https://doi.org/10.33778/kcsa.2023.23.3.013

안드로이드 환경에서 Signal과 Telegram 보안 메신저 디지털 포렌식 분석 연구★

권 재 민*, 박 원 형**, 최 윤 성***

요 약

본 연구는 안드로이드 환경에서 널리 사용되는 두 개의 보안 메신저인 Signal과 Telegram에 대한 디지털 포렌식 분석을 진행하였다. 현재 모바일 메신저가 일상생활의 중요한 역할을 하는 만큼, 이들 앱 내부의 데이터 관리와 보안성은 매우 중요한 이슈가 됐다. Signal과 Telegram은 그중에서도 사용자들 사이에서 높은 신뢰성을 받고 있는 보안 메신저로, 암호화 기술을 바탕으로 사용자들의 개인 정보를 안전하게 보호한다. 하지만 이러한 암호화된 데이터를 어떻게 분석할 수 있을지에 대해서는 아직까지 많은 연구가 필요하다. 본 연구에서는 이러한 문제점을 해결하기 위해 Signal과 Telegram의 메시지 암호화와 안드로이드 디바이스 내 데이터베이스 구조 및 암호화 방식에 대하여 깊이 있는 분석을 진행하였다. Signal의 경우, 복잡한 알고리즘으로 인해 외부에서 접근하기 어려운 암호화된 메시지를 성공적으로 복호화 하여 내용을 확인할 수 있었다. 또한 두 메신저 앱의 데이터베이스 구조를 세밀하게 분석하여 해당 정보를 수시로 활용할 수 있는 폴더 구조 및 파일 형태로 정리하는 작업도 진행했다. 이렇게 분석한 정보를 바탕으로 보다 발전된 기술과 방법론을 적용함으로써, 앞으로 더욱 정확하고 세밀한 디지털 포렌식 분석이 가능할 것으로 기대된다. 이 연구가 Signal과 Telegram 같은 보안 메신저에 대한 이해를 높이는 데 도움을 주며, 이로 인해 개인 정보 보호와 범죄 예방 등 여러 측면에서의 활용 가능성이 열릴 것으로 예상된다.

Signal and Telegram Security Messenger Digital Forensic Analysis study in Android Environment

Jae-Min Kwon*, Won-Hyung Park**, Youn-sung Choi***

ABSTRACT

This study conducted a digital forensic analysis of Signal and Telegram, two secure messengers widely used in the An droid environment. As mobile messengers currently play an important role in daily life, data management and security wit hin these apps have become very important issues. Signal and Telegram, among others, are secure messengers that are hi ghly reliable among users, and they safely protect users' personal information based on encryption technology. However, much research is still needed on how to analyze these encrypted data. In order to solve these problems, in this study, an in-depth analysis was conducted on the message encryption of Signal and Telegram and the database structure and encry ption method in Android devices. In the case of Signal, we were able to successfully decrypt encrypted messages that are difficult to access from the outside due to complex algorithms and confirm the contents. In addition, the database structure of the two messenger apps was analyzed in detail and the information was organized into a folder structure and file forma t that could be used at any time. It is expected that more accurate and detailed digital forensic analysis will be possible in the future by applying more advanced technology and methodology based on the analyzed information. It is expected that this research will help increase understanding of secure messengers such as Signal and Telegram, which will open up pos sibilities for use in various aspects such as personal information protection and crime prevention.

Key words : Digital Forensic Analysis, Data Encryption, Data Decryption, Database Structure

접수일(2023년 08월 28일), 수정일(2023년 09월 05일), 게재확정일(2023년 09월 11일)

★ 이 논문은 2023년도 융합보안학회 하계학술대회에서 수상한 논문을 수정·보완한 논문이며, 본 연구는 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다.(2021RIS-003)

* 인제대학교 컴퓨터공학부 학사과정 (주저자)

** 성신여자대학교 융합보안공학과 부교수

*** 인제대학교 AI빅데이터학부 조교수 (교신저자)

1. 서 론

최근 안드로이드 기반의 메신저 애플리케이션 중에서, 특히 암호화 통신을 지원하는 Signal과 Telegram이 사용자들 사이에서 두각을 나타내고 있다. 이러한 메신저 애플리케이션들은 사용자의 개인 정보와 메시지를 안전하게 보호하기 위해 강력한 암호화 기술을 적용하고 있으며, 이는 개인정보 보호에 크게 기여하고 있다.

