Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.5, Oct. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.5.753

macOS용 카카오톡 데이터베이스 복호화 방안*

박 범 준,[1†] 이 상 진[2‡]
[1,2]고려대학교 (대학원생, 교수)

Decryption of KakaoTalk Database for macOS*

Beomjun Park,[1†] Sangjin Lee[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

국내 메신저 중 카카오톡이 가장 높은 점유율을 보유하고 있다. 그만큼 카카오톡의 대화 내용은 디지털포렌식에서 중요한 증거가 되고 있는데, 대화 내용이 사용자 기기에 암호화된 데이터베이스 형태로 저장되어 있다. 또한 macOS는 디스크 암호화 기능이 기본적으로 활성화되어 있어 접근이 어렵다는 특성을 가지고 있다. Windows용 카카오톡 데이터베이스의 복호화 방법은 연구되었지만, macOS용 카카오톡에 대해서는 복호화 방법이 연구된 바 없다. 본 논문에서는 macOS용 카카오톡 데이터베이스와 수신 파일에 대한 복호화 방안과 카카오톡의 UserID가 생성되는 특성을 이용하여 효율적인 전수조사 방안을 제시하고 Windows용 카카오톡과 비교 검토하여 공통점과 차이점을 살펴보았다. 본 논문의 결과는 macOS를 이용한 범죄를 수사할 때 사용자의 행위 및 사건의 전개를 분석하는 데에 활용될 수 있을 것으로 기대된다.

ABSTRACT

KakaoTalk has the highest market share among domestic messengers. As such, KakaoTalk's conversation content is an important evidence in digital forensics, and the conversation is stored in the form of an encrypted database on a user's device. In addition, macOS has the characteristic that it is difficult to access because the disk encryption function is basically activated. The decryption method of the KakaoTalk database for Windows has been studied, but the decryption method has not been studied for KakaoTalk for macOS. In this paper, research the decryption method of the KakaoTalk database for macOS and a way to Brute-Force plan using the characteristics of KakaoTalk's UserID and compare it with KakaoTalk for Windows to examine the commonalities and differences. The results of this paper are expected to be used to analyze users' actions and events when investigating crimes using macOS.

Keywords: Kakaotalk, macOS, Digital Forensic

I. 서 론

최근 디지털 기술의 발전과 함께 스마트폰과 컴퓨터의 사용이 급증하면서 사람들은 전화보다 메신저 애플리케이션을 통해 많이 소통하고 있다. 그중에서도 카카오톡은 국내 메신저 시장의 95% 이상을 점유하고 있고, 대부분의 국민들이 카카오톡을 이용하여 텍스트, 이미지, 동영상 등의 데이터를 주고받으며 일상을 공유하고 있다. 또한 메신저의 기본 기능뿐만 아니라 금융이나 생활 정보 제공 등의 기능이 계속 추가되고 있어 카카오톡의 데이터베이스에는 "사용자의 일상이 담겨있다"라고 할 정도이다.

따라서 카카오톡의 대화 내용은 범죄 수사 관점에

Received(08. 03. 2023), Modified(09. 18. 2023), Accepted(09. 19. 2023)
* 이 논문은 2023년도 정부(문화체육관광부)의 재원으로 한국저작권보호원의 지원을 받아 수행된 연구임(No 2023. 저작권 특화 디지털포렌식 전문인력 양성사업)
† 주저자, h4ck4s3cur1ty@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

서 매우 중요한 디지털 증거가 되고 있다. 이러한 중요 정보를 담고 있는 카카오톡의 데이터베이스는 사용자 기기에 저장되지만, 암호화 되어 있어 일반적인 디지털 포렌식 방법으로는 해당 데이터를 분석하는데 한계가 있다. 또한 macOS는 디스크 암호화 기능인 FileVault가 기본적으로 설정되어 있다. 사용자 기기가 로그온된 상태거나 사용자 암호를 알고 있는 상태가 아니라면, 모든 데이터가 암호화 되어 있기 때문에 사용자의 데이터 획득이 어렵다는 특성이 있다. 이러한 특성으로 인해 Windows용 카카오톡 데이터베이스의 복호화에 대한 연구가 진행되었으나[1], macOS용 카카오톡 데이터베이스의 복호화 연구는 아직 진행된 바 없다. 본 논문에서는 masOS용 카카오톡 데이터의 획득 및 데이터베이스 복호화 방안을 제시하고, Windows용 카카오톡과 비교하여 공통점과 차이점을 살펴본다.

