인스턴트 메시지 아티팩트 분석

System Software Lab
이근탁

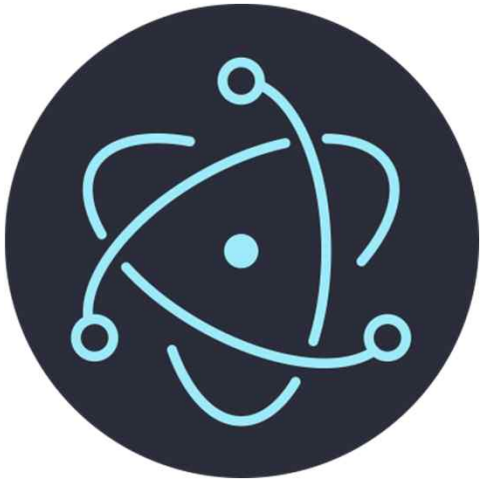
CONTENTS

CHAPTER

01

Electron

01 Electron

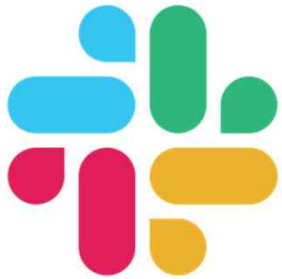


OpenJS Foundation 개발

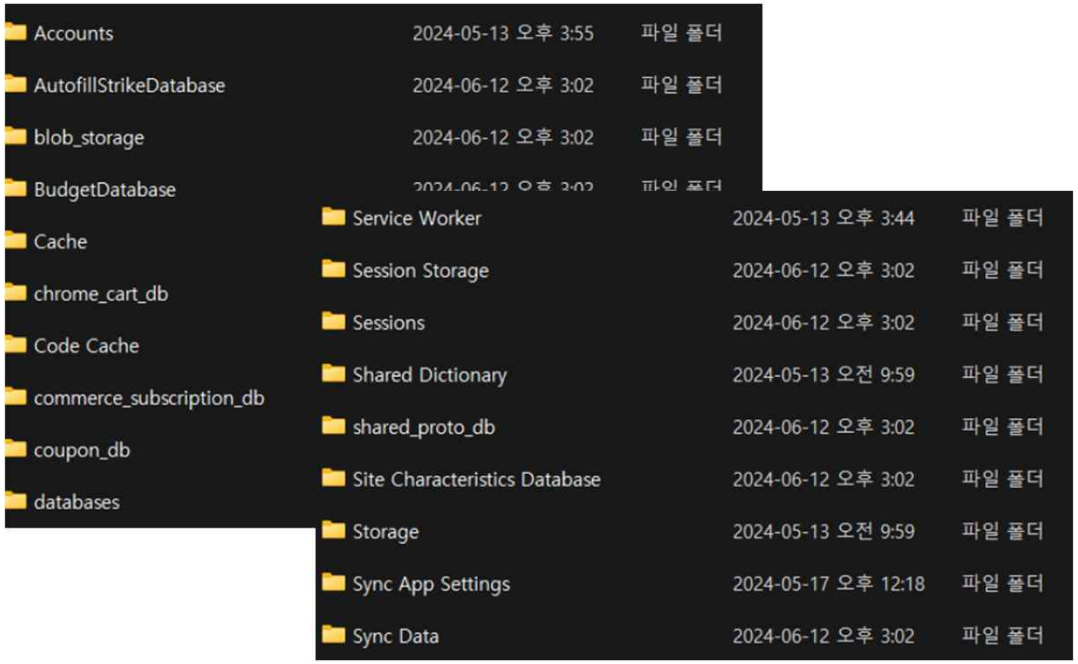
Chromium, NodeJS 기반

HTML, CSS, JS 이용하여
웹 제작하듯이
데스크톱 앱 만드는 플랫폼

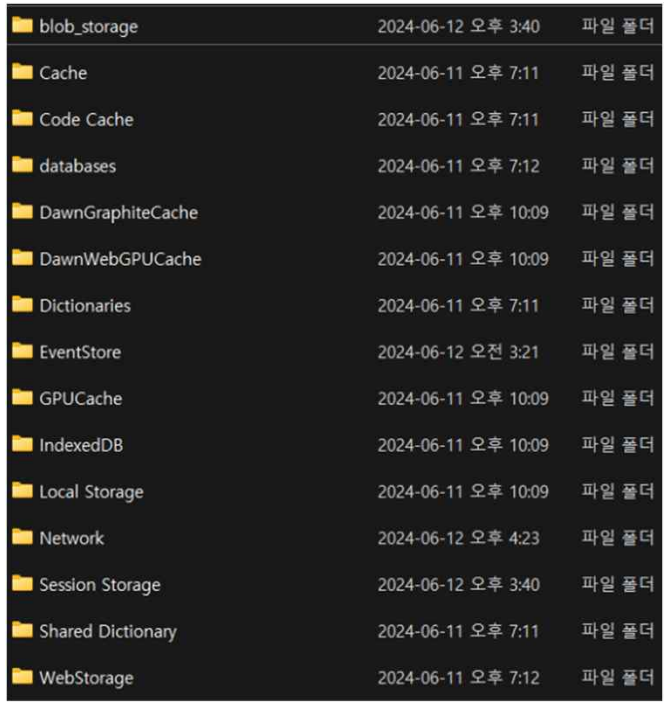
역으로 데스크톱 to 웹도 가능



01 Electron



Chrome 폴더



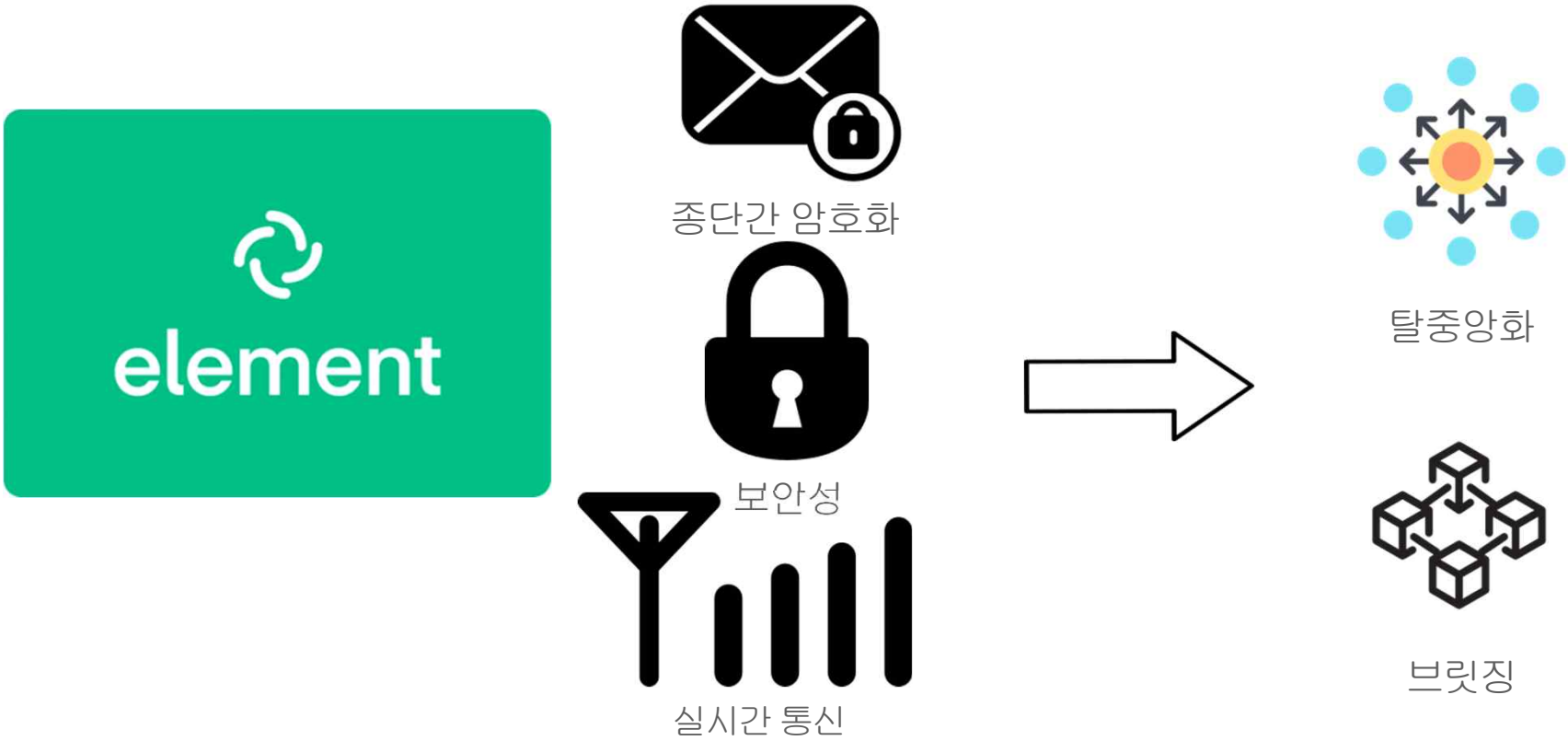
Electron 출신 친구들 폴더

CHAPTER

02

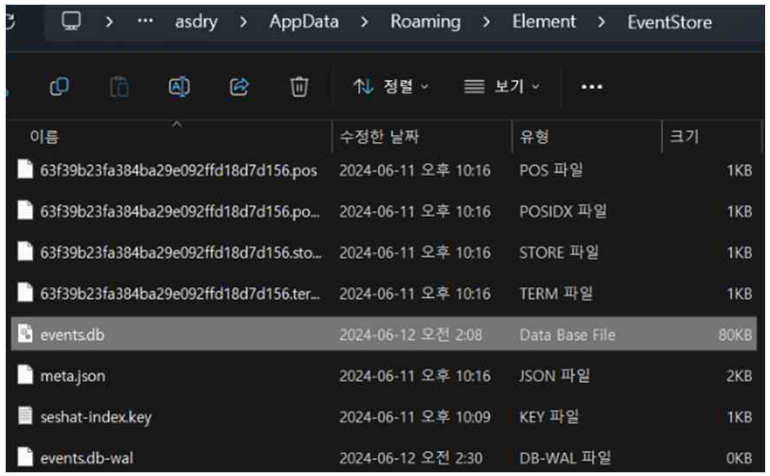
Element

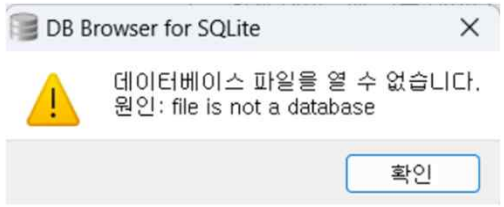
02 Element –Matrix 프로토콜 이용



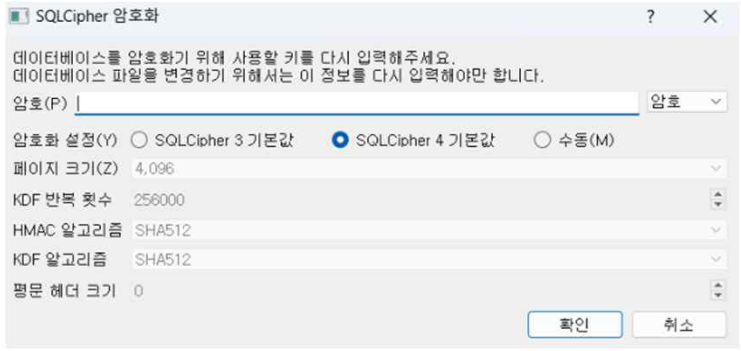
02 Element –DB 암호화

