Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.6, Dec. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.6.1113

포렌식 관점에서의 Element 인스턴트 메신저 아티팩트 분석*

조 재 민,[1†] 변 현 수,[2] 윤 희 서,[2] 서 승 희,[3] 이 창 훈[4‡]
[1]고려대학교 (대학원생), [2,3,4]서울과학기술대학교 (학생, 대학원생, 교수)

Forensic Analysis of Element Instant Messenger Artifacts*

Jae-min Cho,[1†] Hyeon-su Byun,[2] Hui-seo Yun,[2] Seung-hee Seo,[3] Chang-hoon Lee[4‡]
[1]Korea University (Graduate student), [2,3,4]Seoul National University of Science And Technology (Undergraduate student, Graduate student, Professor)

요 약

최근 개인정보보호를 목적으로 데이터를 암호화해 저장하고 보안에 초점을 맞춰 종단 간 암호화 등의 서비스를 제공하는 메신저들이 등장하면서 수사에 어려움을 겪고 있다. 이에 보안 메신저를 악용하는 범죄사례는 늘고 있지만, 보안 메신저에 대한 데이터 복호화 연구는 필요하다. Element 보안 메신저는 대화 참여자만 대화 이력을 확인할 수 있도록 종단 간 암호화 기능을 제공하고 있으나 이를 복호화하는 연구는 미흡하다. 따라서 본 논문에서는 종단 간 암호화 기능을 제공하는 인스턴트 메신저 Element를 분석하고, 사용자의 패스워드 없이 Windows 자격 증명 관리자 서비스에 저장된 복호화키를 활용하여 암호화된 보안 채팅방의 이력을 평문으로 확인하는 방안을 제안한다. 또한, 디지털 포렌식 수사관점에서 유의미한 일반 및 보안 채팅 관련 아티팩트를 분석한 결과를 정리한다.

ABSTRACT

Recently, the investigation has been difficult due to the emergence of messengers that encrypt and store data for the purpose of protecting personal information and provide services such as end-to-end encryption with a focus on security. Accordingly, the number of crime cases using security messengers is increasing, but research on data decoding for security messengers is needed. Element security messengers provide end-to-end encryption functions so that only conversation participants can check conversation history, but research on decoding them is insufficient. Therefore, in this paper, we analyze the instant messenger Element, which provides end-to-end encryption, and propose a plaintext verification of the history of encrypted secure chat rooms using decryption keys stored in the Windows Credential Manager service without user passwords. In addition, we summarize the results of analyzing significant general and secure chat-related artifacts from a digital forensics investigation perspective.

Keywords: messenger, forensics, end-to-end encryption, Element

I. 서 론

국내외 전반에 걸쳐 COVID-19로 인한 비대면 문화가 자리 잡으며 사이버범죄 발생률이 증가하고 있다. 특히 피해가 큰 분야는 피싱으로 경찰청에 따르면 최근 3년간 피싱 범죄 발생 건수는 2020년

Received(10. 05. 2022), Accepted(11. 08. 2022)
* 본 논문은 2022년도 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2021-0-00540, GPU/ASIC 기반 암호알고리즘 고속화 설계 및 구현 기술개발)
본 논문은 2022년도 한국정보보호학회 하계학술대회에 발표한 우수논문을 개선 및 확장한 것임.
† 주저자, whwoals1023@gmail.com
‡ 교신저자, chlee@seoultech.ac.kr(Corresponding author)

18,726건, 2021년 20,402건으로 약 8.9% 증가했으며 그 피해액은 2020년 3,955억 원에서 2021년 5,006억 원으로 약 26.5% 증가했다[1]. 전체 피싱 범죄 피해의 절반 이상이 메신저 피싱으로, 메신저를 이용한 피싱은 2021년 상반기 11.2%에서 하반기 55.1%로 급증했다[2]. 이처럼 온라인 인스턴트 메신저를 이용한 사이버범죄는 꾸준히 증가해왔다.

