JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 21, No. 4, pp. 799-809, Apr. 2020

Check for updates

디지털 포렌식 관점에서의 Slack 및 Discord 메신저 아티팩트 분석

신 수 민[1] · 박 은 후[1] · 김 소 람[2] · 김 종 성[3*]
[1]국민대학교 금융정보보안학과 석사과정
[2]국민대학교 금융정보보안학과 박사과정
[3]국민대학교 정보보안암호수학과 & 금융정보보안학과 부교수

Artifacts Analysis of Slack and Discord Messenger in Digital Forensic

Sumin Shin[1] · Eunhu Park[1] · Soram Kim[2] · Jongsung Kim[3*]

[1]Master's Course, Department of Financial Information, Kookmin University

[2]P.HD student, Department of Financial Information, Kookmin University

[3]Associate Professor, Department of Information Security, Cryptology, and Mathematics & Financial Information, Kookmin University

[요 약]

인스턴트 메신저는 텍스트, 음성, 사진 및 동영상 전송 등 다양한 기능을 제공하기 때문에 현대인들이 필수적으로 사용하는 애플리케이션이 되었다. 이러한 메신저는 사용자의 행위에 따라 다양한 정보가 기록되며 해당 정보는 디지털 포렌식 수사 시 중요한 증거로 활용될 수 있다. 그러나 메신저마다 저장하는 데이터가 다르고, 많은 양의 데이터가 혼재되어 있으므로 중요한 데이터를 선별하고 의미를 파악하는 것에 관한 체계적인 연구가 필요하다. 본 논문에서는 팀 협업용으로 많이 사용되는 인스턴트 메신저인 Slack과 Discord를 대상으로 분석을 진행했다. 모바일 애플리케이션과 PC 프로그램에서 메시지 수/발신, 공유한 파일, 메시지 채팅방, 사용자 계정 정보 등과 같은 주요 아티팩트의 위치를 파악하여 정리했다. 또한, 시나리오를 제시하고 포렌식적으로 활용될 수 있는 방안을 제시한다.

[Abstract]

Instant messenger is an essential application for modern people because it provides the services to send text, voice, photos and videos. Therefore, a variety of user data are recorded in a messenger app, and the data can be used as important evidence in digital forensic investigations. It is necessary to classify important data and figure out its meanings because the data stored in each messenger is different and a large amount of data is mixed. In this paper, we study instant messenger, Slack and Discord applications which are used as messengers for team collaboration. We have identified and organized locations of artifacts, such as received/sent messages, shared files, chat rooms and user account information in mobile application and PC program. In addition, we suggest a method of data utilization through scenario.

색인어 : 인스턴트 메신저, 디지털 포렌식, Slack, Discord, 아티팩트

Key word : Instant Messenger, Digital Forensic, Slack, Discord, Artifacts

http://dx.doi.org/10.9728/dcs.2020.21.4.799

Received 14 March 2020; **Revised** 15 April 2020
Accepted 25 April 2020

***Corresponding Author; Jongsung Kim**

Tel: +82-2-910-5750
E-mail: jskim@kookmin.ac.kr

Ⅰ. 서 론

2019년 한국정보화진흥원의 스마트폰 과의존 실태조사 결과에 따르면 스마트폰 이용자의 주 이용 콘텐츠는 메신저(94.7%)이다[1]. 메신저는 텍스트, 사진, 동영상 및 파일 공유 등의 편의 기능을 제공한다. 그러나 인스턴트 메신저의 편의성으로 인해 일부 범죄 활동에 메신저가 소통 수단이 되었고, 다양한 사건에 악용된 사례가 존재한다. 2019년 1월, 댓글 조작 사건에서 스마트폰 내에 삭제되지 않은 텔레그램 메시지의 일부가 주요 증거로 채택되어 유죄 입증에 결정적인 역할을 한 사건이 발생했다[2]. 2019년 2월, 성폭행 혐의를 부인하고 있던 가해자의 자택과 차량 등에 대한 압수 수색을 통해 다양한 전자기기를 확보했으며, 이를 복원해 피해자와의 메신지 대화 내용, 문자메시지 등 중요 증거를 획득해 범죄사실을 입증했다[3]. 또한, 2019년 11월, 뉴욕에서 3명의 학생이 학교 테러를 목적으로 무기를 불법 구매한 사실이 발각되었으며, 이러한 사실은 Discord 메신저를 통해 확인할 수 있었다. 메신저 데이터에서 구체적인 공격 계획과 날짜를 공유한 메시지가 발견되었으며, 이를 근거로 학생들을 체포하였다[4]. 이처럼 메신저가 직접적인 증거로 활용되는 사례가 증가함에 따라 메신저 데이터 분석은 디지털 포렌식 조사에 필수적 요소가 되었다. 하지만, 많은 양의 데이터 중에 수사에 필요한 데이터를 찾기는 매우 어려운 일이다. 따라서 애플리케이션의 데이터베이스, 캐시 데이터, 설정 정보와 같은 아티팩트를 선정하여 체계적으로 분석하는 연구가 필요하다.

본 논문에서는 Slack과 Discord 메신저 데이터의 아티팩트를 식별하여 포렌식적으로 의미 있는 데이터에 관해 제시한다. 2장에서 관련 연구를 소개하고 3장에서는 Slack 메신저 데이터에 대해 분석한 결과를 정리하고, 4장에서는 Discord 메신저 데이터에 대한 분석결과를 정리한다. 4장에서는 가상 시나리오를 통해 메신저 데이터의 디지털 포렌식 활용방안에 관해 제시하고, 마지막 5장에서 결론으로 마무리한다.

Ⅱ. 관련 연구

모바일 애플리케이션에 관한 연구들은 다양한 연구가 진행되고 있다. 포렌식적 활용 가치가 있는 데이터들에 대해 분석한 기존의 연구 결과는 다음과 같다. iOS의 WeChat에서 데이터를 획득하는 방법과 절차를 설명하고 주요 아티팩트를 분석한 연구가 제시된 바 있다[5]. iTunes 백업을 통해 WeChat의 데이터를 획득했으며, 음성 메시지, 텍스트 메시지, 공유한 사진이나 동영상 등을 각기 다른 폴더에 저장한다. 또한, Messenger, Hangouts, Line과 WhatsApp 애플리케이션의 주요 데이터 위치를 파악하여 분석된 바 있다[6]. Messenger는 일반 채팅은 암호화되지 않은 데이터베이스 파일에 저장되지만, 비밀 채팅은 설정 시간이 만료되면 데이터베이스 파일에서 모든 데이터가 삭제된다. Hangouts과 Line은 데이터베이스를 암호화하지 않은 상태로 저장하며, WhatsApp은 주요 데이터가 저장된 데이터베이스가 루팅된 기기에서는 암호화되어 있지 않지만 루팅되지 않은 기기에는 암호화되어 있는 것을 확인함으로써 결과를 보여주었다. 국내 랜덤 채팅 애플리케이션 6종에 대해 사용자 행위에 따른 수/발신한 메시지 내용, 시간, 친구 프로필 등을 확인하고 분석한 결과가 발표되었다[7]. SNS (Social Network Service)인 인스타그램에서 사용자 행위에 따라 생성되는 정보가 저장된 데이터베이스 파일과 XML 파일을 분석한 연구가 수행되었다[8].

