2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control

The Forensic Analysis of WeChat Message

Lijun Zhang
Science and Technology on Communication Security Laboratory,
Chengdu, 610041, China
e-mail: achong731@sohu.com

Fei Yu, Qingbing Ji
Science and Technology on Communication Security Laboratory,
Chengdu, 610041, China
e-mail: jqbdxy@163.com

***Abstract*—WeChat is a popular instant messaging application on Android, iPhone and BlackBerry smart phones, whose chat messages are all stored in local installation folder. This paper studied the data forensic techniques of WeChat messages including the identification of storage location, storage structure and information extraction methods. Since the text message is stored in encrypted SQLite database, we detailedly analyze its cryptographic algorithm, key derivation principle and present the corresponding database decryption process in different practical forensic circumstances. In addition, we exploit the data recovery of voice and deleted messages which would also be helpful in data forensic for criminal investigation.**

Keywords- WeChat, database decryption, key derivation, data forensics

I. INTRODUCTION

WeChat [1] is a free instant messaging application which provides text and voice communication service for smart terminal users. It is available on various phone operating systems such as Android, iPhone and BlackBerry. Besides the basic functionality of sending text, voice, image and video message, WeChat has other multiple convenient services including payment, public account and location sharing. Especially, people can make use of "Moments" to post personal photos, interesting music as well as articles. Only the friends from the user's contact will be able to view their Moments content and comments. Due to the free and abundant services, WeChat has become one of the largest standalone messaging applications. As of the fourth quarter of 2015 [2], it has covered more than 90% smart phones and the monthly active users reached 650 million in China.

On the other hand, with the popularization of WeChat, more and more criminals employ this newfashioned tool to communicate with each other. So it is very helpful to study data forensics and find out criminal clue or evidence from the message records of WeChat [3]. In the WeChat forensics, the most important data are various messages including text, voice, image and video information. These kinds of data are encrypted and stored in SQLite database of WeChat's local directory. Zhao [4] studied the data recovery method of deleted records in SQLite database on Android system while Ma [5] presented how to recover the WeChat's deleted messages on IOS system. Qu et al [6, 7] exploited the interaction protocol framework and encryption mode of WeChat, but could not be able to acquire the message content. As can be seen from the current forensic research, they mainly focus on the recovery of deleted data. However, these methods cannot be applied in the practical forensic environments since they are all based on the plaintext of database whereas most content of the WeChat's message in database is encrypted. Therefore, we should first study the decryption of database which is the premise of further obtaining the existing and deleted chat records.

This paper will investigate the data forensics of WeChat messages in local encrypted database. At first we will identify the data storage location and the format of ciphertext. Furthermore, we will detailedly analyze the data encryption algorithm as well as decryption key generation process then present the decryption method of encrypted database under different forensic circumstances. Finally, for the case of deleted message, we also discuss the data recovery way. After an actual test, our forensic techniques can successfully recover the encrypted and deleted messages which provide a complete solution for WeChat data forensics.

II. THE STORAGE PRINCIPLE OF WECHAT'S MESSAGE

All kinds of WeChat's message are stored in the specified installation folders. In this paper, we take Android operating system as an example to explain message storage location. Concretely, WeChat's messages are saved in the SQLite type database named "EnMicroMsg.db". Note that this database is unencrypted before version 4.5 but all later versions use encrypted storage. The database is located in the system directory "data/data/com.tencent.mm/MicroMsg/MD5(ID)", where MD5(ID) is the MD5 hash (with total 32 characters) of WeChat account ID logged in the smartphone. Therefore, if several account IDs have logged in before, then there would be more than one such subfolders under "MicroMsg" directory and each subfolder contains an encrypted database. All text messages are stored directly in the database EnMicroMsg.db, but voice, image and video are only recorded by their path information of storage location while the original files are saved in corresponding folders. For example, voice messages are stored in the subfolder called "voice2" with a special file extension "amr". Images are stored in the subfolder "image2" and videos are in the subfolder "video". The storage pathname of all messages is described in Fig. 1.