그러나, 어떤 기술이든 그 자체는 중립적일 뿐, 그것을 어떻게 사용하는가에 따라 긍정적 혹은 부정적 결과를 초래할 수 있다. 이러한 강력한 암호화 기능도 마찬가지로 범죄에 악용될 수 있는 가능성을 태포하고 있다. 2020년 한국을 충격에 빠트린 N 번 방 사건과 같은 대형 사건부터 시작하여 최근 화두가 되고 있는 마약 및 다른 불법적인 물품 거래까지, 메신저 애플리케이션은 범죄나 불법적인 행위를 계획하고 조작하는 도구로서도 사용되는 것으로 나타났다.

따라서 이러한 메신저 애플리케이션의 데이터를 수사 대상으로 분석하는 경우, 디지털 포렌식 분석 기법을 통해 해당 애플리케이션에서 생성된 데이터를 복호화하고 분석하는 고도의 전문성과 세밀함이 요구된다.

본 연구에서는 안드로이드 환경에서 널리 사용되는 메신저 애플리케이션 중 Signal 과 Telegram의 데이터를 디지털 포렌식 방법론으로 체계적으로 분석한다. 이 과정에서 우리는 각각의 데이터베이스 구조와 저장 방식 등 다양한 측면들을 깊게 파악하여 그 결과로부터 수사관들에게 유용할 수 있는 디지털 증거들을 도출하였다. 이러한 연구 결과는 현장 수사에서 실질적으로 활용될 수 있을 뿐만 아니라, 디지털 포렌식 분야의 발전에도 기여할 것으로 기대된다. 특히, 이번 연구를 통해 파악된 데이터베이스 구조와 저장 방식 등의 정보는 암호화된 메신저 애플리케이션의 데이터를 보다 효과적으로 분석하는 데 도움을 줄 것으로 예상된다.

2장에서는 관련 연구를 소개하고 3장에서는 Signal의 암호화된 정보를 복호화 하는 방안을 소개하고 Signal에서 디지털 포렌식 관점에서 유의미한 아티팩트를 정리한다. 4장에서는 Telegram에서 디지털 포렌식 관점에서 유의미한 아티팩트를 정리한다. 마지막으로 5장에서는 결론으로 마무리한다.

2. 관련 연구

본 장에서는 모바일 포렌식 선행 연구 및 데이터베이스 보안, 종단 간 암호화를 포함하는 기본 개념에 대해 설명한다.

2.1 모바일 메신저 포렌식 선행 연구

보안 메신저 앱에서 암호화된 데이터를 복호화하는 방법에 대한 연구 결과들이 있다. Jihun Son 외 3인은 보안 메신저 앱 Signal, Wickr, Threema에 대해서 역공학을 통해 이러한 앱들의 암호화 메커니즘을 분석하여, 암호화된 정보를 해독하거나 앱 설정 파일에 저장된 암호화 키를 추출하여 복구하는 방법을 제시했다[1].

또한 조재민 등은 인스턴트 메신저 앱인 Element에 대해서 아티팩트 분석을 한 결과들이 있다. Element 보안 메신저는 대화 참여자만 대화 이력을 확인할 수 있도록 종단 간 암호화 기능을 제공하고 있으나 이를 복호화 하는 연구가 미흡하다. 사용자의 패스워드 없이 Windows 자격 증명 관리자 서비스에 저장된 복호화 키를 활용하여 암호화된 보안 채팅방의 이력을 평문으로 확인하는 방안을 제안했다[2].