암호화된 데이터를 획득하여 복호화 방안을 제시하고 주요 데이터를 분석한 연구도 존재한다. 강력한 암호화를 제공하는 안전한 인스턴트 메시지인 ChatSecure에서 SQLCipher 모듈로 암호화된 데이터베이스의 복호화 키 획득 방법을 제시하고, 데이터베이스에 저장된 데이터를 분석했다[9]. 보안 메신저인 Surespot에서 암호화된 데이터 세 가지를 확인하고, Android와 iOS에서 각각 복호화 방안을 제시하고 주요 데이터를 저장하는 Friends.sss 파일과 Message.sss 파일에 대해 분석하였으며, 로그인 패스워드와 삭제된 메시지 일부를 복구하는 방안을 제시한 연구가 수행되었다[10].

Ⅲ. Slack 데이터 분석

Slack (Searchable Log of All Conversation and Knowledge)은 2013년 8월에 출시한 클라우드 기반 팀 협업 애플리케이션으로 2019년 1월 기준으로 전 세계 250개국에서 50만 개 이상의 기업이 사용하고 있다[11]. 2019년 1월을 기준으로 구글 플레이에서의 애플리케이션 다운로드 수는 1,000만 이상이며, Android는 20.01.10.0 그리고 Windows는 4.3.4 버전까지 출시되었다.

계정이 Workspace 단위로 관리되며, Workspace는 다양하게 생성할 수 있다. 이메일 또는 URL을 공유함으로써 Workspace에 다른 사용자를 초대할 수 있다. Workspace 내에서 채널을 생성할 수 있으며, 채널의 공개 여부도 선택할 수 있다. 그리고 채널 내에서 단체 채팅, 일대일 채팅, 음성 통화와 사진, 동영상 및 문서 공유 등의 기능을 제공한다.

3-1 모바일 애플리케이션

메신저 데이터는 패키지 명 하위에 저장되며, Slack의 패키지 명은 com.Slack이다. 애플리케이션 데이터의 구조는 그림 1과 같다. 모든 데이터가 암호화되지 않은 상태로 저장되어있다.

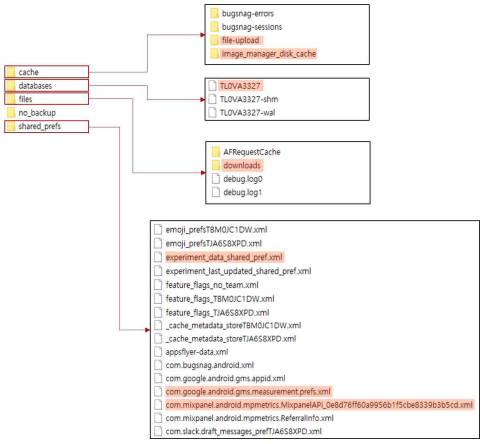

그림 1. 모바일 버전 Slack의 구조
Fig. 1. Structure of Slack Mobile Data

메신저 내에는 캐시 데이터, 주고받은 사진, 동영상, 파일 정보, 대화 내역, 사용자 정보, 스마트폰의 정보과 스마트폰에 애플리케이션을 설치한 시간 정보 등과 같은 데이터가 저장된다. 각 데이터를 저장하고 있는 파일명과 상세 내용은 표 1과 같다.

표 1. 모바일에 저장되는 파일명과 내용 및 경로
Table 1. File name, content and path in mobile app

File Path	Content
com.Slack/cache/ image_manager_disk_cache	profile photo, thumbnail information such as shared files
com.Slack/cache/file-upload	upload files
com.Slack/files/downloads download files	download files
com.Slack/databases/user_identity_team_ID	files information, channel information, message threads information, message information user information,
com.Slack/shared_prefs/com.mixpanel.android.mpmetrics.MixpanelAPI_0e8d76ff60a9956b1f5cbe8339b3b5cd.xml	user ID, team ID, application version, device information, android version
com.Slack/shared_prefs/com.google.android.gms.measurement.prefs.xml	application installation time, application runtime
com.Slack/shared_prefs/experiment_last_updated_shared_pref.xml	Workspace last update time

1) image_manager_disk_cache 디렉터리

프로필 사진, 채팅방에 사용자가 공유한 사진이나 친구가 공유한 사진의 썸네일 정보가 남아있다. 그러나 파일의 확장자는 모두 '.0'이기 때문에 내부의 시그니처를 확인해 이에 대응하는 확장자로 변경해야 원본 파일을 획득할 수 있다.

2) file-upload / downloads 디렉터리

Slack을 통해 사용자가 채팅방에 업로드한 사진, 동영상, 문서의 원본이 file-upload 디렉터리에 저장되어있다. downloads 디렉터리에는 사용자가 Slack에서 다운로드한 파일이 저장되어있다.

3) user_identity_team_id 데이터베이스

user identity team ID로 생성되는 데이터베이스 파일에는 6개의 테이블에 의미 있는 데이터가 존재한다. 표의 종류와 저장되는 데이터는 표 2와 같다.

표 2. 데이터베이스 내의 전체 테이블 정보
Table 2. Entire Table information in database

Table Name	Content
files	shared file information
message_threads	thread information (thread :message comment function)
message	shared message information
messaging_channels_counts	message channel information such as channel type, unread message etc
messaging_channels	message channel information
users	user information such as user ID, name, phone number etc

o 'files' 테이블

데이터베이스 파일 내 'files' 테이블에 저장되는 데이터는 표 3과 같다. 파일의 삭제 여부, 파일 생성시간, 파일 이름, 파일의 크기, 발신자 ID 등을 포함한다. 파일 삭제 여부는 deleted 컬럼에 나타나며, 속성값은 파일이 삭제되지 않았으면 0이고 파일이 삭제되었으면 1이다. file_blob 컬럼은 JSON 형식으로 기록되어 있으며, 대부분 데이터를 포함한다. 파일 생성시간, 파일 유형, 파일 이름 및 파일 크기 등은 Unix Time 형태이다.