하지만 개인정보보호를 목적으로 Kakao Talk, Line 등의 주요 메신저가 사용자 데이터를 암호화해 저장하기 시작하고 대화 내용 암호화, 종단 간 암호화, 화면캡처 차단 등의 보안 기능을 제공하는 Wickr me, Element 등의 인스턴트메신저가 등장하며 수사 과정에서 범죄사실 입증에 어려움을 겪고 있다. E2EE(End-to-End Encryption)라고도 부르는 종단 간 암호화는 발신 원부터 수신 원까지의 모든 과정에서 데이터를 암호화 상태로 전송하는 방식으로 2014년 정부의 카카오톡 서버 감청 사건 이후 대중화된 개념이다. 디지털 포렌식 관점에서 주요 개인정보 및 대화 정보를 암호화하는 보안 메신저에 대한 복호화 방안 연구는 매우 중요하다. 보안 메신저는 피싱, 마약 유통, 기술 유출 등 범국가적 범죄에 악용되고 있지만 이에 대한 복호화 방안 연구는 필요하다. 따라서 본 논문에서는 종단 간 암호화 서비스를 제공하는 오픈 소스 인스턴스 메신저 Element를 대상으로 Windows PC 환경에서 수집할 수 있는 사용자 행위 관련 아티팩트를 정리하고 암호화된 데이터에 대한 복호화 방안을 제시한다.

2장에서는 관련 연구를 소개하고 3장에서 Element의 보안 채팅 기능과 관련 데이터의 암호화 프로세스를 분석하고 복호화 방안을 제안한다. 4장에서는 Element의 일반 및 보안 채팅 관련 데이터를 분석하고 디지털 포렌식 관점에서 유의미한 아티팩트를 정리한다. 마지막으로 5장에서는 향후 연구 계획을 제시하고 결론으로 마무리한다.

II. 관련 연구

보안 메신저상에서 암호화된 데이터를 복호화하는 연구 결과들이 있다. Jihun Son 외 3인[3]은 보안 메신저 앱 Signal, Wickr, Threema 앱을 역공학을 통해 암호화 메커니즘을 파악해서 암호화된 정보들을 복호화하거나 앱 설정 파일에 저장되어있는 암호화에 사용된 키를 추출하여 복구하는 방안을 소개했다. Guido Cornelis Schipper외 2인[4]은 Element와 같이 matrix 프로토콜을 사용하는 애플리케이션인 Riot.im에 대한 포렌식 분석을 제시하였으나 암호화된 채팅 기록에 대한 평문 복구 방안이 제시되지 않았다. 이처럼 기존 연구 중에는 matrix 프로토콜을 사용하는 인스턴트메신저의 보안 채팅방의 대화 이력을 평문으로 확인할 방법은 없다. 따라서 본 연구에는 matrix 프로토콜을 사용하는 Element 메신저 앱에서 암호화된 채팅방 이력의 복호 방안과 연관된 아티팩트들을 정리한다.

III. Element 보안 채팅 데이터의 복호화 방안 분석

Element는 matrix 프로토콜을 기반으로 동작하는 보안 메신저이다. matrix 프로토콜은 2014년에 소개되었다. matrix는 “보안성, 탈중앙화, 실시간 통신을 지향하는 오픈 소스 프로젝트”로 보안 메신저 구현을 위한 종단 간 암호화(End-to-End Encryption), 브릿징(Bridging) 등의 기능을 SDK로 제공한다[4]. Element는 사용자가 채팅방을 개설할 때 보안 채팅 기능을 설정하면 종단 간 암호화(End-to-End Encryption)가 적용된다. 종단 간 암호화는 채팅방의 참여자만이 대화 이력을 확인할 수 있게 하는 기술이다.

Element의 보안 채팅은 보안 채팅방 내부 메시지에서 검색기능을 지원하기 위해 “C:\Users\user\AppData\Roaming\Element\EventStore” 하위에 ′Events.db′ 파일에 메시지 내용, 전송자, 전송 시각, 채팅방 ID 등이 평문으로 저장된다. 하지만 ‘Events.db’는 Element 앱 내부에서 생성된 임의의 키값을 이용하여 SQLCipher 모듈로 암호화하여 해당 파일을 보호한다. 따라서, 본 논문에서는 ′Events.db′ 파일의 암호화 프로세스를 분석하고 복호화하는 키를 찾는 방안을 제시한다.