본 논문의 구성은 다음과 같다. 2장에서는 다양한 플랫폼에서 카카오톡 분석에 대한 기존 연구에 관해 기술한다. 3장에서는 macOS 분석 환경과 분석 데이터를 식별하고 데이터베이스와 멀티미디어 파일의 복호화 방안을 제시한다. 4장에서는 효율적인 전수조사 방법을 제시한다. 5장에서는 Windows와, macOS 상의 카카오톡 데이터베이스를 비교 분석하여 공통점과 차이점을 살펴본다.

II. 관련 연구

기존 메신저에 관한 연구는 모바일과 PC에서 사용자 관련 정보 및 대화 내역이 저장된 데이터베이스의 복호화와 관련 아티팩트에 대한 연구가 진행되었다.

윈도우 환경에서 카카오톡 메신저의 백업 프로세스를 분석해, 백업 데이터베이스의 암호화 과정과 암호키 생성 과정을 밝혀내거나[2] 카카오톡과 같은 인스턴스 메신저의 데이터베이스 암·복호화 동작 과정 분석을 통해 사용자의 패스워드 없이 암호화된 데이터베이스를 복호화하는 방법에 대한 연구[3]가 진행되었다. 안드로이드 환경과 윈도우 환경에서 카카오톡 아티팩트를 비교 분석하는 연구[4]도 진행되었다. 이 연구에서는 각 환경에서만 남는 아티팩트를 특정하고 윈도우 환경에서의 메시지 유형별 행위를 명시했다.

안드로이드상의 카카오톡 데이터와 아티팩트 유형을 분석하고, 백업파일을 통해 삭제된 메시지를 복구하는 방법에 대한 연구[5]도 진행되었다. 이 연구에서는 WhatsApp과 DB 암호화상태에 대해서도 비교 분석했으며, 각 어플리케이션에서의 삭제된 메시지를 시각 분석하는 방법에 대해 기술했다.

이외에도 윈도우 환경에서 사용되는 메신저의 AES 키 생성 및 데이터베이스 암호화 방식, 패스프레이즈 생성 과정에 대해 연구하고 삭제된 메시지를 식별하는 방법 및 복구 방법을 제시하는 연구[6]도 진행되었다.

윈도우와 모바일 환경에서 Slack과 Discord의 아티팩트를 분석하여 각 메신저의 데이터가 어떻게 활용될 수 있는지에 대해 가상 시나리오를 이용해 활용점을 제시하는 연구[7]도 진행되었다.

윈도우 환경에서 인스턴스 메신저의 데이터를 획득하고 분석하는 연구 이외에도 모바일 환경에서의 메신저 데이터 획득 방안을 제시하고, 획득한 데이터의 구조와 암호화 프로세스에 대한 연구[7]가 진행되었다. 이 연구에서는 로그인 패스워드 및 삭제된 메시지 복구 방안에 대해서도 연구했으며, 다양한 방법으로 삭제된 메시지를 복구하는 방법을 제시했다. 또한, SQLite 데이터베이스를 암호화하기 위한 확장 모듈인 SQLCipher를 이용한 암호화 기능을 제공하는 메신저의 암호화 과정을 분석하고 복호화 후 주요 아티팩트를 선별 및 정리하는 연구[8]도 진행되었다.

모바일 환경에서는 SQlite의 저널링 방식을 이용한 메신저 삭제된 데이터 복구 가능성에 대한 연구[9]가 이루어졌다. 이 연구는 메신저별 설정되어있는 저널 모드에 따라 삭제된 메시지 복구 방법이 다르다는 것을 언급하며, 메신저별 저널 모드를 확인하고 메시지 삭제 방법에 따른 복구 가능 여부를 제시했다.

이외에도 윈도우 환경과 모바일 환경에서 WeChat 메신저의 데이터를 획득한 후 각 환경에서의 데이터 복호화 과정을 제시하고, WeChat 삭제된 기능을 살펴보며 환경별 데이터베이스를 분석해 삭제된 메시지 복구를 시도하는 연구[10]가 진행됐다.