표 3. 데이터베이스 내의 files 테이블 정보
Table 3. files Table information in database

Columns	Content	Remarks
deleted	delete file	0 : not deleted file 1 : deleted file
file_blob	file creation time file type, file name, file size, sender ID etc	JSON

o 'message_threads' 테이블

thread는 메시지에 대한 댓글 기능이다. 'message_threads' 테이블에 저장되는 데이터는 표 4와 같다. thread 작성 시간, 채널 ID, 메시지 수/발신 시간, thread 내용 및 thread를 작성한 사용자의 ID 등을 포함한다. ts 컬럼은 thread의 시간이 Unix Time 형태로 기록되어 있으며, msg_send_state 컬럼은 thread의 전송 상태를 나타내는 것으로, 1이면 읽음, 4인 경우 읽지 않음, 5인 경우 전송실패 의미한다. thread_ts 컬럼은

thread가 작성된 메시지의 수/발신 시간을 나타낸다. message_blob 컬럼은 JSON 형식으로 thread를 작성한 사용자 ID, thread의 내용, thread를 작성한 시간 등이 포함한다.

표 4. 데이터베이스 내의 message_threads 테이블 정보
Table 4. message_threads Table information in database

Columns	Content	Remarks
ts	thread start time	Unix Time
channel_id	channel ID	
msg_send_state	thread transmission status	1 : read 4 : not read 5 : not send
thread_ts	message receive/send time	Unix Time
message_blob	thread creation ID, thread content, thread creation time, message receive/send time etc	JSON

o ‘message’ 테이블

‘message’ 테이블에 저장되는 데이터는 표 5와 같다. 메시지의 수/발신 시간, 채널 ID, 메시지 전송 상태, 메시지 유형, 사용자 ID 및 평문으로 기록된 메시지의 내용을 포함한다. msg_send_state 컬럼은 메시지의 전송 상태를 나타내는 것으로 1은 읽음, 4는 읽지 않음, 5는 전송되지 않음을 의미한다. JSON 형식으로 된 message_blob 열에는 메시지 내용이 평문으로 저장되어있다.

표 5. 데이터베이스 내의 message 테이블 정보
Table 5. message Table information in database

Columns	Content	Remarks
ts	message receive/send time	Unix Time
channel_id	channel ID	
msg_send_state	message transmission status	1 : read 4 : not read 5 : not send
message_blob	message content, message sender ID etc	JSON
subtype	message type	channel_join, channel_name, channel_purpose, channel_topic, group_join etc
user_id	user ID	

o ‘message_channel_counts’ 테이블

‘message_channel_counts’ 테이블에 저장되는 데이터는 표 6과 같다. 채널의 유형, 읽지 않은 메시지 존재 여부, 마지막 메시지 수/발신 시간과 읽지 않은 메시지의 개수를 포함한다. 채널 유형을 의미하는 channel_type 컬럼은 일반 채널일 경우에는 PUBLIC, 비공개 채널일 때 PRIVATE, 일대일 채팅은 DM (Direct Message), 그룹 메시지는 MPDM (Multi Party Direct Message)으로 나타난다. is_unread 컬럼은 읽지 않은 메시지가 존재하지 않을 때는 0, 존재할 때는 1이다.

표 6. 데이터베이스 내의 messaging_channel_counts 테이블 정보
Table 6. messaging_channel_counts Table information in database

Columns	Content	Remarks
channel_type	channel type	PUBLIC : Public Channel PRIVATE : Private Channel DM : Direct Message MPDM : Group Message
is_unread	whether unread message exists	0 : unread message does not exist 1 : unread message exists
latest_ts	last receive/send message time	Unix Time
unread_count	the number of unread messages	

o ‘messaging_channels’ 테이블

‘messaging_channels’ 테이블에 저장되는 데이터는 표 7과 같다. 채널 ID와 사용자의 ID, 마지막으로 채널을 읽은 시간 등을 포함한다. type 컬럼은 메시지의 유형을 의미하며, 0은 일반 채널, 1은 비공개 채널, 2는 일대일 채팅, 3은 그룹 메시지를 나타낸다. msg_channel_blob 컬럼은 JSON 형식이고 채널 생성시간, 채널 ID, 마지막으로 메시지를 읽은 시간 등을 포함한다. 공개 채널, 비공개 채널과 그룹 메시지일 경우에는 채팅방에 속한 사용자 ID가 모두 기록되며, 채팅방을 생성한 사용자의 ID도 기록된다.

표 7. 데이터베이스 내의 messaging_channels 테이블 정보
Table 7. messaging_channels Table information in database

Columns	Content	Remarks
msg_channel_id	channel ID	
name_or_user	user ID	
type	message type	0 : Public Channel 1 : Private Channel 2 : Direct Message 3 : Group Message
msg_channel_blob	channel creation time, channel ID, user ID, last read message time etc	JSON

o ‘users’ 테이블

‘users’ 테이블에 저장되는 데이터는 표 7과 같다. 사용자의 ID, 이름, 핸드폰 번호, 이메일 주소, 마지막으로 업데이트한 시간 및 사용자가 프로필에 설정해 놓은 이모티콘 등이 저장된다. 이때, 마지막 업데이트 시간은 Unix Time 형태이다.

표 8. 데이터베이스 내의 users 테이블 정보
Table 8. users Table information in database

Columns	Content	Remarks
id	user ID	
name	user name	
deleted	able or disable	true : disable
updated	last update time	Unix Time
presence	active or not	active away : not active
tz	region of timezone	Asia/Seoul etc
tz_label	Standard timezone	Korea Standard Time etc
tz_offset	time offset	
team_id	team ID	
profile_first_name	first name of user	
profile_last_name	last name of user	
profile_current _status	profile message of user status	
profile_current_ status_emoji	profile emoji of user status	
profile_current_ status_expiration	expiration time of profile status	
profile_phone	user phone number	
profile_real_name	real user name	
profile_display_ name	user nickname	
profile_email	user email address	
profile_title	purpose of user at a workspace	

4) shared_prefs 디렉터리

com.mixpanel.android.mpmetrics.MixpanelAPI_0e8d76ff60a9956b1f5cbe8339b3b5cd.xml 파일에는 사용자의 ID, 팀 ID, 애플리케이션의 버전, 스마트폰 정보 및 안드로이드 버전이 저장되어있으며, com.google.android.gms.measurement.prefs.xml 파일에는 애플리케이션을 설치한 시간과 실행한 시간이 Unix Time 형태로 저장되어있다. 마지막으로 experiment_last_updated_shared_pref.xml 파일에는 Workspace별 마지막 업데이트 시간이 Unix Time 형태로 저장되어있다.