The file "EnMicroMsg.db" is encrypted so the ordinary database browser software can not directly view the content of tables in this database. It is necessary to analyze the encryption mechanism and decryption method. After this

978-1-5090-1195-7/16 $31.00 © 2016 IEEE
DOI 10.1109/IMCCC.2016.24

database is recovered in plaintext, one could parse the database's tables to acquire the text message of chat record accordingly. Furthermore, the image, voice and video messages can be extracted and decoded from corresponding folders.

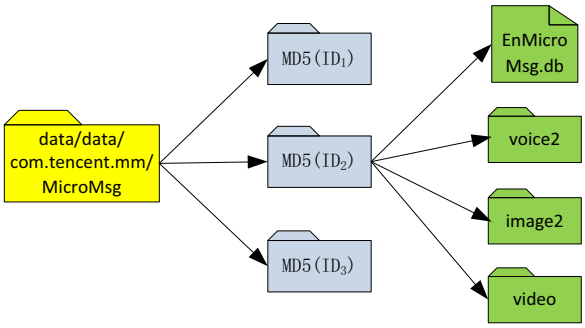

Figure 1: the storage path of WeChat messages

III. THE DATA FORENSICS OF WECHAT'S MESSAGE

After analysis and practical test, we ensure that WeChat's message database file "EnMicroMsg.db" is SQLite type [8] and encrypted by the open source encryption tool called sqlcipher [9]. Now, we will we analyze the physical structure and encryption mechanism of SQLite database which is indispensable for the plaintext recovery of chat content.

A. The Physical Structure of SQLite Storage

Physically, the file of SQLite database consists of pages with fixed size which is between 512 bytes and 32768 bytes (but must be exponent of 2). For the case of WeChat, it adopts the default size of 1024 bytes , i.e. 1KB. These pages are numbered sequentially from page 1, then page 2 and so forth. Logically, a SQLite database is composed of multiple Btrees (a kind of data structure). A Btree page could store a table's index or data and generally table's data is placed in a data structure called "B+tree" while table's index is represented in "B-tree" structure. The whole database's structure information (called schema) is placed in the SQLite system table "sqlite master" which stores the root page number of all data tables or indexes.

For the decryption of WeChat's encrypted database, the most important page is page 1, because it contains parameter information of decryption key. Now, we first present the unencrypted file header in page 1 which describes the permanent parameters when the SQLite database is created. These parameters include SQLite version, file format version, page size and other related information. Note that SQLite data uses unified big-endian storage mode, i.e., the biggest byte is stored in the lowest address. The first 100 bytes of file header are described in Table Ⅰ.

SQLite file header information is essential for decrypting the encrypted database. We will analyze the encryption mechanism of WeChat database and present the decryption method in the next section.

TABLE I. UNENCRYPTED FILE HEADER OF SQLITE DATABASE

Offset	Size	Description
0	16	fixed string "SQLite format 3"
16	2	page size, default is 1024 bytes
18	1	file format version (for written), default is 0x01
19	1	file format version (for read), default is 0x01
20	1	reserved space at the end of page, default is 0x00
21	1	maximal space used in Btree internal page
22	1	minimal space used in Btree internal page
23	1	minimal space used in Btree leaf page
24	4	the modified times of database file
28	4	not used
32	4	pointer of free page list, 0x00 means empty list
36	4	number of free page list
40	60	15 metadata with each 4 bytes

A concrete instance of file header with first 48 bytes is shown in Fig. 2.

```
53 51 4C 69 74 65 20 66 6F 72 6D 61 74 20 33 00 ; SQLite format 3.
04 00 01 01 00 40 20 20 00 00 00 D4 00 00 01 99 ; .....@  ...?..?
00 00 00 00 00 00 00 00 00 00 00 C1 00 00 00 04 ; ............?...
```

