Digital forensic analysis of encrypted database files in instant messaging applications on Windows operating systems

Jusop Choi\textsuperscript{a}, Jaegwan Yu\textsuperscript{b}, Sangwon Hyun\textsuperscript{c}, Hyoungshick Kim\textsuperscript{a}

\textsuperscript{a} Sungkyunkwan University, Korea
\textsuperscript{b} LIGNEX1 Co., Ltd, Korea
\textsuperscript{c} Chosun University, Korea
Agenda

1. Introduction
2. Our analysis framework
3. Analysis of example IMs
4. Conclusion
Agenda

1. Introduction
2. Our analysis framework
3. Analysis of example IMs
4. Conclusion
Instant messengers (IM) everyday

- We are using instant messengers every day.
- There are a lot of users (WeChat: 1.1 billion, Telegram: 200 million, and Facebook Messenger: 1.3 billion users by 2019).
Who can see our messages?

- Government agencies can obtain messages to use as criminal evidence.
- Government agencies can censor messages.
- Hackers can steal messages from us.
How can messages be obtained?

1. Network packet analysis  
2. Server database analysis  
3. Client database analysis
How can messages be obtained?

1. Network packet analysis
2. Server database analysis
3. Client database analysis

- Network traffic is securely protected by well designed security protocols.
- Overall, it is **not easy to analyze** the securely protected network traffic.
How can messages be obtained?

1. Network packet analysis
2. Server database analysis
3. Client database analysis

- It is hard for the attacker to access the server.
- Because of the privacy issue, many messengers do not store the messages in the server over a certain time period (e.g., 3 days for KakaoTalk).
How can messages be obtained?

1. Network packet analysis
2. Server database analysis
3. Client database analysis

3. Client database analysis

- All messages are stored in database files in the client device.
- In our research, we focused on analyzing the client chat database.
Agenda

1. Introduction
2. Our analysis framework
3. Analysis of example IMs
4. Conclusion
What databases are used in IM

• There are various database types in instant messengers.
  – Contacts
  – Chats
  – Calls
  – Photos
  – Video
Interaction with database in IM

- To minimize network overhead, chat databases are stored in local files in the client device.
Chat database file on client side

- If chat database files are not encrypted (e.g., in WhatsApp), it is easier to analyze!
- However, in some messengers, the database files are encrypted.
- The number of IMs using the encryption feature is increasing over time.
Use of encryption key

- After login, encryption key can often be (temporarily) created (in memory).
- After logout, the encryption key can be deleted from memory.
- In such cases, it is hard to obtain the encryption key.
Adversary model

- The attacker can obtain the *encrypted* database file.
- The attacker’s goal is to obtain the database *encryption key*.
Key observations in the encryption key generation

- To securely generate the dynamic encryption key, most IMs used the following parameters for key generation.
  1) Device specific local parameters: UUID, HDD serial number, …
  2) Externally obtained parameters after login: user password, user key stored at the server, …
Overall procedure

User
- User Message
  - “Hello?”

Storage
- Encrypted Database
- Encrypted Database
- Encrypted Database

Messenger
- Login
- Decrypt
- Plaintext database file
- Logout
- Encrypt

= Local parameters + External parameters
How to obtain the encryption key

1. Find the (encrypted) chat database files in the target IM.
2. Check whether the chat database files are encrypted.
3. Find the code locations to encrypt/decrypt the chat database files during the execution of the IM.
4. Identify the cryptographic algorithms (e.g., AES, SHA256) used in encrypting chat database files by examining the assembly codes and the dynamic behaviors of the IM.
5. Analyze the procedure of generating an encryption key.
Agenda

1. Introduction
2. Our analysis framework
3. Analysis of example IMs
4. Conclusion
Target messengers

- KakaoTalk is the most popular messenger in South Korea with 49.8 million users.

- NateOn is also popular in South Korea with 15 million users.

- QQ is the most popular messenger in China with 899 million users.
Example 1. KakaoTalk

KakaoTalk

NateOn

QQ
Experiment environment

• Environment and tools
  – VMware workstation 12
  – Windows 7
  – OllyDbg v1.10
  – KakaoTalk (version 2.0.8.990 ~ 2.5.6.1545)
    → Current version (3.0.0.2110)
  – Packed by Themida
Database file path

- The databases are stored in a pre-configured file system path, and the user cannot change it.
  - chatLogs_[random value].db: chat database file
  - TalkUserDB.db: contact database file
Database encryption process

- With the assembly codes and S-Box for AES-128, we can guess that AES-128-CBC algorithm was used.
- We also figured out the chat database file format (SQLite format 3).
Extraction of encryption key and IV

Through the dynamic analysis, we found which values are used for the encryption key and IV.

With those values, we tried to trace back the key generation procedure.

Key: 3a4ddec7a813e86796a42b282e24457f
IV: 342e1eea12095623400123408bc914e2
How to generate the key

1) $S_u = K_{PRAGMA} || N_u = \text{LSB}g+UolTpP ... +np6TN/yQ==12345\text{LSB}g+UolTp...==123$

2) $K_u = \text{MD5}(S_u) = 3a4ddec7a813e86796a42b282e24457f$

3) $IV = \text{MD5(Base64}(K_u)) = \text{MD5(Ok3ex6g...RFfw==)) = 342e1eea...08bc914e2}$

- $K_u$ is used as the encryption key.
- The input of the hash function ($S_u$) is the PRAGMA key combined with the user sequence number ($N_u$) given by the server.
How to generate the PRAGMA key