Slack에서 메시지를 삭제한 후, 데이터베이스를 확인하면 레코드가 삭제된다. 메시지 삭제 후 앱을 종료한 뒤 다시 실행하기 전까지 데이터베이스를 수집하면 WAL 파일에 삭제된 메시지 내용뿐만 아니라 메시지의 수/발신 시간, 사용자 ID 등의 데이터가 남아있다. 그러나 앱을 다시 실행한 뒤 데이터베이스를 획득하면 삭제된 메시지에 대한 데이터가 사라진다. 그리고 Raw-data 상에도 존재하지 않는다.

3-2 PC 프로그램

PC 버전 응용 프로그램은 'C:₩Users₩[User Name]₩AppData₩Roaming₩Slack' 하위 경로에 데이터가 저장되며, 암호화되어 있지 않다. 데이터 저장구조는 (그림 2)와 같으며, 중요 정보는 Cache 디렉터리, indexedDB 디렉터리, Local Storage 디렉터리, logs 디렉터리와 storage 디렉터리에 저장되어있다.

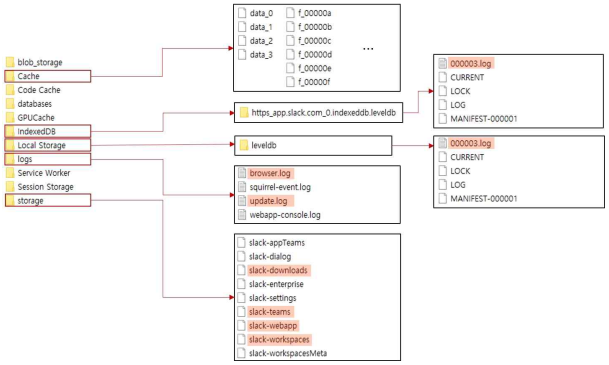

그림 2. PC 버전 Slack의 구조
Fig. 2. Structure of Slack PC data

1) Cache 디렉터리

Cache 디렉터리의 데이터는 Chrome 브라우저의 Cache 파일 데이터 구조와 동일하다[12]. data_0, data_1, data_2, data_3, f_(0){4}([0-9]|[a-z]){2} 파일들로 구성되어 있고 data_0에는 Cache 데이터 인덱스 정보가 저장되며, data_1~ data_3 파일은 URL, Cache 데이터를 포함한다. f_(0){4}([0-9]|[a-z]){2} 파일들은 프로필 사진이나 주고받은 사진, 동영상의 썸네일이다. data_0과 data_1~ data_3 파일을 통해 Cache 데이터의 파일 이름을 알 수 있다.

2) http_app.slack.com_0.indexeddb.leveldb 디렉터리

IndexedDB₩https_app.slack.com_0.indexeddb.leveldb 경로에 존재하는 [0-9]{6}.log 파일에는 그림 3과 같이 사용자 프로필에 대한 정보가 저장되어있다. real name, tz, title, phone, display_name 그리고 Skype 등과 같은 항목이 포함된다. real name은 사용자의 이름(Full name), tz는 사용자의 현재 시간대를 의미한다. title은 workspace의 목적(What I do), phone은 사용자의 전화번호(Phone number), display_name은 사용자가 설정한 이름이나 별명으로 채널이나 채팅방에 보이는 이름이다. Skype는 Skype 애플리케이션의 계정, status는 이모티콘과 글로 표현한 사용자의 상태, email은 사용자의 이메일 주소이다. image_original은 사용자의 프로필 사진이다. URL에 접속하면 프로필 사진의 원본을 볼 수 있다.

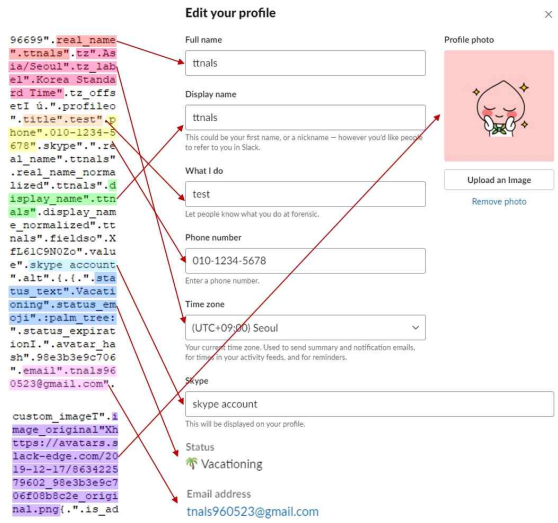

그림 3. PC에 저장된 사용자 정보
Fig. 3. User data stored in PC

3) leveldb 디렉터리

Local Storage₩leveldb 경로에 있는 [0-9]{6}.log 파일에는 실행한 시간, 마지막으로 활동한 팀 ID, 마지막으로 활동한 시간 정보와 프로그램을 종료한 시간이 저장되어있다. 시간 정보는 Unix Time 형태로 저장된다.

4) logs 디렉터리

browser.log 파일에는 버전 정보와 처음 다운로드한 시간 정보, update.log 파일에는 업데이트한 시간 정보가 저장되어있다.

5) storage 디렉터리

다운로드한 파일, Workspace, 팀 등에 관한 정보가 JSON 파일 형태로 storage 디렉터리 내에 저장되어있다. 그 중, slack-webapp 파일에 읽지 않은 메시지의 개수는 저장되지만, 메시지의 내용은 저장되지 않는다.

Ⅳ. Discord 데이터 분석

Discord는 2015년에 5월에 출시된 게이머를 위한 메신저 애플리케이션이다. PC 버전의 경우 따로 소프트웨어를 설치할 필요 없이 웹 브라우저에서도 사용할 수 있다. 2020년 1월을 기준으로 Google Play Store에서 다운로드 수는 5,000만 이상이며, Android는 10.2.5 그리고 Windows는 0.0.306 버전까지 출시되었다.

여러 사용자를 묶어 하나의 서버로 만들 수 있으며. 서버의 초대 코드를 생성한 후, 원하는 사용자를 초대할 수 있다. 서버는 가장 기본적이고 큰 개념이며, 한 서버에서 원하는 주제로 여러 개의 채널을 생성해서 대화할 수 있다. 채널의 종류는 채팅과 음성이 존재한다. 또한, 채널의 공개 여부를 선택할 수 있다. 이외에도 일대일 채팅, 음성 통화, 영상 통화, 사진, 동영상 및 문서 공유 등의 기능을 제공한다.