Figure 2: an instance of SQLite database file header

B. The Encryption Mechanism of WeChat Database

From version of 4.5, WeChat database makes use of an open source encryption tool called "sqlcipher" to encrypt the entire database file so that the resulting database becomes completely unreadable. An encrypted WeChat database is shown in Fig. 3 from which we can see the file header is also encrypted.

```
56 A2 0E 97 72 A9 47 06 C4 C3 1C DF 9E 55 82 09 ; V?枌三.箾.邯U?
85 E3 7A 44 5A 59 06 D1 B8 52 D5 CB 00 CA F5 2B ; 呫zDZY.迅R账.术+
F4 C5 08 FC EB 38 D9 B6 3F 6D 17 68 7F 20 9B EA ; 騷.  8俣?m.h  泞
08 B5 F0 CB 32 3E 53 85 52 18 BB 99 78 2C 39 8F ; .叼?>S匯.桿x,9?
```

Figure 3: an encrypted WeChat database file header

By analyzing the encryption functions in sqlcipher source code, we obtain the encryption mechanism of WeChat database.

Encryption Algorithm. Sqlcipher adopts the specific function "sqlcipher_page_cipher" to encrypt database file in which the encryption algorithm is AES. In the realization of this algorithm, it chooses the operation mode of cipher block chaining (CBC mode) and key of 256 bits length where the initialization value (IV) in this mode can be extracted from the database's ciphertext.

Process of Key Derivation. After a detailed analysis, we find out that the encryption key is derived from a password with length of 7 bytes by using key derivation function "PBKDF2" [12]. While this password can be calculated by MD5 hash function from the parameters of phone's IMEI (international mobile equipment identity number) and UIN (user information number) belonging to the encrypted database where IMEI is generally a string of numbers with length 14 or 15 and UIN is a string of numbers with length 9 or 10. Note that UIN of every database file uniquely corresponds to a WeChat's user ID number. More concretely,

let Hash=MD5(IMEI||UIN), then password is truncated the first 7 characters of "Hash", i.e., password=Hash[0-6]. Here we give a practical example.

Let IMEI=99000524965332, UIN=332902842, first we need to compute MD5 hash value of their concatenation MD5(IMEI||UIN) to acquire the hash value

Hash=8f07b3a5f613c346894890bca0716f7b,

then password is 8f07b3a.

After obtaining the password, we can calculate the 256-bit AES decryption key "dec_key" by using derivation function PBKDF2. This function is implemented in sqlcipher "kdf " function, namely,

dec_key=kdf(password, pass_sz, salt, salt_sz, iter, key_sz), where "pass_sz" is the length of password (here is the fixed value 7) and "salt" is the 16 bytes random value read from the database's ciphertext. "salt_sz" is the length of salt and its value here is 16. "iter" is the iteration times of PBKDF2 and its default value is 4000. "key_sz" here is valued by 32 bytes (i.e., 256 bits).

Remark. In the sqlcipher's newest version 3.8.10, the default value of iteration has been changed from 4000 to 64000 which can improve the security strength. However, WeChat has not change this parameter value until now.

C. Obtain the Parameters of Key Derivation

Now, we will present how to obtain all the parameters needed in the decryption key derivation under different forensic circumstances.

Case 1. The phone has been root. In this case, we can acquire all the file access privilege in the WeChat's installation folder and all the parameters needed in the key derivation can be obtained from relevant files. The parameters involved in the calculation of decryption key are IMEI, UIN and salt. Here we give explicit method to obtain these parameters. IMEI is the unique identification number of global mobile phone which can be obtained directly from the label on the back of the phone or from the phone's screen after you enter "*#06#" on the phone dial pad. In addition, WeChat software itself also manages IMEI number which is embedded in the configuration file "CompatibleInfo.cfg". In the same way, UIN can be extracted from the configuration file "systemInfo.cfg". While in the decryption key derivation, we also need the random value "salt" with length of 16 bytes which is exactly the first 16 bytes in the encrypted database "EnMicroMsg.db". The complete computation process of decryption key is described in Fig. 4.

