to further develop our script to be included in tools like the Autopsy Forensic Browser, and the Android Logs Events And Protobuf Parser (ALEAPP) (https://github.com/abrignoni/ALEAPP, accessed on 28 December 2021).

Author Contributions: All authors conceived the presented idea. The development of the conceptualization, methodology, analysis, and validation was produced by every author in equal share. A.B., R.A. and T.M. made the required tests, data extractions, and software development. M.F. encouraged the investigation and supervised the findings of this work. All authors discussed the results and contributed to the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by CIIC under the FCT/MCTES project UIDB-04524-2020.

Institutional Review Board Statement: This study did not require ethical approval.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers for their insightful comments, which helped to improve this work.

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

References

1. Sathe, S.C.; Dongre, N.M. Data acquisition techniques in mobile forensics. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 280–286.
2. Alter, C. Whitney Wolfe Wants to Beat Tinder at Its Own Game. *Time*, 15 May 2015. Available online: https://time.com/3851583/bumble-whitney-wolfe/ (accessed on 28 December 2021).
3. Bumble-Date, Meet, Network Better. 2021. Available online: https://bumble.com (accessed on 14 December 2021).
4. Yashari, L. Bumble C.E.O. Tries to Change Dating After Dramatic Tinder Exit. *Vanity Fair*, 7 August 2015. Available online: https://www.vanityfair.com/culture/2015/08/bumble-app-whitney-wolfe (accessed on 28 December 2021)
5. Bumble Revenue and Usage Statistics (2021). 2021. Available online: https://www.businessofapps.com/data/bumble-statistics (accessed on 16 December 2021).
6. Branson, J. Bumble Statistics and Facts in 2021 [with Charts]. 2021. Available online: https://boostmatches.com/bumble-statistics (accessed on 16 December 2021).
7. MacLeod, C.; McArthur, V. The construction of gender in dating apps: An interface analysis of Tinder and Bumble. *Fem. Media Stud.* **2019**, *19*, 822–840. [CrossRef]
8. Bumble-Bumble's Latest Safety Feature Uses A.I. to Blur Unwanted Nude Photos. 2021. Available online: https://bumble.com/en-us/the-buzz/privatedetector (accessed on 21 December 2021).
9. Bumble-Privacy. 2021. Available online: https://bumble.com/en/privacy (accessed on 16 December 2021).
10. FBI, I.C.C.C. Internet Crime Report 2020. Available online: https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf (accessed on 22 December 2021).
11. Hayes, D.R.; Snow, C. Privacy and Security Issues Associated with Mobile Dating Applications. In Proceedings of the Conference on Information Systems Applied Research ISSN 2473-3857, Norfolk, Virginia, 31 October–3 November 2018; Volume 2167, p. 1508.
12. Kim, K.; Kim, T.; Lee, S.; Kim, S.; Kim, H. When harry met tinder: Security analysis of dating apps on android. In Proceedings of the Nordic Conference on Secure IT Systems, Oslo, Norway, 28–30 November 2018; pp. 454–467.
13. Shetty, R.; Grispos, G.; Choo, K.K.R. Are you dating danger? an interdisciplinary approach to evaluating the (in) security of android dating apps. *IEEE Trans. Sustain. Comput.* **2017**, *6*, 197–207. [CrossRef]
14. Farnden, J.; Martini, B.; Choo, K.K.R. Privacy risks in mobile dating apps. *arXiv* **2015**, arXiv:1505.02906.
15. Leith, D.J. Mobile Handset Privacy: Measuring The Data iOS and Android Send to Apple And Google. In Proceedings of the International Conference on Security and Privacy in Communication Systems, online, 6–9 September 2021.

16. Bays, J.; Karabiyik, U. Forensic Analysis of Third Party Location Applications in Android and iOS. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 1–6.
17. Kim, H.; Kim, D.; Jo, W.; Shon, T. Digital Forensic Analysis using Android Application Cache Data. In Proceedings of the 2019 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 28–30 January 2019; pp. 1–4. [CrossRef]
18. 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, 29 December–1 January 2020; pp. 836–847. [CrossRef]
19. Mobile App Analysis Part 5—The Leahy Center for Digital Forensics & Cybersecurity. 2017. Available online: https://leahycenterblog.champlain.edu/2017/04/14/mobile-app-analysis-part-5/ (accessed on 16 December 2021).
20. Knox, S.; Moghadam, S.; Patrick, K.; Phan, A.; Choo, K.K.R. What's really 'Happning'? A forensic analysis of Android and iOS Happn dating apps. *Comput. Secur.* **2020**, *94*, 101833. [CrossRef] [PubMed]
21. Phan, A.; Seigfried-Spellar, K.; Choo, K.K.R. Threaten me softly: A review of potential dating app risks. *Comput. Hum. Behav. Rep.* **2021**, *3*, 100055. [CrossRef]
22. Fatehi, S. SchemaCrawler. 2021. Available online: https://www.schemacrawler.com/ (accessed on 16 December 2021).
23. Shang, S.; Wu, Q.; Wang, T.; Shao, Z. LiteIndex: Memory-Efficient Schema-Agnostic Indexing for JSON documents in SQLite. In Proceedings of the 26th Asia and South Pacific Design Automation Conference, Tokyo, Japan, 18–21 January 2021; pp. 435–440.
24. Domingues, P.; Nogueira, R.; Francisco, J.C.; Frade, M. Post-mortem digital forensic artifacts of TikTok Android App. In Proceedings of the 15th International Conference on Availability, Reliability and Security, Virtual Event, Ireland, 25–28 August 2020; pp. 1–8.
25. Firebase's Official Documentation | com.google.android.gms.measurement. Available online: https://firebase.google.com/docs/reference/android/com/google/android/gms/measurement/package-summary (accessed on 1 February 2022).
26. National Vulnerability Database—Vulnerability Metrics. 2021. Available online: https://nvd.nist.gov/vuln-metrics/cvss (accessed on 16 December 2021).
27. Contributors to Wikimedia Projects. HTTP Public Key Pinning—Wikipedia. 2021. Available online: https://en.wikipedia.org/w/index.php?title=HTTP_Public_Key_Pinning&oldid=1056993476 (accessed on 2 February 2021).
28. Frade, M. Android Network Traffic Interception. Available online: https://github.com/labcif/Tutorial-AndroidNetworkInterception (accessed on 2 February 2021).
29. Almeida, R.; Barros, A.; Melo, T. Bumble-Extraction: Bumble Parsing Script 2021. Available online: https://doi.org/10.5281/zenodo.5805024 (accessed on 28 December 2021).