4-1 모바일 애플리케이션

메신저 데이터는 Discord의 패키지 명인 com.discord 하위에 저장된다. 애플리케이션 데이터의 구조는 그림 4와 같다. 모든 데이터가 암호화되지 않은 상태로 저장되어있다. Discord는 데이터베이스 형식으로 저장되지 않아 파일 분석 도구가 존재하지 않으므로 데이터를 헥사 뷰어를 통해 직접 확인해야 한다.

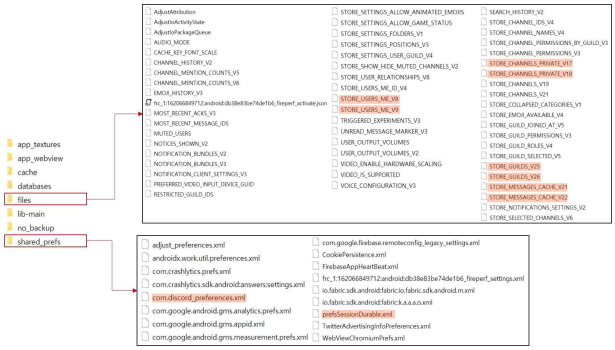

그림 4. 모바일 버전 Discord 구조
Fig. 4. Structure of Discord mobile data

각각의 경로에 저장된 주요 파일명과 내용은 표 9와 같다. 주고받은 사진, 동영상 및 파일 정보, 대화 명세, 사용자와 사용자의 친구 정보 등이 저장되어있다.

표 9. 모바일 버전 Discord에 저장되는 파일명, 내용 및 경로
Table 9. File name, content and path in mobile app

Path	File Name	Content
/data/data/com.discord/files	STORE_CHANNELS_PRIVATE_V* (* : number)	friend ID
	STORE_MESSAGE_CACHE_V* (* : number)	sender ID, message content, message receive/send time, attachment URL
	STORE_GUILDS_V* (* : number)	server name, server creation date, server creation time, server location
	STORE_USERS_ME_V* (* : number)	user email address, user name
/data/data/com.discord/shared_prefs	com.discord_preferences.xml	user email address, user ID
	prefsSessionDurable.xml	user email address

1) STORE_CHANNELS_PRIVATE_V* 파일

files 폴더 내 STORE_CHANNELS_PRIVATE_V* 파일

에는 연락처에 저장된 친구 ID가 저장되어있다.

2) STORE_MESSAGE_CACHE_V* 파일

메시지의 내용, 시간, 발신자 ID가 저장되어있다. 이때, 상대방이 읽지 않은 메시지는 저장되지 않는다. 사진, 동영상, 문서를 수/발신했을 경우에 남는 데이터는 표 10과 같다.

표 10. 사진, 동영상, 문서를 수/발신했을 때 저장되는 데이터
Table 10. Data of sending/receiving photo, video and document

Case	File	Content	Remarks
Sending/ Receiving Photo	STORE_MESSAGE _CACHE_V*	file name	
		original photo URL	
		thumbnail URL	
		sender ID	
		photo receive/send time	UTC+0
Sending/ Receiving Video		file name	
		video URL for playing	
		video URL for downloading	
		sender ID	
		video receive/send time	UTC+0
Sending/ Receiving Document		file name	
		document URL for downloading	
		sender ID	
		document receive/send time	UTC+0

사진을 전송한 경우에는 사진 파일명, 원본 크기의 사진을 볼 수 있는 URL, 사진의 썸네일 볼 수 있는 URL, 발신자 ID와 사진의 수/발신 시간이 저장된다. 동영상을 전송한 경우에는 동영상 파일명, 동영상을 시청할 수 있는 URL, 동영상의 시청은 불가능하고 다운로드 할 수 있는 URL, 발신자 ID와 동영상의 수/발신 시간이 저장된다. 마지막으로 문서를 전송한 경우에는 문서의 파일명, 해당 문서를 다운로드 할 수 있는 URL, 발신자 ID와 문서를 수/발신 시간이 저장된다. 이때, 모든 시간은 UTC+0으로 기록되어 있으므로 분석 시에는 한국 표준 시간인 UTC+9로 변경해야 한다.

3) STORE_GUILDS_V* /STORE_USERS_ME_V* 파일

서버의 이름, 서버 생성 날짜 및 시간과 서버 위치가 STORE_GUILDS_V* 파일에 저장된다. 이때, 생성 날짜 및 시간은 UTC+0으로 기록되어 있다. 서버 위치는 서버를 생성할 때, 사용자가 선택한 것으로 South Korea, Brazil, Europe 등 총 14개가 존재한다. STORE_USERS_ME_V*에는 사용자의 이메일과 사용자 명이 저장되어있다.

5) shared_prefs 디렉터리

com.discord_preferences.xml 파일과 prefsSessionDurable.xml 파일에는 Discord 가입 시 사용한 사용자의 이메일이 존재하며, 추가로 com.discord_preferences.xml 파일에는 사용자의 ID도 함께 존재한다.

Discord 메신저에서 삭제한 메시지는 메시지에 관한 정보가 저장되는 파일인 STORE_MESSAGE_CACHE_V* 파일에 저장되지 않는다.

4-2 PC 프로그램

PC 버전 응용 프로그램은 'C:₩Users₩[User Name]₩AppData₩Roaming₩Discord', 'C:₩Users₩[User Name]₩AppData₩Local₩Discord' 두 가지 경로에 데이터를 기록하며, 암호화되지 않은 상태로 저장한다. 하위 경로의 구조는 그림 5와 같다. 설치 관련 DLL 파일과 실행 파일 등은 'Local₩Discord'에 저장되고, 캐시 정보는 'Roaming₩Discord'에 저장된다.

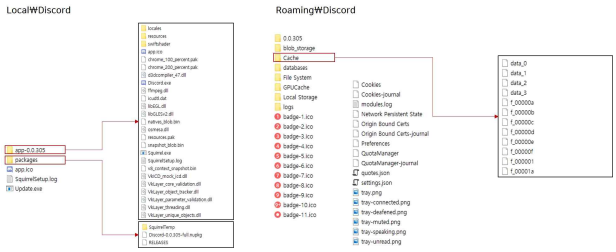

그림 5. PC 버전 Discord의 구조
Fig. 5. Structure of Discord PC data

1) Cache 디렉터리

Discord의 Cache 데이터도 또한 Slack과 마찬가지로 Chrome 브라우저의 Cache 파일 데이터 구조와 동일하다. data_0, data_1, data_2, data_3, f_(0){4}([0-9]|[a-z]){2} 파일들로 구성되어 있다. data_0에는 Cache 데이터 인덱스 정보가 저장되며, data_1~data_3 파일은 URL, Cache 데이터를 포함한다. f_(0){4}([0-9]|[a-z]){2} 파일들은 Cache 데이터로, 프로필 사진이나 주고받은 사진, 동영상의 썸네일이 존재한다. data_0과 data_1~ data_3 파일을 통해 Cache 데이터의 파일 이름을 알 수 있다.