3.1 Events. db 암호화 프로세스

해당 DB 파일에는 보안 채팅방과 관련된 정보만이 저장된다. Element는 github에 공개된 코드를 분석해 데이터베이스의 암호화 프로세스를 확인할 수 있다. 코드를 분석해 확인한 암호화 과정은 (Fig. 1)과 같다.

Element 앱 실행 후 로그인 시마다 32byte 길이의 임의의 값을 생성하고 Base64 인코딩한 후 패

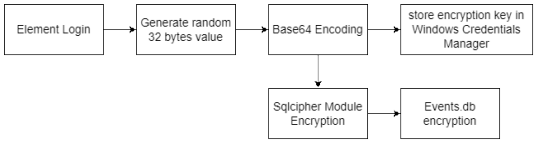

Fig. 1. Events.db encryption process

딩된 부분을 제거한다. 이 값을 사용해 Element는 SqlCipher 모듈의 AES-256으로 'Evnets.db'를 암호화한 후 자격 증명 관리자(Windows Credentials Manager)에 저장한다. 해당 값은 복호화를 할 때 사용하는 대칭키이다. '자격 증명 관리자'는 사용자가 지정한 사용자 이름 및 암호를 저장하기 위해 로컬 컴퓨터에 안전한 저장 공간을 만들고 이용할 수 있도록 하는 서비스이다. 제어판의 '자격 증명 관리자'가 관리하는 증명 정보는 (Fig. 2)와 같이 제어판, Windows API를 이용하여 접근할 수 있다.

자격 증명관리자가 관리하는 증명정보는 키워드 문자열을 이용해 Windows API를 통해 획득할 수 있다. 자격 증명 관리자는 증명 정보와 관련한 네트워크 주소를 키워드로 증명들을 관리하고, 엘리먼트 키워드 문자열은 〔element.io/seshat | @user_id | device_id〕이다. user_id는 로그인한 사용자의 계정 id, device_id는 로그인 시마다 생성되는 무작위의 10글자 길이 영문 대문자 문자열이다.

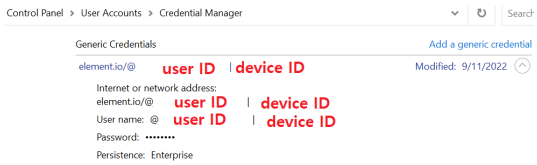

Fig. 2. Encryption key stored in Windows Credential Manager

3.2 Events. db 복호화 프로세스

본 연구에서 제안하는 'Events.db'의 복호화 방안 프로세스는 (Fig. 3)과 같다.

(Fig. 3)의 1번은, "C:\Users\user\AppData\Roaming\Element\Local Storage\leveldb\" 경로의 ".log" 확장자 파일 내부 'mx_device_id', 'mx_user_id' 문자열 옆에 기록된 가장 최근에 로그인한 계정의 user_id, device_id 값을 확인하는 과

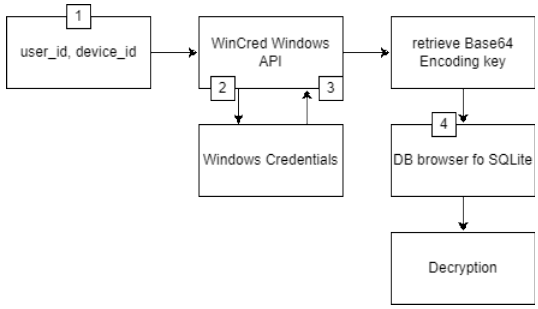

Fig. 3. Events.db Decryption process

정이다. 해당 값들을 확인하여 4.1절에서 확인한 "element.io/seshat | @user_id | device_id" 키워드 문자열을 확인한다.

(Fig. 3)의 2번은 생성한 키워드 문자열을 통해 Windows API로 자격 증명 관리자에게 쿼리를 보낸 과정이다. (Fig. 3)의 3번은 쿼리에 대한 요청으로 복호화키를 획득하고, (Fig. 3)의 4번에 DB파일 내부를 확인할 수 있는 "DB Browser for Sqlite" 도구에 복호화키 값을 전달하여 'Events.db'를 복호화한다.