Zhang Hao 등은 인스턴트 메신저 디지털 포렌식 연구를 위해 디바이스의 내부 메모리에서 어떤 데이터와 정보를 찾을 수 있는지 확인하고 기기의 파일 시스템에서 추출한 위치 정보도 확인하였다. 그리고 온라인 채팅 데이터 스트림과 Line, 안드로이드 메신저, 행아웃, WhatsApp의 로컬 데이터베이스의 아티팩트에 대해서 암호화 방법에 대해서 조사하였다[3].

최주섭 등은 한국과 중국에서 가장 많이 사용되는 3개의 인스턴트 메신저인 카카오톡, 네이트온, QQ에 대해서 분석을 진행하여 개인 데이터 파일

의 위치와 파일 형식을 분석하였다. 그리고 리버스 엔지니어링을 통해 이러한 메시징 응용 프로그램의 내부 데이터베이스에 대한 암호화 및 암호 해독 절차에 대해서 제안했다[4].

2.2 데이터베이스 보안

Signal 애플리케이션의 데이터베이스 같은 경우 데이터베이스 보안을 위해 데이터베이스에 대한 암호화를 걸어서 다른 사용자가 손쉽게 확인할 수 없게 한다. 이러한 애플리케이션의 데이터베이스는 개인정보를 저장하고 보호하는 데 중요한 역할을 한다. 다음은 데이터베이스 암호화 방식이다.

- Plug-in 방식 : 응용 프로그램의 수정 없이 쉽게 암호화가 적용되어 있으며, 대용량 데이터인 경우 암/복호화 과정이 DBMS에서 수행되므로 DBMS에 부하 발생된다[5].
- API 방식 : 대용량 데이터의 암/복호화 과정이 프로그램 단에서 수행되므로 DBMS의 부하가 분산 효과 된다. 그리고 DBMS 부하가 적어 속도 증가 효과 발생으로 시스템 영향도가 낮다. 데이터베이스의 성능을 저하시키지 않으나 애플리케이션의 전체적인 수정이 필요하다[5].
- Kernel 방식 : DBMS의 엔진 레벨에서 직접 암/복호화를 수행하기 때문에 다른 어떤 방식보다 구축이 간편하며, 빠른 암/복호화 성능을 제공한다. 다만, 엔진 레벨의 수정이 필요하며 DBMS 벤더들이 직접 제공을 하거나 MySQL/Oracle와 같은 DBMS 또는 보안 전문 업체와 데이터베이스 업체의 협력이 이루어진 DBMS에서만 지원이 가능하다[6].
- Appliance 방식 : 암호화를 위한 전용 장비를 사용함으로써 처리 성능이 우수하나 도입 비용이 크다.

2.3 종단 간 암호화 구성요소

종단 간 암호화는 보안 메신저인 Signal과 Telegram 애플리케이션에서 다른 제3자가 데이터를 가로채도 확인할 수 없게 한다. 이러한 종단 간 암호화 때문에 로컬에서의 애플리케이션 데이터를 분석하여 아티팩트를 수집하여야 한다. 이 종단 간 암호화는 발신자와 수신자만 메시지의 내용을 읽을 수 있는 보안 통신 프로세스이다. 메시지를 보내기 전에 보낸 사람의 장치에서 메시지를 암호화한 다음 메시지를 받은 후 받는 사람의 장치에서 암호를 해독하게 된다. 다음은 종단 간 암호화에 필요한 구성 요소이다.