Ⅴ. 메신저 데이터의 디지털 포렌식 활용방안

본 장에서는 Slack과 Discord의 활용방안 예시로 가상 시나리오를 작성하고 이를 기반으로 각 메신저 데이터가 포렌식적으로 어떻게 활용될 수 있는가에 대해 제시한다.

국내 노트북 제조 회사인 D사에 재직하던 최 모씨(용의자)는 신제품 개발 프로젝트에 참여해 기술 개발 업무를 수행 중이었다. 회사에서 업무시간인 오전 10시부터 오후 6시에는 개인 스마트폰 사용을 금지했고 지급한 스마트폰만 사용하게 했다. 지급한 스마트폰에는 Slack과 Discord 메신저만 다운로드 되어있었다. 해당 프로젝트가 완성될 때 최 모씨는 갑작스럽게 사직서를 낸다. 그로부터 정확히 일주일 뒤, D사의 경

쟁사인 F사에서 동일한 사양의 신제품을 출시한다. D사의 기술 개발 팀장은 최 모씨가 퇴사하기 전, 신제품의 기술 유출을 의심하여 회사에서 지급한 최 모씨의 스마트폰에 대해 분석을 의뢰한다.

5-1 Slack을 이용한 증거 분석

데이터베이스의 messages 테이블은 그림 6과 같다. channel_id에는 CRV2Y29L3, CRWR794Z, DS6392MC4, 과 DSBCK7EQ7가 존재한다. CRV2Y29L3과 CRWR794Z 채널은 메시지 유형이 channel_join이므로 채널에 초대했을 경우 전송되는 기본 메시지이다. 기본 메시지 채팅창을 제외하면 실제 대화를 나눈 채널은 DS6392MC4과 DSBCK7EQ7이다.

local_id	ts	channel_id	client_msg_id	msg_send_state	...eral_msg	alls_room_...	thread_ts	message_blob	...ly_broadc	subtype
필터	필터	필터	필터	필터	필터	필터	필터	필터	필터	필터
4efd5b7b-...	157787732...	CRV2Y29L3	NULL	1	0	NULL	NULL	{"ephemeral...	NULL	channel_join
6d9f0f2a-5...	157787751...	CRV2Y29L3	NULL	1	0	NULL	NULL	{"ephemeral...	NULL	channel_join
f9a63867-...	157787732...	CRW9R794Z	NULL	1	0	NULL	NULL	{"ephemeral...	NULL	channel_join
5630a94e-...	157787751...	CRW9R794Z	NULL	1	0	NULL	NULL	{"ephemeral...	NULL	channel_join
f447a814-...	157788261...	DS6392MC4	517a7abd-3...	1	0	NULL	NULL	{"blocks":[{"...	NULL	NULL
35e9b1d9-...	157788261...	DS6392MC4	NULL	1	0	NULL	NULL	{"ephemeral...	NULL	NULL
e7c8975f-...	157787760...	DS8CK7EQ7	9e658b01-7f...	1	0	NULL	NULL	{"blocks":[{"...	NULL	NULL
a647f44a-...	157787764...	DS8CK7EQ7	0B2F5065-18...	1	0	NULL	NULL	{"client_msg...	NULL	NULL
d0949505-...	157787862...	DS8CK7EQ7	f05cd74-23a...	1	0	NULL	NULL	{"blocks":[{"...	NULL	NULL
73d0f441-...	157787867...	DS8CK7EQ7	2B4BBAAA-9...	1	0	NULL	NULL	{"channel":"...	NULL	NULL
16dbe847-...	157787867...	DS8CK7EQ7	F1B1084B-0F...	1	0	NULL	NULL	{"channel":"...	NULL	NULL
95d1e486-...	157787870...	DS8CK7EQ7	4883C69E-6...	1	0	NULL	NULL	{"channel":"...	NULL	NULL
accd06e1-...	157787863...	DS8CK7EQ7	310c46b2-97...	1	0	NULL	1577878639...	{"client_msg...	NULL	NULL
b1a1d37d-...	157787871...	DS8CK7EQ7	b0a55a5c-d...	1	0	NULL	NULL	{"blocks":[{"...	NULL	NULL

그림 6. messages 테이블
Fig. 6. messages Table

messaging_channels 테이블에는 그림 7과 같이 채널의 정보가 존재한다. DS6392MC4 채널 ID는 SLACKBOT이므로 슬랙에 존재하는 채팅 로봇과 대화한 채널이다. 따라서 실제 상대방과 대화를 나눈 채널은 DSBCK7EQ7이다.

msg_channel_id	name_or_user	name_or_user_normalized	type	is_starred	is_open	is_member	needs_msg_sync	msg_channel_blob
필터	필터	필터	필터	필터	필터	필터	필터	필터
CRV2Y29L3	general	general	0	0	1	1	0	{"id":"CRV2Y29L3",...
CRW9R794Z	random	random	0	0	1	1	1	{"id":"CRW9R794Z",...
DRW9R77FB	URV2Y25FD	URV2Y25FD	2	0	1	NULL	1	{"id":"DRW9R77FB",...
DS6392MC4	USLACKBOT	USLACKBOT	2	0	1	NULL	1	{"id":"DS6392MC4",...
DSBCK7EQ7	URV2Y46Q3	URV2Y46Q3	2	0	1	NULL	1	{"id":"DS8CK7EQ7",...

그림 7. messaging_channels 테이블
Fig. 7. messaging_channels Table

메시지 내용은 message 테이블의 message_blob 컬럼에 저장되어 있다. DSBCK7EQ7 채널에서 ID가 URV2Y25FD와 URV2Y46Q3인 사용자가 주고받은 메시지는 그림 8과 같다. 이때 ts에 저장된 Unix Time은 한국 표준시인 UTC+9를 적용하여 변환한다.