3.3 시사점 및 한계

본 연구에서 제안하는 'Events.db'의 복호화 방안은 Windows API를 이용하여 복호화키를 획득해야 하므로, 활성화된 Windows 시스템상에서 획득해야한다.

또한, 'Events.db' 파일은 사용자 로그인 시 생성되고 로그아웃 시 자동으로 삭제되기 때문에 Element의 사용자가 앱에서 로그아웃하지 않아야 'Events.db' 파일이 로컬에서 확인할 수 있다. 하지만 Element 앱의 초기 설정으로는 'Events.db'를 생성하게 되어있다. 따라서 설정값에 변화를 주지 않으면, 보안 채팅방의 이력을 평문으로 확인할 수 있다.

IV. Element의 사용자 아티팩트 분석

4.1 보안 채팅 아티팩트

본 절에서는 Element 앱에서 사용자의 보안 채팅 기능 활용시 기록되는 아티팩트를 확인하여, 포렌식 수사관점에서 활용할 아티팩트와 앞의 복호화 된 DB의 정보를 정리한다. Element 앱의 사용 이력은 "C:\Users\user\AppData\Roaming\Eleme

Table 1. Secure Chat Related Artifacts

Specific Path	File type	Table	Feature	Forensic Artifact
\Events\Events.db	DB	events	sender	account information that caused the event
			server_ts	time when the event occurred
			room_id	room ID cached in rooms table
			type	Event type (sending messages, creating chat rooms, mentioning specific users, etc.)
			msgtype	Message Event Type (text, image, file transfer, etc.)
			source	json format data received from the server (Transmission and reception messages in plaintext format)
			profile_id	Profile ID cached in profile table
		rooms	id	Chat room identification ID applied within DB file
			room_id	The room_id registered on the Matrix home server and the corresponding home server address
		profile	id	User account identification ID applied within the DB file
			user_id	User account, user identification ID registered on the Matrix home server, and the corresponding home server address
\IndexedDB\vector_vector_0.indexeddb.leveldb\[0-9]{6}.log	log	-	transaction_id	time the message was sent(Unix Miliseconds)
			sender	the account information that sent the messag
			ciphertext	Encrypted message content

nt" 경로에 기록된다. 해당 경로의 하위에서 확인할 수 있는 아티팩트들을 정리한 것은 (Table 1)와 같다.

(Table 1)의 '\Events\Events.db'는 앞장의 복호화 과정을 거친 후 확인할 수 있는 보안 채팅방의 이력이다. 보안 채팅방 상에서, 대화 참여자들이 실제로 주고받은 평문 메시지는 (Table 1)의 'events' Table의 'source' Feature에 기록된다. 'source'는 서버로부터 전송받은 암호화된 이벤트를 복호화한 데이터로 json 형태로 저장된다. 'body'에서 평문 형태의 메시지를 확인할 수 있다. 또한 'source' Feature의 json 형태의 데이터에 (Table 1)의 'rooms' , 'profile'에 대응하는 데이터가 있다. 따라서 'source' Feature의 데이터를 확인하면, 메시지가 어느 채팅방에서 어떤 대화 참여자에 의해 전송되었는지 확인할 수 있다.

4.2 일반 채팅 아티팩트

본 절에서는 Element 앱에서 보안기능 적용 없이 기록되는 아티팩트를 확인하여, 포렌식 관점에서 활용할 방안을 다룬다. 해당 사용자의 이력은 이전 절에서 확인한 경로에 기록된다. 해당 경로에서 확인할 수 있는 아티팩트들을 정리한 것은 (Table 2)와 같다.

(Table 2)의 '\Session Storage\' 하위 경로의 '.log' 확장자 파일에는 앱상에서 발생하는 키보드 타이핑 이벤트, 클릭 이벤트 등의 기록이 존재한다.

키보드 타이핑 이벤트는 사용자가 채팅방에서 메시지를 작성하고 전송할 경우 기록되는 로그 파일로, 저장되는 형식은 (Table 3)과 같다.