- 키 교환 : 두 장치 간에 통신할 때 데이터를 보호하기 위해 사용되는 공유 비밀 키에 동의해야 한다. 종단 간 암호화에 사용되는 키는 주로 대칭 키와 비대칭 키 두 가지로 나뉜다. 대칭 키는 암호화와 복호화에 동일한 키가 사용되며, 송신자와 수신자의 동일한 키를 공유하여 데이터를 안전하게 전송한다. 비대칭 키는 공개 키와 개인 키라는 두 가지 다른 키 쌍을 사용한다. 공개 키는 누구나 알 수 있지만, 개인 키는 엄격하게 비밀로 보호된다. 이렇게 함으로써 데이터를 안전하게 전송하고 서명하는 프로세스를 보다 안전하게 유지할 수 있다.
- 암호화 : 키 교환이 완료되면 발신자는 공유 비밀 키를 활용하여 데이터를 암호화할 수 있게 된다. 암호화 알고리즘은 데이터를 혼란스럽게 만들어 가지지 않은 이에게는 해독이 불가능하게 한다. 종단 간 암호화에서 데이터는 송신자의 디바이스에서 암호화된 후 수신자에게 전송되기 전까지 안전하게 보호된다.
- 암호 해독 : 암호화된 데이터가 수신자의 장치에 도착하면, 그 데이터를 해독하려면 오로지 공유 비밀 키만을 활용할 수 있다. 수신자의 장치는 이 키를 활용하여 데이터를 해독하고, 그럼으로써 데이터를 읽을 수 있게 된다. 종단 간 암호화는 오직 의도된 수신자만이 이 키에 접근할 수 있기 때문에 데이터가 안전하게 유지된다[7].

3. Signal 아티팩트 분석

Signal의 데이터를 분석하기 위해서는 안드로이드

핸드폰을 루팅 하여 최고 관리자 권한인 루트를 얻는 과정이 선행되어야 한다. 그리고 안드로이드 디버그 브리지를 이용하여 안드로이드 이미징을 한다. 루트 권한을 얻어야 안드로이드의 data/data 파일에 접근이 가능하다[8].

3.1 Signal 데이터베이스 복호화 방안

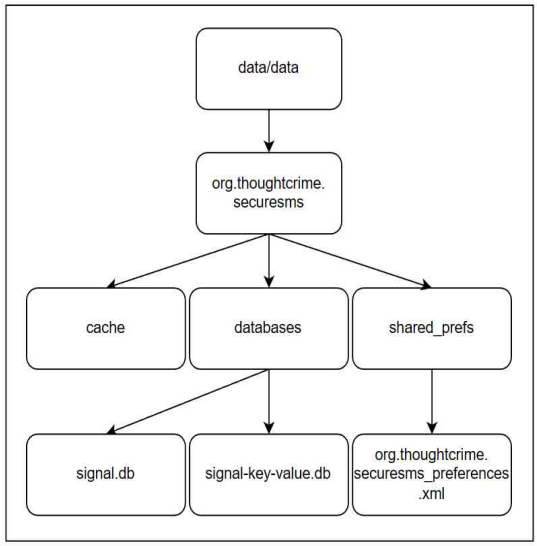
Signal 애플리케이션은 signal.db라는 파일의 메시지, 대화방 및 연락처와 관련된 데이터를 저장하는 SQLite 데이터베이스 파일을 가지고 있다. 이 데이터베이스는 암호화되어 있기 때문에 복호화를 하지 않으면 데이터베이스에 접근을 할 수 없다[9]. 안드로이드 5.1.1 버전에서는 org.thoughtcrime.securesms_preferences.xml 파일에는 아래와 그림같이 pref_database_unencrypted_secret으로 key 값이 저장되어 있는 것을 확인할 수 있다.