그림 8. message 테이블 내의 message_blob 컬럼과 시간 정보
Fig. 8. message_blob Column and time in message Table

users 테이블은 그림 9와 같이 사용자의 ID, 이름, 프로필에서 설정한 이름을 포함한다. URV2Y25FD는 프로필 이름이 D_Mr.CHOI이므로 D사 최 모씨의 ID이고 URV2Y46Q3는 프로필 이름이 F_Mr.KIM이므로 F사 김 모씨의 ID이다.

id	name
필터	필터
URV2Y25FD	
URV2Y46Q3	
USLACKBOT	slackbot

profile_first_name	profile_last_name
필터	필터
D_Mr.	CHOI
F_Mr.KIM	
slackbot	

그림 9. users 테이블
Fig. 9. users Table

message 테이블, messaging_channels 테이블과 users 테이블을 통해 획득한 데이터들을 연관시키면 그림 10과 같다. messages 테이블의 channel_id 컬럼과 messaging_channels 테이블의 msg_channel_id 컬럼을 매칭시켜 name_or_user 컬럼에 저장된 사용자의 ID를 알 수 있다. 사용자의 실제 이름은 messaging_channels 테이블의 name_or_user 컬럼과 user 테이블의 id 컬럼을 매칭시켜 알 수 있다.

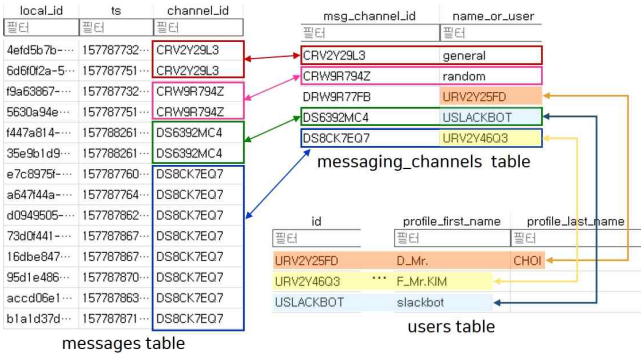

그림 10. message 테이블, messaging_channels 테이블과 users 테이블에서 획득한 데이터
Fig. 10. Data in message Table, messaging_channels Table and users Table

최종적으로 데이터베이스를 통해 획득한 증거로 타임라인을 구성하면 그림 11과 같다. 용의자인 최 모씨는 업무시간이 아닌 2020년 1월 1일 오후 8시 37분 19초에 비밀문서인 'new product technology.hwp' 파일을 F사의 김 모씨에게 전송했으므로 기밀 유출 사건의 범인은 최 모씨이다.

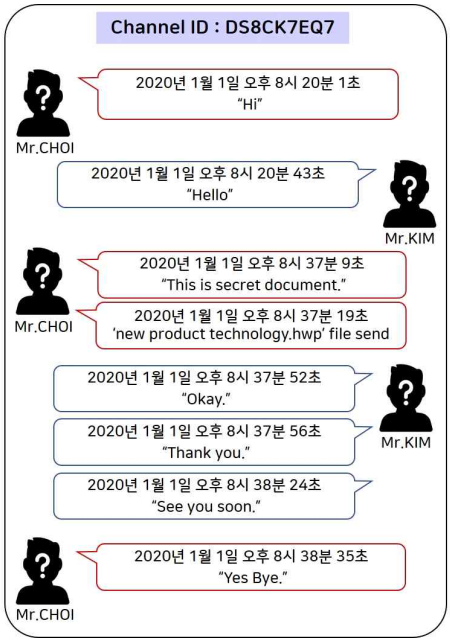

그림 11. Slack에서 획득한 증거로 재구성한 대화 내용
Fig. 11. Dialogue with evidence acquired in Slack

최 모씨가 공유한 'new product technology.hwp' 파일은 'com.Slack₩cache₩file-upload' 경로에서 그림 12와 같이 원본 파일을 획득할 수 있다. 또한, 데이터베이스의 files 테이블에서도 파일에 관한 URL이 존재하지만, 파일을 획득하기 위해선 해당 Workspace에 로그인해야 한다.

그림 12. file-upload에 저장된 원본 파일
Fig. 12. Original file in file-upload directory

5-2 Discord를 이용한 증거 분석

STORE_MESSAGE_CACHE_V* 파일에 메시지 내용이 그림 13과 같이 저장되어있다. 메시지의 수신자 정보가 저장되지 않지만, 시간과 내용을 통해 Mr.PARK과 Mr.CHOI가 메시지를 주고받은 사실을 알 수 있다. 메시지 내용의 맨 끝 문자는 알 수 없는 문자로 표현되어 있지만, 헥사 값에 0x80을 빼서 정확한 문자를 획득하고, UTC+0으로 저장된 메시지 발신 시간은 한국 표준시를 고려해 UTC+9를 적용해 변환한다. Mr.CHOI가 공유한 'new_product_technology.hwp' 파일을 다운로드 할 수 있는 URL은 문서 전송 시 저장되는 두 개의 URL 중에 두 번째 URL이다.

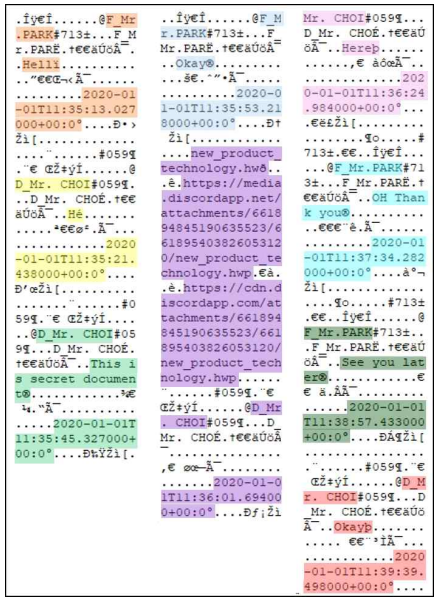

그림 13. STORE_MESSAGE_CACHE_V* 파일
Fig. 13. STORE_MESSAGE_CACHE_V* file

획득한 증거를 통해 타임라인을 구성하면 그림 14와 같다. 용의자인 최 모씨가 업무시간이 아닌 2020년 1월 1일 오후 8시 35분 45초에 F사의 박 모씨에게 비밀문서인 'new product technology.hwp' 파일을 공유한 사실이 확인되므로 신제품 기술 유출 사건의 범인은 최 모씨이다.