사용자가 직접 타이핑한 경우뿐만 아니라 복사-붙여넣기 기능을 사용했을 때도 동일한 형식의 로그가 남는다. 채팅방 식별정보는 'room_id'로 기록되며 사용자가 실제로 입력한 메시지는 종단 간 암호화 지원 여부와 상관없이 'text'에 평문으로 저장된다. 따라서 Element 사용자의 가장 최근의 사용행위를 파악할 수 있다.

(Table 2)의 '\Cache\Cache_Data' 에는 사용자가 암호화 설정을 하지 않은 일반 채팅방에서 내려받은 파일 캐시가 저장된다. 이미지, 영상, 영상 썸네일, 문서 파일 등이 여기에 해당된다. 이 때 저장된 캐시 파일의 이름이 임의로 생성되고 파일 확장자가 존재하지 않기 때문에 원본 데이터를 확인하기 위해서는 HxD 등의 Hex Editor가 필요하다. Hex

Table 2. General Chat Related Artifacts

Specific Path	File type	Keyword	Forensic Artifact
Quota Manager	DB	Buckets.last_accessed	The most recent time the app was launched(Google Chrome Timestamp)
		Buckets.last_modified	the most recent time the app was closed (Google Chrome Timestamp)
\IndexedDB\vector_vector_0.indexeddb.leveldb\[0-9]{6}.log	log	transaction_id	time the message was sent (Google Chrome Timestamp)
		body	The content of the message sent
		sender	the account information that sent the message
\SessionStorage\[0-9]{6}.log	log	mx_cider_history	Keyboard input typing event while using app, click event log
		text	Contents of the typing event
\Cache\Cache_Data\f_[0-9]{6}	bin	-	File cache downloaded from a chat room (image, video, video thumbnail, document file)
Local Storage\leveldb\[0-9]{6}.log	log	mx_device_id	Device id of the recently logged-in account
		mx_user_id	Account information recently logged in

Table 3. Example of a typing event log

```
map-55-mx_cider_history_!room_id:home_server{1}X{"parts":[{"type":"plain","text":"testmessagefrom client"}]}
```

Editor를 통해 파일 시그니쳐를 확인하고 이에 대응하는 확장자로 확장자명을 변경하면 원본 데이터를 확인할 수 있다.

4.2.1 일반 채팅 아트팩트 대화내역

본 절에서는 Element 메신저에서 효율적으로 로그를 저장하기 위해 사용하는 Indexed DB 상의 내용을 확인하는 방법을 다룬다. Indexed DB는 Google에서 개발한 데이터베이스로 여러 분야에 사용되고 있다. 해당 DB가 관리하는 내역은 모두 압축되어 저장한다. 따라서 DB파일을 직접 접근하는 경우, 압축된 내역에 접근하기에 읽을 수 없는 형태로 확인할 수 없다.

하지만 이 압축된 DB를 압축해제 하여 확인할 수 있는 파이썬 라이브러리[9]가 있다. 해당 라이브러리를 이용하여 내부의 내용을 확인할 수 있게 스크립트[8]를 작성했다. 주의할 점은 Indexed DB는 읽기전용 모드로 DB를 읽을 수 없다. 불가피하게 DB에 접근할때 DB 구성 파일에 대한 쓰기 작업이 일어난다. 따라서 Indexed DB 자체의 무결성을 보장하려면 복사본에서 진행할 것을 권한다. 해당 스크립트[8]로 (Table 2)의 Indexed DB를 압축해제하여 확인할 수 있었던 정보는 다음 (Table 4)와 같다.

Table 4. Artifacts found in indexedDB

DB Id	DB Name	Forensic Artifact
1	logs	Information about logging in/out to element, creating/modifying chat rooms
2	matrix-react-sdk	Access token value for element account
9	matrix-js-sdk:crypto	Secure Chat Room Key Exchange Log
10	matrix-js-sdk:riot-web-sync	Secure chat room/general chat room chat and activity details in json format

(Table 4)에서 확인할 수 있는 정보 중에 일반 채팅 아티팩트정보를 획득 할 수 있는 정보는 DB Id 10의 matrix-js-sdk:rit-web-sync에 있다. 이는 element 서버 내에 저장하고 있던 보안 채팅방 및 일반채팅 방의 활동 내역을 디바이스에 설치되어 있는 element 메신저 애플리케이션에서 확인할 수 있도록 데이터를 가져올 때 남겨지는 값들이다. 여기에 남겨지는 데이터를 통해 일반 채팅방에서의 대화 내역들은 평문으로 확인할 수 있다. 하지만, 보안 채팅방의 대화내역들은 암호화가 적용되어 평문으로 확인할 수 없다. 다음 (Table 5)는 (Table 4)의 일반 채팅방 채팅 및 활동 내역에서 확인할 수 있는 정보들이다.