(그림 1) 안드로이드 5.1.1 버전의 org.thoughtcrime.securesms_preferences.xml 파일

이 key 값을 가지고 데이터베이스를 SQLCipher을 이용하여 페이지 크기를 4,096로 설정하고 HMAC 알고리즘과 KDF 알고리즘을 SHA1알고리즘을 선택하여 복호화를 진행하면 데이터베이스를 열어서 확인할 수 있다. 그리고 안드로이드 버전 7버전 이상부터는 Android Keystore를 도입하여 권한이 없으면 key 값을 추출할 수 없도록 하였다. 그래서 안드로이드 버전 7버전 이상에서는 먼저 data/keystore/user_0 경로에 있는 “app-id”_USERKEY_SignalSecret 파일에 오프셋 2D에서 3C까지 값을 저장해 준다. 그리고 안드로이드 7버전 이상의 org.thoughtcrime.securesms_preferences.xml 파일에는 pref_database_encrypted_secret라고 data와 iv 값이 저장되어 있다. 이 데이터 값을 Base64 to Hex로 변환하여 뒤에서 32자리는 Auth tag가 되고 나머지 값은 데이터 값이 된다. 그리고 AES Decrypt를 진행하여 데이터 값을 넣어주고 key 값에는 아까 저장해 준 오프셋 2D에서 3D까지 값을 넣어주고 GCM Tag에는 Auth Tag를 넣고 IV 값을 넣어서 복호화를 진행하게 되면 안드로이드 5.1.1버전의 prev_database_unencrypted_secret 값과 같은 값을 가지게 된다.

3.2 Signal의 포렌식 관점에서 유의미한 아티팩트

Signal이라는 애플리케이션을 분석하기 위해서 확인해야 하는 디렉터리 구조는 아래 그림 2와 같다.



(그림 2) Signal 주요 아티팩트 위치

Signal의 주요 아티팩트는 data/data 디렉터리 밑에 org.thoughtcrime.securesms 폴더안에 있다. 먼저 shared_prefs 디렉터리는 Signal의 설정 정보와 같은 앱 데이터가 저장되는 디렉터리이다. 이 디렉터리의 설정 파일은 XML 형식으로 작성되어 있다. 이 디렉터리 안에 있는 org.thoughtcrime.securesms_preferences.xml 파일은 복호화에 사용되는 secret 키

가 저장되어 있다. 그리고 databases 디렉터리 안에 있는 signal.db 파일은 분석한 결과 사용자가 애플리케이션을 사용한 흔적을 확인할 수 있었고, 사용자의 전화 기록과 보낸 메시지의 내용과 보낸 시간을 확인할 수 있었다. 특히 signal-key-value.db에서는 사용자의 휴대전화 번호가 남아있는 것을 확인할 수 있고 새로운 그룹방에 들어간 횟수와 사용자가 친구를 초대한 횟수도 확인할 수 있다. 다음은 signal.db와 signal-key-value.db에서 얻을 수 있는 정보를 나열한 표이다.

<표 1> signal.db에서 추출 가능한 정보

Specific Path	File type	Table	Feature	Forensic Artifact
\databases\signal.db	DB	groups	group_id	A unique number set for group chats in the user's app
			title	Chat room name in the group chat room
			timestamp	Time the user entered the group chat room
		call	timestamp	time the user made a call
		message	date_sent	time the message was sent
			thread_id	A unique number set for chat in the user's app
			body	Conversations in the chat room
		story_sends	sent_timestamp	when i posted my story
		thread	snippet	The most recent conversation in each chat room
			last_seen	Last time the chat room was viewed

<표 2> signal-key-value.db에서 추출 가능한 정보

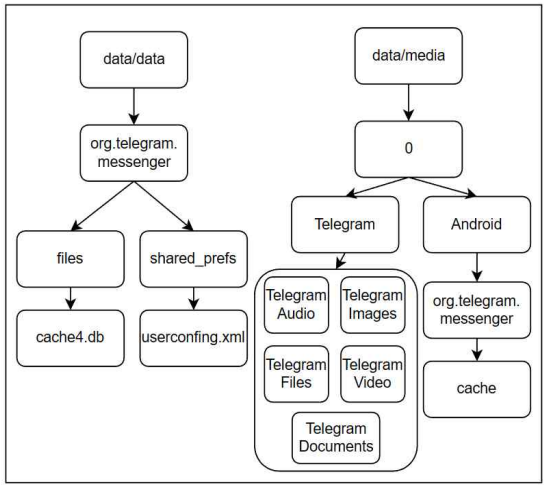
Specific Path	File type	Table	Feature	Forensic Artifact
\databases\signal-key-value.db	DB	key_value	account.e164	User's mobile phone number
			onboarding.new_group	The number of times you entered a new group room
			onboarding.invite_friends	Number of times a user invited a friend
			user.has.added.to.a.story	The number of times users have added stories
			storage.last_sync_time	Storage Last Sync Time

4. Telegram 아티팩트 분석

안드로이드 핸드폰에서 Telegram을 분석하기 위해서는 위 Signal을 분석하기 위해 준비했던 과정이랑 동일하다. 안드로이드 핸드폰이 루팅이 되어있는 상태에서 이미징 도구인 안드로이드 디버그 브리지를 이용하여 이미징을 하였다.