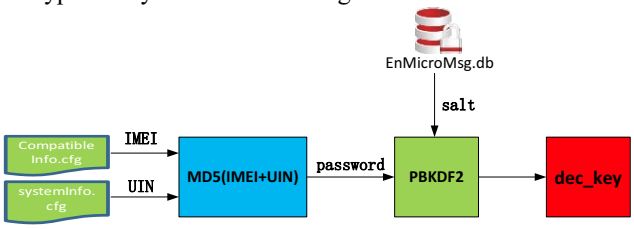

Figure 4: the process of decryption key derivation

Case 2. The phone is not root or we only have the encrypted database file "EnMicroMsg.db". In this case, we cannot get the UIN number and even the IMEI number when only the encrypted database file is provided. From the process of decryption key derivation, we know that the function PBKDF2 needs both password and salt. However, the password could not be computed now because of lacking UIN number which results in the failure to acquire decryption key directly. We will study in this case how to get the correct decryption key.

We analyzed the encryption principle of sqlcipher for WeChat database. The file "EnMicroMsg.db" is encrypted entirely as the file header is also ciphertext as we have seen before. When encryption is implemented, the page of database is divided into every block of 1024 bytes which is encrypted individually. The front 16 bytes of the first page are replaced by a random salt value while the initial value IV in CBC operation mode of AES algorithm is stored in the last 16 bytes at every page, i.e., the start position of IV value is 1008 bytes offset from page header. Since the first page contains the file header of the whole database and several bytes at some position are fixed, we could attempt every possible password and then derive its corresponding decryption key. Then decrypt that ciphertext block containing fixed bytes and compare them with its plaintext counterpart. If they match, we know this decryption key is correct. More explicitly, we find that the bytes from offset position 17 to 24 are fixed as shown in table 1 which are always " 0x40, 0x00, 0x01, 0x01, 0x00, 0x40, 0x 20, 0x20". Note that although the first 16 bytes are fixed in unencrypted database (i.e., these bytes are "SQLite format 3"), however theses bytes are not encrypted but substituted by the salt value directly. Hence we cannot make use of them to decryption and comparison. The password attempt method is described in Fig. 5.

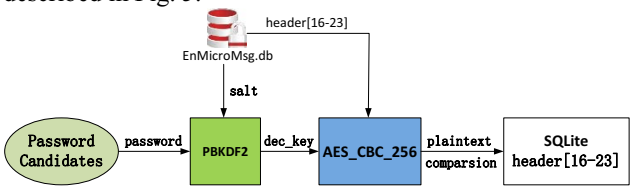

Figure 5: The correct password crack attempting

D. Decrypt the WeChat's Database

Once the correct decryption key is computed, we could use sqlcipher to decrypt the entire encrypted database file "EnMicroMsg.db". While implementing decryption, we should pay attention to setting the appropriate parameters including decryption key, PBKDF2 iterations, database page size and so on.

Parameters are configured in the specific sqlcipher command as follows :

sqlite> PRAGMA key = '8f07b3a' ; //set decryption key

sqlite> PRAGMA cipher_use_hmac=off; //close hmac authentication

sqlite> PRAGMA kdf_iter=4000; //set iterations of PBKDF2

sqlite> PRAGMA cipher_page_size=1024; //set default page size of database

Once the database is decrypted, we can parse the data table inside the database to acquire the chat content of WeChat users where the friends list is stored in the table "contact" which contains the field name and nickname. Chat message information is stored in the table "message" and "qmessage", where qmessage table stores the offline chat content. From the data table of chat message, it can be seen text and emotional picture are directly stored in the table but image, voice, and video messages are recorded by their storage pathname while the actual files can be found in the corresponding file folders.

IV. THE FORENSICS OF VOICE MESSAGE

The voice message of WeChat in Android phone is encoded with "amr" file extension. However, in fact it is not the actual "amr" file format but indeed another file format called "silk". We can use the hex editor to open the voice file and will see bytes "#!SILK" in the file header. Silk encoding is a wideband audio coding format developed internally by SKYPE company [14] which has been released with open source. This encoding is specially provided for the third party developers and hardware manufacturer freely, but it could not be decoded by the currently ordinary voice player and need transcoding. In fact, it can be converted to "wav" format by "silk" source code then played on a voice player.