(Table 4)에서 계정의 접근 권한을 취득 할 수 있는 정보가 있다. 해당 표의 Indexed DB가 가지고 있는 DB인 matrix-react-sdk에 저장된 access token 값이다. 값의 예시는 다음 (Table 6)와 같다.

해당 값은 본래 element 메신저에서 채팅 봇 등 메신저 관련 자동화 서비스를 구축할 때 서비스에서 메신저로의 자원접근 권한을 취득하기 위해 사용하는

Table 5. General chat room history json keyword

Keyword	Forensic Artifact
body	Plain text messages sent to chat rooms, File name transferred
msgtype	Types of messages sent to chat rooms Send (files/texts)
origin_server_ts	The time the activity details received from the chatroom server occurred
semder	The name of the account that triggered the activity in the chat room
room_id	Chat room address where the event was processed
m.read	Whether the message was read or not
room_version	the version of the chat room
creator	The account that created the chat room
displayname	The name of the account in the chat room that appears

Table 6. Access token value format

syt_amp(alphabetic case, combination of numbers in 20 digits)_(alphabetic case, combination of numbers in 20 digits)_(alphabetic case, combination of numbers in 6 digits)

권한이다. 서비스를 구축하는 사용자의 계정의 권한을 서비스를 제공할 프로그램에 위임할 수 있도록 만든 역할이다. element 메신저에서 제공하는 api에는 해당 접근권한을 이용하여 메신저 내의 계정의 자원에 접근할 수 있다. 해당 자원들은 모두 json 형식으로 이루어진 정보들이다. 본 연구에서는 (Table 4)에서 확인한 계정권한을 이용하여 서버에 저장되어있는 채팅내역 및 활동내역을 조회할 수 있는 자바스크립트[9]를 작성하였다.

해당 스크립트[9]는 서버에서 보관하고 있던 계정이 참가한 채팅방의 타임라인 정보들을 획득하여 채팅 방에서의 활동내역을 확인할 수 있게한다. 확인가능한 이벤트의 종류는 다음 (Table 7)와 같다.

(Table 7)이외에도 확인할 수 있던 이벤트 중에 암호화된 채팅내역도 확인할 수 있다. 암호화 채팅내역은 서버에서도 암호화되어 저장되어있음을 확인할 수 있다. 하지만 암호화에 사용된 알고리즘의 정보와 암호화에 사용된 device_id, sender_key, session_id 정보를 같이 확인할 수 있다.

Table 7. Chat room timeline event type

Keyword	Forensic Artifact
name	Chat room name setting event
guest_access	Chat Room Access Settings Event
topic	Chat Room Chat Topic Settings Event
membership	Invite to chat room and confirm attendance event
join_rule	Chat room disclosure setting event
msgtype	Chat room chat history and the file name you sent

V. 결론 및 향후 과제

본 논문에서는 보안 메신저 중 하나인 Element를 대상으로 앱 사용 시 남는 아티팩트와 암호화된

db 파일을 복호화하는 방안을 찾았다. 또한, Windows API를 이용하여 복호화에 필요한 정보를 활성 시스템에서 획득할 수 있는 도구[6]를 개발하여 github에 공개하였다. 그 외에도 Indexed DB의 비직렬화를 지원하는 python 라이브러리[10]를 이용하여 로컬에 저장된 채팅내역을 확인할 수 있었다. 그에 따른 아티팩트를 분석하였고 디지털 포렌식 증거 수집에 도움을 줄 것으로 기대한다.