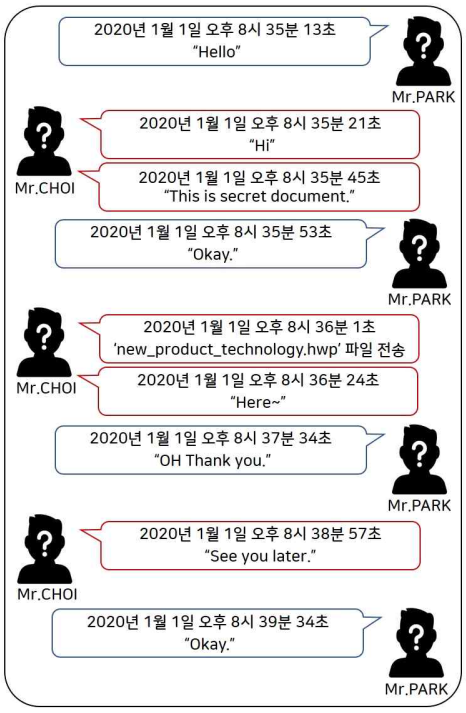

그림 14. Discord에서 획득한 증거로 재구성한 대화 내용
Fig. 14. Dialogue with evidence acquired in Discord

Ⅵ. 결 론

현대인들이 필수적으로 사용하는 메신저는 대화 내용뿐만 아니라 계정 정보 등과 같은 개인정보도 기록된다. 이러한 메신저들은 사용자의 행위 정보를 대부분 저장하고 있다. 그러나 메신저는 기밀유출 등 일부 범죄 활동에 악용되는 사례가 존재하기 때문에 주요 메신저들에 대해 각 메신저가 저장하고 있는 데이터의 내용, 범위 및 특징을 파악해 분석하는 것이 필요하다.

본 논문에서는 팀 협업을 목적으로 제작된 메신저인 Slack과 Discord를 대상으로 분석을 진행하였으며, 각 메신저에 대한 아티팩트 분석을 수행하였다. 주요 데이터가 저장되는 경로와 파일을 분류하고 해당 데이터에 대한 의미를 파악하였으며, 시나리오를 통해 디지털 포렌식 관점에서의 데이터 활용방안을 제시하였다. 본 논문의 내용을 기반으로 메신저 데이터가 디지털 포렌식 수사 시 효율적으로 활용될 것을 기대한다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1F1A1060634).

참고문헌

[1] National Information Society Agency, 2019 Survey on smart phone dependency [Internet]. Available: https://msit.go.kr/cms/www/policyCom/report/__icsFiles/afieldfile/2020/02/20/(%EB%B6%99%EC%9E%84)%202019%EB%85%84%20%EC%8A%A4%EB%A7%88%ED%8A%B8%ED%8F%B0%20%EA%B3%BC%EC%9D%98%EC%A1%B4%20%EC%8B%A4%ED%83%9C%EC%A1%B0%EC%82%AC%20%EA%B2%B0%EA%B3%BC.pdf

[2] Joongang Article. Decision Proof from Telegram Forensic Recovery Success [Internet]. Available: https://news.joins.com/article/23338922

[3] Sports Donga. Memo Definitive Evidence [Internet]. Available: https://sports.donga.com/sports/article/all/20190207/94003145/2

[4] International Business Times. 3 Students Charged In Alleged Plot To Attack School In Upstate New York [Internet]. Available: https://www.ibtimes.com/3-students-charged-alleged-plot-attack-school-upstate-new-york-2862645

[5] Gao Feng, Zhang Ying, "Analysis of WeChat on iPhone," *2nd International Symposium on Computer, Communication,Control and Automation, Atlantis Press,* April, 2013.

[6] Aditya Mahajan, Dahiya Ms, Sanghvi, H. P. "Forensic Analysis of Instant Messenger Applications on Android Devices," *International Journal of Computer Applications,* Vol No pp. April, 2013.

[7] Seunghee Seo, Gihoon Nam, Yeog Kim, Changhoon Lee. "Artifacts Anaylsis of User Behavior in Korea Random Chat Application," J*ournal of Digital Forensics,* Vol. 12, No. 3, pp. 1-7, 2018,

[8] Seunghee Seo, Yeong Kim, Changhoon Lee. "Instagram Users Behavior Analysis in a Digital Forensic Perspective," *Journal of the Korea Institute of Information Security&Cryptology*, Vol. 28, No. 2, pp. 407-416, 2018.

[9] Cosimo Anglano, Massimo Canonico, Marco Guazzone, "Forensic analysis of the ChatSecure instant messaging application on android smartphones", *Digital Investigation*, Vol. 19, pp. 44-59, Dec, 2016.

[10] Giyoon Kim, Uk Hur, Sehoon Lee, Jonsung Kim. "Forensic Analysis of the Secure Instant Messenger Surespot," J*ournal of Digital Forensics*, Vol. 13, No. 3, pp. 176-188, 2019.

[11] Joongang Article. Business App [Internet]. Available: https://news.joins.com/article/23569931

[12] Github. Chrome Cache file format.asciidoc [Internet]. Available: https://github.com/libyal/dtformats/blob/master/documentation/Chrome%20Cache%20file%20format.asciidoc

신수민(Sumin Shin)

2020년 2월 : 국민대학교 정보보안암호수학과 졸업

2020년 3월~현 재 : 국민대학교 금융정보보안학과 석사과정
※관심분야 : 디지털 포렌식, 정보보호

박은후(Eunhu Park)

2018년 7월 : 국민대학교 정보보안암호수학과 졸업

2018년 8월~현 재 : 국민대학교 금융정보보안학과 석사과정
※관심분야 : 디지털 포렌식, 정보보호

김소람(Soram Kim)

2016년 2월 : 국민대학교 수학과 졸업
2018년 2월 : 국민대학교 금융정보보안학과 석사

2018년 3월~현 재 : 국민대학교 금융정보보안학과 박사과정
※관심분야 : 디지털 포렌식, 정보보호

김종성(Jongsung Kim)

2000년 8월/2002년 8월 : 고려대학교 수학 학사/이학석사
2006년 11월 : K.U.Leuven, ESAT/SCD-COSIC 정보보호 공학박사
2007년 2월 : 고려대학교 정보보호대학원 공학박사
2007년 3월 ~ 2009년 8월 : 고려대학교 정보보호기술연구센터 연구교수
2009년 9월 ~ 2013년 2월 : 경남대학교 e-비즈니스학과 조교수
2013년 3월 ~ 2017년 2월 : 국민대학교 수학과 부교수

2014년 3월 ~ 현재 : 국민대학교 일반대학원 금융정보보안학과 부교수
2017년 3월 ~ 현재 : 국민대학교 정보보안암호수학과 부교수
※관심분야 : 정보보호, 암호 알고리즘, 디지털 포렌식