4.1 Telegram의 포렌식 관점에서 유의미한 아티팩트

Telegram의 데이터베이스는 files 디렉터리에 저장되어 있으며 사용자의 연락처 목록, 음성 통화 로그 등을 SQLite 데이터베이스 형식으로 저장되어 있다. 그리고 shared_prefs 디렉터리의 userconfig.xml 파일로 Telegram 계정의 세부 정보가 저장되어 있다. 그리고 data/media 디렉터리 안에는 Telegram에서 생성된 사용자의 캐시 파일과 Telegram 메신저를 통해 주고받은 파일의 사본이 저장되는 것을 확인할 수 있다. 아래는 Telegram 애플리케이션을 분석할 때 확인할 수 있는 주요 아티팩트들의 위치이다.



(그림 3) Telegram의 주요 아티팩트 위치

위의 그림 3을 확인하여 보면 org.telegram.messenger 폴더 밑에 shared_prefs 폴더가 있는 것을 확인할 수 있는데 거기에는 userconfig.xml 파일이 있는 것을 확인할 수 있다. Signal의 userconfig.xml 파일에는 킷값이 저장되어 있는 것을 확인할 수 있었는데 안드로이드 5.1.1 버전의 Telegram 애플리케이션 구동 환경에서는 데이터베이스를 암호화하지 않아 키값이 들어가지 않은 것을 확인할 수 있다.

(그림 4) 안드로이드 5.1.1 버전의
Telegram userconfing.xml 파일

위 그림 4와 같이 userconfig.xml 파일을 확인하여 보면 user의 값이 base64방식으로 인코딩되어 저장되어 있는 것을 확인할 수 있다. 이러한 인코딩되어 있는 데이터 값을 base64로 디코딩 하여 확인하여 보면 해당 애플리케이션을 사용하는 유저의 이름과 사용자의 기기 전화번호가 저장되어 있는 것을 확인할 수 있다.

Telegram 애플리케이션은 Signal 애플리케이션과는 다르게 데이터베이스에 대한 암호화가 되어있지 않다. 그리고 테이블 안의 data 칼럼이 blob 칼럼이 blob 데이터 형식으로 저장되어 있는 것을 확인할 수 있다[10]. blob 데이터는 UTF-8로 디코딩 하여 확인하면 평문 데이터로 확인할 수 있다. cache4.db를 확인하여 분석하여 아래 표 3와 같이 정보가 저장되어 있는 것을 확인할 수 있다.

<표 3> cache4.db에서 추출 가능한 정보

Specific Path	File type	Table	Feature	Forensic Artifact
\files \cache4.db	DB	contacts	uid	TID of the user's contact (the unique number set for the contact's user in the record)
		chats	uid	TID of the user's chat room (a unique number set for the user's chat room in the record)
		chats	name	The name of the chat room (null if private)
		users	uid	User's friend's TID (the unique number set for the contact's friend in the record)
		users	name	User's friend name
		users	data	User's name and phone number
		messages_v2	date	Creation time of conversation content
		messages_v2	data	User conversation history

5. 결 론

현재의 모바일 메신저 애플리케이션은 우리의 일상생활에 깊숙이 뿌리를 내린 도구로, 소통과 정보 공유를 용이하게 만들어 주는 역할을 한다. 그러나 이러한 편리함은 악의적인 용도로도 활용될 수 있음을 인식해야 한다. 이러한 가능성은 모바일 메신저 애플리케이션을 통해 범죄행위를 조사하고 예방하는 중요성을 강조하고 있다.