C:₩Users₩asdry₩AppData₩Roaming₩Element₩EventStore₩events.db





SQLite



SQL Cipher

Encrypted.....

02 Element

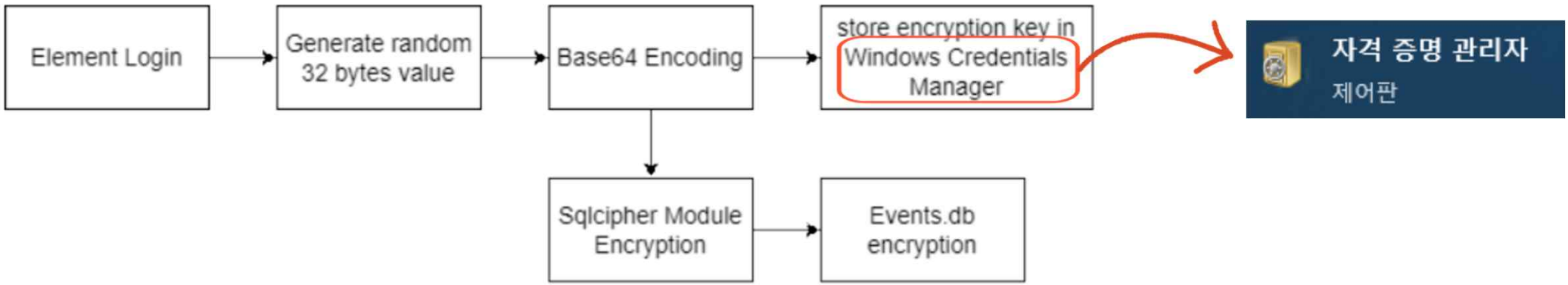
오픈소스 프로젝트

```
async function getOrCreatePassphrase(key: string): Promise<string> {
    if (keytar) {
        try {
            const storedPassphrase = await keytar.getPassword("element.io", key);
            if (storedPassphrase !== null) {
                return storedPassphrase;
            } else {
                const newPassphrase = await randomArray(32);
                await keytar.setPassword("element.io", key, newPassphrase);
                return newPassphrase;
            }
        } catch (e) {
            console.log("Error getting the event index passphrase out of the secret store", e);
        }
    }
    return seshatDefaultPassphrase;
}
```

```
bool GetPassword(const std::string& service,
                 const std::string& account,
                 std::string* password) {
  std::string target_name = service + '/' + account;

  CREDENTIAL* cred;
  if (::CredRead(target_name.c_str(), CRED_TYPE_GENERIC, 0, &cred) == FALSE)
    return false;

  *password = std::string(reinterpret_cast<char*>(cred->CredentialBlob),
                          cred->CredentialBlobSize);
  ::CredFree(cred);
  return true;
}
```