이렇게 살펴본 선행연구는 아래와 같은 한계점이 있는 것을 확인했다.

첫째, 인스턴스 메신저의 디지털 포렌식 연구에 대한 분석 환경이 윈도우와 모바일에만 국한되어 있지만, 디지털 범죄 수사에서 macOS의 비율이 점차 증가하고 있다. 따라서 macOS 환경에서의 인스턴스 메신저 관련 연구는 더욱 중요하다.

둘째, 인스턴스 메신저에 대한 디지털 포렌식 연

구는 주로 시장 점유율이 낮은 인스턴스 메신저에 국한되어 있음을 확인했다.

따라서 본 연구에서는 선행 연구의 한계점을 극복하기 위해 연구 환경 및 대상을 설정했다. 최근 macOS 환경에서의 디지털 범죄 수요가 증가한다는 점과 함께 고려하여 국내 메신저 시장 점유율이 가장 높은 메신저를 연구 대상으로 선정했다.

III. 분석 대상 데이터 식별 및 복호화 방안

3.1 분석 환경

본 논문에서는 macOS용 카카오톡을 분석하기 때문에 macOS 환경에서 분석을 진행했다. 카카오톡 데이터의 암호화 및 복호화 과정을 정적 및 동적으로 분석하기 위해 IDA Pro를 사용했다. 복호화된 데이터베이스를 분석하기 위해 DB Browser for SQLite와 SQLCipher를 사용했다. 암·복호화 과정을 증명 코드로 구현하기 위해 Python을 사용했다.

Table 1. Analysis Target

Classification	Detail
Model	Macbook Air (M2, 2022)
OS	macOS Ventura 13.4
Kakaotalk	3.1.9(1025)

Table 2. Analysis Tools

Name	Version
IDA Pro	8.2.230124
Python	3.11
DB Browser for SQLite	3.12.2
SQLCipher	3.15.2

3.2 분석 데이터 식별

macOS용 카카오톡 데이터의 저장 경로는 Table 3과 같다. 카카오톡이 실행되는 Mach-O 바이너리 파일은 Apple의 ARM 기반 M 프로세서 호환성을 위해 x86과 ARM 바이너리가 합쳐진 상태로 존재한다는 것이 특징이다. 대화내역 등 주요 정보는 모두 데이터베이스에 저장되어 있으며, 주고받은 멀티미디어 파일은 사용자별 디렉터리에 저장된다. 이는 모두 암호화된 상태로 저장되어 있다.

Table 3. KakaoTalk Data Path

Name	Path
KakaoTalk Binary	/Applications/KakaoTalk.app/Contents/MacOS/KakaoTalk
Database	/Users/{username}/Library/Containers/com.kakao.KakaoTalkMac/Data/Library/ApplicaitionSupport/com.kakao.KakaoTalkMac/{dbname}
Multimedia File	/Users/{username}/Library/Containers/com.kakao.KakaoTalkMac/Data/Library/ApplicaitionSupport/com.kakao.KakaoTalkMac/{userDirectory}/{chatId}/{messageId}.{extension}

3.3 데이터베이스 복호화 방안

macOS용 카카오톡의 대화 내용이 저장되어 있는 데이터베이스의 파일명은 hashedDeviceUUID와 UserID값 그리고 특정 문자열을 조합하여 키 파생 함수인 PBKDF2(Password-Based Key Derivation Function)를 통해 생성된다. hashedDeviceUUID는 DeviceUUID를 이용하여 생성되고, 최종적으로 데이터베이스 파일명 생성

Table 4. Process of generating database file name

Input: UserID, DeviceUUID
Output: DB_NAME

```
KEY1 ← SHA1(DeviceUUID)
KEY2 ← SHA256(DeviceUUID)
hashedDeviceUUID ←
        Base64_Encode(KEY1 + KEY2)
KEY3 ← ".".join(["","F",UserID,"A",
        "F",DeviceUUID[::-1],"","","|"])
dkLen ← 128
iterations ← 100,000
DB_NAME ←
    PBKDF2_HMAC("SHA256",
    KEY3, hashedDeviceUUID[::-1],
    iterations, dkLen)
return DB_NAME
```

과정은 Table 4와 같다.