대표적인 암호화 메신저 애플리케이션인 Signal과 Telegram을 안드로이드 환경에서 분석함으로써, 주요 아티팩트를 분석하여 데이터베이스의 구조를 파악하고, 주요 아티팩트의 디렉터리를 경로별로 정리했다. 특히 Signal 데이터베이스의 복호화 방법을 통해 암호화된 메시지를 복호화 하여 추출하고, 사용자의 활동 흔적을 분석함으로써 수사관이 필요로 하는 증거를 확인했다. 이러한 연구 결과는 모바일 포렌식 분석 분야에서 유용하게 활용될 뿐 만 아니라, 불법 활동을 조사하고 수사하는데 필요한 중요한 정보를 제공하는 기술적 기반 자료로 활용될 것으로 기대된다.

참고문헌

[1] 손지훈, 김영웅, 오동빈, 김경곤, "Forensic analysis of instant messengers: Decrypt Signal, Wickr, and Threema", Forensic Science International: Digital Investigation, Vol.40, 2022.

[2] 조재민, 변현수, 윤희서, 서승희, 이창훈, "포렌식 관점에서의 Element 인스턴트 메신저 아티팩트 분석", 정보보호학회논문지 제32권 제6호, pp.1113-1120, 한국정보보호학회, 2022.

[3] Zhang, Hao, Lei Chen, and Qingzhong Liu,. "Digital forensic analysis of instant messaging applications on android smartphones" 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2018.

[4] 최주섭, 유재관, 현상원, 김형식, “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.50-59, 2019.

[5] 이병엽, 임종태, 유재수, "데이터베이스 암호화 솔류션 구현 및 도입을 위한 기술적 아키텍처", 한국콘텐츠학회논문지, 제14호제6호, pp.1-10, 2014.

[6] 박형도, “Is your database system secure?“, ORACLE KOREA MAGAZINE, pp.33-37, 2011.

[7] 김정윤 외5인, "종단 간 암호화 통신을 위한 키 전달 프로토콜에 관한 연구" 동계학술발표대회 논문집, 제16권 제2호, pp 391-394, 한국컴퓨터정보학회, 2008.

[8] 오정훈, 이상진, "안드로이드 스마트폰 포렌식 분석 방법에 관한 연구" 한국디지털포렌식학회, Vol.6, No.1, pp.47-76, 2012.

[9] 조석준, 최형기, "Mobile Instant Messenger인 Signal에 대한 forensic 분석기법". 한국통신학회 학술대회논문집, pp 456-458, 2020.

[10] Cosimo, A., Massimo, C., Macro, G. : "Forensic analysis of Telegram Messenger on Android smartphones" Digital Investigation, 2017, vol.23,pp 31-49, 2017.

〔저자소개〕

권재민 (Jae-Min Kwon)
2018년 3월~현재
인제대학교 컴퓨터공학부
정보보호 전공 (학사과정)
email : gun2475@gmail.com

박 원 형 (Won-Hyung Park)
2002년 서울과학기술대학교 산업정보시스템공학과 공학사
2006년 서울과학기술대학교 정보산업공학과 공학석사
2009년 경기대학교 정보보호학과 이학박사
2015년 성균관대학교 컴퓨터교육학과 박사수료
2020년 ~ 2022년 상명대학교 정보보안공학과 부교수
2023년 3월 ~ 현재 성신여자대학교 융합보안공학과 부교수
email : whpark@sungshin.ac.kr

최 윤 성 (Youn-sung Choi)
2006년 2월 성균관대학교 정보통신공학부 학사
2007년 8월 성균관대학교 전자전기컴퓨터공학부 석사
2015년 8월 성균관대학교 전자전기컴퓨터공학부 박사
2016년 3월 ~ 2020년 2월 호원대학교 사이버보안학과 조교수
2020년 3월 ~ 현재 인제대학교 AI융합대학 AI빅데이터학부 조교수
email : cys2020@inje.ac.kr