1) $\text{UUID} = 03000200 - 0400 - 0500 - 0006 - 000700080009$
   $\text{MN} (\text{Model Name}) = \text{Samsung SSD 850 EVO 250GB}$
   $\text{SN} (\text{Serial Number}) = S21RNXAGA03016Z$

2) $C = E_{K_{built-in}}(\text{UUID}||\text{MN}||\text{SN}) = 0x71efab84c9d810d8e810...$

3) $K_{PRAGMA} = \text{Base64}(SHA512(C)) = \text{vaQP/y1VSZWD/…==}$

- PRAGMA key generation codes \textit{temporarily appear} in memory to protect the key generation procedure.
  - Dynamic analysis is necessary.
Summary of encryption process

1. Collect the device specific information.
2. Generate PRAGMA key from the collected information.
3. Get the user sequence number (the only external parameter for KakaoTalk) from the server after the user logins.
4. Generate encryption key and IV.
How to securely generate the encryption key

• KakaoTalk might use the following strategies.
  ① Security by obscurity
     ➢ (AES → SHA512 → Base64 → MD5 → Base64 → MD5)
  ② Moving target
     ➢ Use of device specific parameter
  ③ Updatable key generation
     ➢ Updatable built-in key
  ④ External user data
     ➢ Best security practice for disk encryption
How to securely generate the encryption key

• KakaoTalk might use the following strategies.
  ① Security by obscurity
  ② Moving target
  ③ Updatable key generation
  ④ External user data
    ➢ Best security practice for disk encryption

However, we might easily obtain external user data.
How to obtain user sequence number

1. Brute-force search to obtain the user sequence number of the target user.

2. Smarter way to obtain the user sequence number using automatic friend addition with the target user’s phone number.

Local parameters: UUID, Model name, ...

External parameters: User sequence number
Brute-force search

• User sequence number is smaller than 300 million.
  – When we signed up the KakaoTalk at Jun 21\textsuperscript{st}, the user sequence number is given 251,582,414.

• Through the brute-force attack, the user sequence number can be obtained within \textbf{8.4 hours} (10,000 trial per second).

• The user sequence number is related to the date when the user signed-up.
  – If you figure out the victim’s registration date of KakaoTalk, guessing range of the user sequence number can be narrowed.
Smarter way to get the user sequence number

Assumption: Target user’s phone number is given.
1. Add the target user as friend in KakaoTalk with the target user’s phone number.

Checked by default
Smarter way to get the user sequence number

2. The target user’s information is added to our contact database file that we can freely access.

3. We can extract the target user’s user sequence number from the user’s record in the contact database file.

<table>
<thead>
<tr>
<th>User ID</th>
<th>Type</th>
<th>NickName</th>
<th>ProfileImageUrl</th>
<th>SocialProfileImageUrl</th>
<th>StatusMessage</th>
<th>LinkedServices</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>:)</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>AJ</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>BuJy</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>Chan</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>David</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>Geumhwan</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://th-p.talk.kakao">http://th-p.talk.kakao</a>...</td>
<td><a href="http://p.talk.kakao">http://p.talk.kakao</a>...</td>
<td></td>
</tr>
</tbody>
</table>

User sequence number: 151741
Demo – KakaoTalk
How can we securely protect the external parameters?

1. User sequence number is not random, but sequential.
2. User sequence number is used as the primary key for contact database.
Use of secure external parameter

1. User sequence number is not random, but sequential.
   - Use a large space of random numbers for external parameters.

2. User sequence number is used as the primary key for contact database.
   - Use the external parameter for the encryption key purpose only.
Example 2. NateOn

KakaoTalk

NateOn

QQ
Demo – NateOn
Example 3. QQ

KakaoTalk

NateOn

QQ
Experiment environment

- Environment and tools
  - VMware workstation 12
  - Windows 7
  - OllyDbg v1.10
  - QQ (version 2.1.1369.0)

- Unfortunately, the encrypted database of QQ cannot be decrypted without the server
DB format and encryption algorithm

OLE Structure

TEA
Encryption key generation algorithm

1) $PW_u = abc1234$

With $TGT$ Key, QQ Messenger decrypts the encrypted database using TEA with modified block cipher mode.

* Ticket Granting Ticket or Ticket to Get Tickets ($TGT$) is a small, encrypted identification file with a limited validity period. (https://en.wikipedia.org/wiki/Ticket_Granting_Ticket)
Encryption key generation algorithm

1) $PW_u = abc1234$

Because $TGT$ Key is changed after some period (8 hours), we cannot decrypt without the connection to the server.

QQ decrypts the encrypted database using TEA with modified block cipher mode.

* Ticket Granting Ticket or Ticket to Get Tickets (TGT) is a small, encrypted identification file with a limited validity period. (https://en.wikipedia.org/wiki/Ticket_Granting_Ticket)
Agenda

1. Introduction
2. Our analysis framework
3. Analysis of example IMs
4. Conclusion
Summary of analysis results

<table>
<thead>
<tr>
<th>Messenger</th>
<th>Local parameters</th>
<th>External parameters</th>
<th>Decryption without user auth.</th>
</tr>
</thead>
<tbody>
<tr>
<td>KakaoTalk</td>
<td>Device specific</td>
<td>User seq. number</td>
<td>O</td>
</tr>
<tr>
<td>NateOn</td>
<td>Fixed string</td>
<td>User seq. number</td>
<td>O</td>
</tr>
<tr>
<td>QQ</td>
<td>X</td>
<td>TGT key</td>
<td>X</td>
</tr>
</tbody>
</table>

• In KakaoTalk and NateOn, we can obtain messages without the user’s password.
• In QQ, the encryption key for chat database files is stored at the server.
Big Brother is always watching you.

Thank you for listening our talk.