데이터베이스는 SQLCipher로 암호화 되어 있으며, 이를 복호화하기 위해서는 복호화 키가 필요하다. 복호화 키는 파일명 생성 과정과 동일하게 hashedDeviceUUID와 UserID값 그리고 특정 문자열을 조합하여 키 파생 함수인 PBKDF2를 통해 생성된다. 다만 특정 문자열의 구성에 차이가 있다. 데이터베이스 복호화 키 생성 과정은 Table 5와 같다.

Table 5. Database decryption key generation process

Input:	UserID, DeviceUUID
Output:	DB_KEY
1:	KEY1 ← SHA1(DeviceUUID)
2:	KEY2 ← SHA256(DeviceUUID)
3:	hashedDeviceUUID ← Base64_Encode(KEY1 + KEY2)
4:	KEY3 ← "F".join(["A",hashedDeviceUUID, "|","F",DeviceUUID[:5],"H", UserID,"|",DeviceUUID[7:]])
5:	dkLen ← 128
6:	iterations ← 100,000
7:	DB_KEY ← PBKDF2_HMAC("SHA256", KEY3[::-1], DeviceUUID[10:], iterations, dkLen)
8:	return DB_KEY

3.4 수신 파일 복호화 방안

카카오톡을 통해 주고받은 이미지 등의 멀티미디어 파일은 카카오톡의 유저 데이터가 저장되는 디렉터리에 유저별로 다른 디렉터리를 생성하여 저장한다. 생성되는 디렉터리명은 각 유저의 UserID에 대해 SHA512 해싱을 수행한 값이다. 해당 값의 생성 과정은 Table 6과 같다.

위 과정으로 생성된 유저별 디렉터리에는 해당 유저가 참여 중인 채팅방별로 다른 디렉터리를 생성하여 해당 채팅방에서 수신된 파일을 저장한다. 생성되는 디렉터리명은 각 채팅방의 고유값인 chatRoomId 값에 대해 SHA1 해싱을 수행한 값이다. 해당 값의 생성 과정은 Table 7과 같다.

Table 6. Process of creating directory names for each user

Input:	UserID
Output:	FOLDER_NAME
1:	hashedUserID ← SHA512(UserID)
2:	FOLDER_NAME ← hashedUserID[20:40]
3:	return FOLDER_NAME

Table 7. Process of creating directory names for each chat room

Input:	chatRoomId
Output:	FOLDER_NAME
1:	hashedchatRoomId ← SHA1(chatRoomId)
2:	FOLDER_NAME ← hashedchatRoomId
3:	return FOLDER_NAME

위 과정으로 생성된 채팅방별 디렉터리에는 해당 채팅방에서 수신한 파일이 암호화 되어 저장된다. 저장되는 멀티미디어 파일 종류는 [Table 8]과 같다.

암호화된 파일명은 Table 8의 확장자별 고유 식별자와 해당 채팅의 고유값인 chatLogId 값을 조합하여 SHA1 해싱을 수행한 값이다. 해당 값의 생성 과정은 Table 9와 같다.

Table 8. Encrypted multimedia's file extensions and unique identifiers

Extension	Unique Identifier	Detail
.thm	t	Image Thumbnail File
.img	p	Image File
.aud	a	Voice Record File
.vid		Video File
.txt	l	Text File

Table 9. Process of generating encrypted multimedia filename

Input:	chatLogId, Extension Eigenvalue
Output:	FILE_NAME
1:	FILE_NAME ←
2:	SHA1(Extension Eigenvalue + chatLogId[::-1])
3:	return FILE_NAME

암호화된 파일의 첫 16바이트는 IV, 이후의 값은 암호문으로 이루어져 있다. KEY는 UserID를 PBKDF2를 통해 생성된 값을 SHA256 해싱을 수행한 값이다. 위에서 획득한 정보를 통해 AES-256-CBC 복호화를 진행하면 원본의 멀티미디어 데이터를 획득할 수 있다. 해당 과정은 Table 10과 같다.