V. THE FORENSICS OF DELETED MESSAGE

If the WeChat messages are deleted, it is unable to view them in the data table. However, according to analysis of SQLite database storage mechanism, we know that the actual message maybe still exists in the database but only its page header information is erased. That is, database will delete the first page header of deleted data and marked it as a free page but its data area is not deleted. So we can recover the deleted message as long as the deleted data area has not been covered by other data.

Specifically, it is possible to locate and extract deleted data according to the logical structure of the database file page. In fact, all the data in SQLite is stored in the page and each page has its corresponding file structure. There are three types of page: Btree page, free page and overflow page. Every data table consists of multiple Btree pages, while the logical organization of Btree page is a tree structure. The root page and internal page in Btree are mainly used for the navigation whose pointer fields point to the lower level pages. All data chat records are stored in the leaf pages including the deleted messages. Therefore, in order to find the deleted data, it is necessary to search the root page and use the navigation pointer to determine the leaf page. Finally, the deleted message can be extracted from the corresponding data area. The data searching process is as shown in Fig. 6.

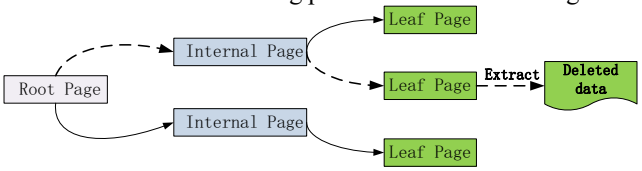

Figure 6: searching process of deleted data in SQLite

VI. CONCLUSION

This paper studied the data forensic principle of WeChat messages and especially presented the detailed decryption analysis of encrypted chat records database. We explicitly explained the usage of cryptographic algorithm in encryption and gave the computation method of correct decryption key under different forensic circumstances. Moreover, we also demonstrated how to recover the encoded voice and deleted messages which is very useful for the real data forensic acquirement.

ACKNOWLEDGMENT

This work is supported by Foundation of National Natural Science (No.61309034). The authors also acknowledge the reviewers for their useful opinions to improve this paper.

REFERENCES

[1] Tencent company. Introduction to Tencent's products: function and feature of WeChat, available at http: //weixin.qq.com

[2] Wikipedia. The user number statistics of Wechat 2016, available at https://en.wikipedia.org/wiki/WeChat

[3] McMillan R, Glisson B, Bromby M. Inverstingating the increase in mobile phone evidence in criminal activities. The 46-th Hawaii International Conference System Science 2013, pp.4900-4909, 2013.

[4] Zhao K. The key technology research on Android forensics, Master's thesis, Guangxi University for Nationalities, 2015.

[5] Ma M. The principle research and technological experiments of WeChat's deleted data recovery, Master's thesis, Yunnan University, 2015.

[6] Qu X, Xue Z. The analysis and research of microletter encryption, Microcomputer Applications, 2014, 5 (1): 13-16.

[7] Wan Y, Gu Y, Qiu W. Research on Interactive Protocol and Encryption Mode of Wechat, Microcomputer Applications, 2015 (31), No.2, 31-34.

[8] Michael O. The Definitive Guide to SQLite, Berkeley of the USA, Apress, 2006.

[9] China open source community. Sqlcipher encryption source code, https://github.com/sqlcipher/sqlcipher, 2016

[10] SQLite official website. The format of SQlite file header, http://www sqlite.org/fileformat Html, 2016

[11] Kelly S, Frankel S, Glenn R. The AES-CBC Cipher Algorithm and Its Use with IPsec, 2003.

[12] Kaliski B. Password Based Cryptography Specification Version2.0. http://toolsiet.org/html/rfc2898, 2000-09.

[13] Zhang Y, Fang Z, Wang K. The overview of Android security vulnerability mining technology, Computer Research and Development, 2015, 52 (10): 2167-2177.

[14] Skype company, silk coding and storage format, 2015, available at http://developer.skype.com/silk