로컬에 저장된 아티팩트 중 서버에 계정이 가진 자원에 접근가능한 접근권한을 획득하여 해당 자원에 접근하는 스크립트[9]를 작성하여 공개했다. 해당 스크립트로 암호화 채팅방에서 대화내역을 암호화하여 송신할 때 사용하는 정보들을 확인할 수 있었다. 본 논문에서 제안된 방법은 로그인 정보가 남아있을 경우에 한하여 유효하다. 관련 정보없이 확인된 정보를 통해 추후에 matrix 프로토콜에서 사용하는 암호화 방식을 연구하여 암호화된 채팅 내역을 복호화할 수 있는 방안을 찾을 예정이다.

References

[1] Public Data Portal, "Monthly Status of Voice Phishing at the National Police Agency" ,https://www.data.go.kr/data/15099013/fileData.do, 2022.11.07

[2] Financial Supervisory Service, "Analysis of damage from voice phishing in 2021", https://www.fss.or.kr/fss/bbs/B0000188/view.do?nttId=55444&menuNo=200218&pageIndex=1, 2022.11.07.

[3] Jihun Son, Yeong Woong Kim, Dong Bin Oh, Kyounggon Kim, "Forensic analysis of instant messengers: decrypt signal, wickr, and threema," Forensic Science International: Digital Investigation ,Volume 40, Mar. 2022, Article: 301347, ISSN 2666-2817

[4] Guido Cornelis Schipper, Rudy Seelt, Nhien-An Le-Khac, "Forensic analysis of matrix protocol and riot.im application," Forensic Science International: Digital Investigation, Volume 36, Supplement, Mar 2021, Article: 301118,ISSN 2666-2817

[5] vector-im, "element-desktop", https://github.com/vector-im/element-desktop, 2022.11.07

[6] loud11, "windows credential", "testrepo_for_credential_retrieve", https://github.com/loud11/testrepo_for_credential_retrieve/releases, 2022.11.07

[7] Matrix.org, "matrix protocol", https://matrix.org/, 2022.11.07

[8] loud11, "indexed DB", https://github.com/loud11/simple-tool/blob/main/deseiral.py, 2022.11.07

[9] loud11,"matrix", https://github.com/loud11/simple-tool/blob/main/matrix_communication.js, 2022.11.07

[10] obsidianforensics,"indexed DB", https://github.com/obsidianforensics/ccl_chrome_indexeddb, 2022.11.07.

〈저자소개〉

조 재 민 (Jae-min Cho) 정회원
2022년 8월: 서울과학기술대학교 컴퓨터공학과 졸업
2022년 8월~현재: 고려대학교 정보보호학화 석사과정
〈관심분야〉 정보보호, 디지털 포렌식, 역공학

변 현 수 (Hyeon-su Byun) 학생회원
2018년 2월~현재: 서울과학기술대학교 학사과정
〈관심분야〉 정보보호, 디지털포렌식, CTI

윤 희 서 (Hui-seo Yun) 학생회원
2018년 2월~현재: 서울과학기술대학교 학사과정
〈관심분야〉 정보보호, 디지털 포렌식, CTI

서 승 희 (Seung-hee Seo) 학생회원
2012년 3월: 서울과학기술대학교 컴퓨터공학과 학사
2017년 3월: 서울과학기술대학교 컴퓨터공학과 석사
2020년 3월~현재: 서울과학기술대학교 컴퓨터공학과 박사과정
〈관심분야〉 정보보호, 디지털 포렌식, 메모리 포렌식, 모바일 포렌식, 패스워드 리커버리

이 창 훈 (Chang-hoon Lee) 종신회원
2001년 3월: 한양대학교 자연과학부 수학전공 학사
2003년 3월: 고려대학교 정보보호대학원 석사
2008년 3월: 고려대학교 정보경영전문대학원 정보보호전공 박사
2008년 4월~2008년 12월: 고려대학교 정보보호연구원 연구교수
2009년 3월~2012년 2월: 한신대학교 컴퓨터공학부 조교수
2012년 3월~2015년 3월: 서울과학기술대학교 컴퓨터공학과 조교수
2015년 4월~2019년 3월: 서울과학기술대학교 컴퓨터공학과 부교수
2019년 4월~현재: 서울과학기술대학교 컴퓨터공학과 정교수
〈관심분야〉 정보보호, 사이버 보안, CTI, IoT보안, 디지털포렌식, 암호학 등