Table 10. Process of decrypt multimedia data

```
Input:  UserID, ENCFILE
Output: DECFILE
IV ← ENCFILE[:16]
ENCDATA ← ENCFILE[16:]
KEY1 ← PBKDF2_HMAC("SHA256",
          UserID, UserID[::-1],
          iterations, dkLen)
KEY2 ← SHA256(KEY1)
DECDATA ← AES256/CBC/DEC
            (ENCDATA, KEY2, IV)
DECFILE ← DECDATA[256:]
return DECFILE
```

IV. 전수 조사 방안

카카오톡의 암호화된 데이터를 복호화하기 위해서는 UserID 값이 필요하다. 해당 값은 카카오톡 서버에 저장되어 있는 사용자의 고유값으로 아티팩트 분석으로는 획득할 수 없다. 하지만 UserID 값은 0부터 1씩 순차적으로 증가하기 때문에 전수 조사를 통해 UserID를 알아낼 수 있다.

4.1 기존 전수 조사 방안

Table 4의 데이터베이스 파일명 생성 알고리즘과 UserID의 특성을 이용하여 UserID 값을 획득하는 알고리즘은 Table 11과 같다.

Table 11. Brute-Force method using database file name generation algorithm

```
Input:  DeviceUUID, User_DB_NAME
Output: UserID
KEY1 ← SHA1(DeviceUUID)
KEY2 ← SHA256(DeviceUUID)
hashedDeviceUUID ←
          Base64_Encode(KEY1 + KEY2)
while (UserID ≤ sizeof(int))
    KEY3 ← ".".join(["."."F",UserID,"A",
              "F",DeviceUUID[::-1],"."."|"])
    dkLen ← 128
    iterations ← 100,000
    DB_NAME ←
        PBKDF2_HMAC("SHA256",
        KEY3, hashedDeviceUUID[::-1],
        iterations, dkLen)
    if (DB_NAME == User_DB_NAME)
        return UserID
```

하지만 위 방안으로 전수 조사하면, PBKDF2_HMAC 알고리즘의 큰 iterations 값으로 인해 연산 부하에 따라 전수 조사 속도가 현저히 저하된다.

4.2 효율적인 전수 조사 방안

Table 6의 유저 디렉터리 생성 알고리즘에 따르면, UserID 값에 대하여 SHA512 해싱을 수행하여 유저 디렉터리명을 도출한다. 이를 이용하여 전수 조사하면, 5.1에서의 암호화 알고리즘을 거치지 않고 UserID 값을 전수 조사할 수 있게 된다. 유저 디렉터리명을 통해 UserID 값을 전수 조사하는 알고리즘은 Table 12와 같다.

다음은 UserID 전수 조사 소요 시간 테스트에 활용된 PC사양과 기준 UserID 값이며, 알고리즘별 전수 조사 소요 시간은 Table 14와 같다.

Table 12. Brute-Force method using a user directory name generation algorithm

```
Input:  User_FOLDER_NAME
Output: UserID
hashedUserID ← SHA512(UserID)
while (UserID ≤ sizeof(int))
    hashedUserID ← SHA512(UserID)
    FOLDER_NAME ←
            hashedUserID[20:40]
return FOLDER_NAME
```

Table 13. Brute-Force test environment

Classification	Detail
CPU	I9-13900K
RAM	64GB
MultiThread	24 Core
Benchmark UserID	10,000,000

Table 14. Brute-Force time required for each algorithm

Algorithm	Duration of time
Database Filename	213538s
User Directory Name	3s

V. 데이터베이스 비교 분석

본 절에서는 Windows 환경과 macOS 환경에서 카카오톡의 복호화된 대화 내용 데이터베이스를 비교하여 각각 어떤 내용을 담고 있는지 살펴본다.

5.1 대화 내용 저장 방식

카카오톡의 대화 내용은 SQLite 데이터베이스 파일로 저장된다. Windows 환경에서는 대화방별로 데이터베이스 파일을 생성하나, macOS 환경에서는 하나의 데이터베이스에 모든 대화방의 대화 내용을 기록한다. 이러한 특성으로 인해, Windows용 카카오톡에서는 대화방을 나가면, 데이터베이스 파일 자체를 삭제하는 반면, macOS 환경에서는 대화방을 나가면 데이터베이스 레코드에서 해당 대화 내용을 삭제하는 방식으로 관리된다. 또한 SQLite의 "secure_delete" 옵션을 활성화하여 삭제된 레코드를 안전하게 관리하고 있음을 확인했다.

```
v4 = objc_retain(a2);
_objc_msgSend(v4, "executeUpdate:", CFSTR("PRAGMA secure_delete = 1"));
_objc_msgSend(v5, "executeUpdate:", CFSTR("PRAGMA journal_mode = wal"));
_objc_msgSend(v6, "executeUpdate:", CFSTR("PRAGMA synchronous = 1"));
_objc_msgSend(v7, "executeUpdate:", CFSTR("PRAGMA fullfsync = 1"));
```

Fig. 1. SQLite Database Options for macOS KakaoTalk

5.2 대화 내용 테이블 비교

macOS용 카카오톡의 대화 내용은 하나의 데이터베이스에 통합 관리되며, NTChatMessage 테이블에 저장된다. Windows용 카카오톡 대화 내용은 각 대화방별 데이터베이스의 chatLogs 테이블에 저장된다. 대화 내용이 저장되는 테이블의 구조는 Table 15와 같다.

Windows 환경에서는 삭제된 메시지를 암호화하기 때문에 암호화 여부를 "deleted" 컬럼과 "is_encrypt_delete_msg" 컬럼을 통해서 식별한다. macOS 환경에서는 삭제된 메시지가 평문으로 남아있기 때문에 별도의 식별자 컬럼을 두고 있지는 않지만, "type" 컬럼의 메시지 타입을 통해 해당 메

Table 15. chatLogs Table structure in KakaoTalk for Windows

OS	Classification	Detail
Windows macOS	logId	ChatLog ID
Windows macOS	authorId	Chat Author UserID
Windows macOS	type	Chat Type
Windows macOS	clientMsgId msgId	Message ID
Windows macOS	sendAt	Send Time
Windows macOS	message	Chat Message
Windows macOS	attachment	Attachment
Windows	prev_msg_missing	Prev Message Missing Flag
Windows	next_msg_missing	Next Message Missing Flag
Windows	deleted	Delete Flag
Windows macOS	prevLogId prevId	Prev ChatLog ID
Windows	feed_flags	Feed Flag
Windows	write_on_pc	Write On PC Flag
Windows	is_link_message	Link Message Flag
Windows macOS	referer	Referer
Windows macOS	supplement	-
Windows	rewrite	Rewrite Flag
Windows	is_encrypt_delete_msg	Delete Message Encrypt Flag
macOS	chatId	Chat ID
macOS	status	Chat Status
macOS	subType	-
macOS	contentFlag	Content Flag
macOS	extra	Extra Infomation
macOS	readAt	Read Time
macOS	localFilePath	-

시지가 삭제된 메시지임을 확인할 수 있다. 또한 macOS 환경에서는 읽은 시각을 기록하는 것을 볼 수 있다. 하지만 이는 대화 상대방이 읽은 시작까지 기록하는 것이 아닌 클라이언트가 읽은 시각을 기록하는 것으로 확인됐다.

5.3 삭제된 메시지

카카오톡에서는 채팅방에서 대화 내용을 모든 대화 상대에게서 삭제할 수 있는 기능을 제공한다. 이는 모든 OS에서 공통으로 사용할 수 있는 기능이다. 이 기능을 사용해서 Windows용 카카오톡에서 메시지를 삭제하면 데이터베이스에는 해당 대화 내용이 암호화된 상태로 저장되며, 해당 메시지의 타입이 "16385"로 기록되고, 식별자 컬럼에 별도의 식별자 값이 기록된다. macOS 환경에서는 해당 메시지가 평문인 상태로 저장되어 해당 메시지가 삭제된 메시지임을 판단할 수 없지만, Windows용 카카오톡과 마찬가지로 메시지 타입이 "16385"로 동일하게 기록되므로, 이를 통해 해당 메시지가 삭제된 메시지임을 알 수 있다.

VI. 결 론

카카오톡 데이터베이스는 "사용자의 일상이 담겨있다"라고 할 정도로 카카오톡은 스마트폰과 PC에서 널리 사용되고 있다. 카카오톡은 윈도우, 안드로이드, iOS 그리고 macOS까지 많이 사용되고 있다. 최근 macOS 사용이 증가함에 따라 macOS용 카카오톡의 분석 횟수도 많아지는 반면에 macOS용 카카오톡 데이터베이스 복호화 연구는 아직 진행된 바 없다.

본 논문에서는 macOS용 카카오톡 데이터 획득 및 데이터베이스 복호화 방안을 제시했다. 이를 통한 카카오톡 데이터 획득은 범죄 수사 관점에서 매우 중요한 단서가 될 것이다. 또한, macOS용 카카오톡의 데이터는 윈도우와 다르게 메시지를 삭제해도 암호화가 안 된다는 점이 의미 있다. 이 연구 결과는 macOS를 이용한 범죄를 수사할 때 사용자의 행위 및 사건의 전개를 분석하는 데에 활용될 수 있을 것으로 기대된다.

References

[1] Minuook Jo and Nam Su Chang. "Study on The Data Decryption and Artifacts Analysis of KakaoTalk in Windows Environment," Journal of the Korea Institute of Information Security & Cryptology, 33(1), pp. 51-61, Feb. 2023

[2] J. Choi, J. Park, and H. Kim, "Forensic analysis of the backup database file in KakaoTalk messenger," 2017 IEEE International Conference on Big Data and Smart Computing, pp. 156-161, Feb. 2017

[3] J, Choi, J. Yu, and H, Kim, "Digital forensic analysis of encrypted database files in instant messaging applications on Windows operating systems: Case study with KakaoTalk, NateOn and QQ messenger," Digital Investigation, vol. 28, pp. S50-S59, Apr. 2019

[4] Nabi Lee, "A Study on Artifacts Classification and Sentiment Analysis by Deep Learning in KakaoTalk Messages under Android and Windows Environments," Journal of Digital Forensics, 16(3), pp. 14-34, Set. 2022

[5] JongCheol Yoon and Yongsuk Park, "Forensic Analysis of chatting messenger service in KakaoTalk and Comparison Study of KakaoTalk and WhatsApp Artifacts," Journal of the Korea Institute of Information and Communication Engineering, 20(4), pp. 777-785, Apr. 2016

[6] Byungchul Youn, Soram Kim and Jongsung Kim, "Study on Malang Malang Talkafe Database Encryption Process and Recovering Its Deleted Messages on Windows," Journal of the Korea Institute of Information Security & Cryptology, 30(3), pp.

397-403, Jun. 2020
〔7〕 Sumin Shin, Eunhu Park, Soram Kim, and Jongsung Kim. "Artifacts Analysis of Slack and Discord Messenger in Digital Forensic," Journal of Digital Contents Society, 21(4), pp. 799-809, April, 2020
〔8〕 Giyoon Kim, Uk Hur, Sehoon Lee, and Jongsung Kim. "For Forensic Analysis of the Secure Instant Messenger SureSpot," Journal of Digital Forensics , 13(3), pp. 175-188, Sept, 2019
〔9〕 Sueun Jung and Sangjin Lee, "Recovering deleted data of encrypted data by SQLCipher," Master. Thesis, Korea University School of Cyber-security, Feb 2022.
〔10〕 Byungchan Jung, Jaehyeok Han, Hoyong Choi, and Sangjin Lee. "A Study on the Possibility of Recover ing Deleted Data through Analysis of SQLite Journal in Messenger Application," Journal of Digital Forensics, 12(2), pp. 11-20, Sept, 2018
〔11〕 Uk Hur, Myungseo Park, and Jongsung Kim. "Study on Improved Decryption Method of WeChat Messenger and Deleted Message Recovery Using SQLite Full Text Search Data," Journal of the Korea Institute of Information Security & Cryptology, 30(3), pp. 405-415, June, 2020

〈저자소개〉

박 범 준 (Beomjun Park) 정회원
2021년 4월~현재: ㈜플레인비트 연구원
2022년 3월~현재: 고려대학교 일반대학원 정보보안학과 석사과정
〈관심분야〉 디지털포렌식, 역공학

이 상 진 (Sangjin Lee) 종신회원
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~현재: 고려대학교 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
〈관심분야〉 디지털포렌식