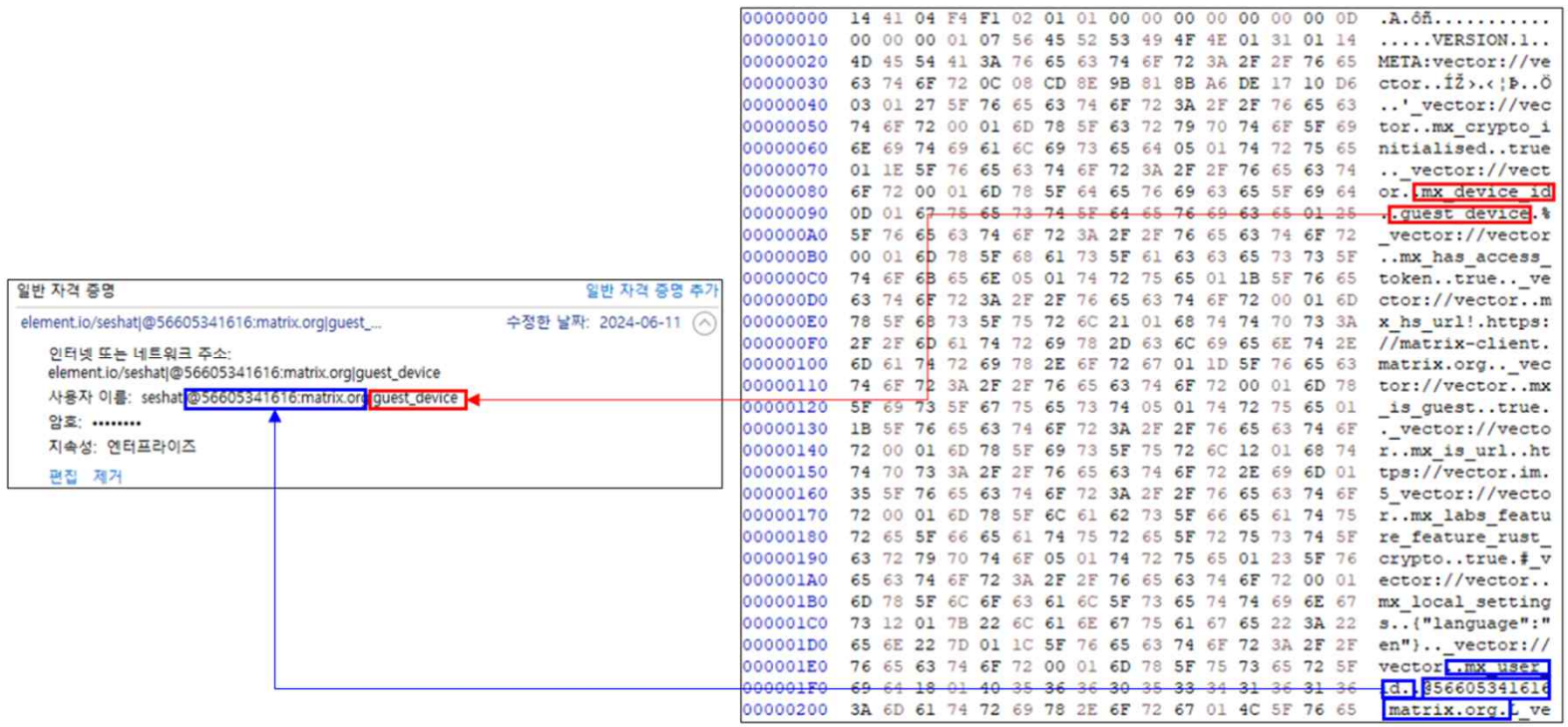
02 Element

동작 구조



02 Element

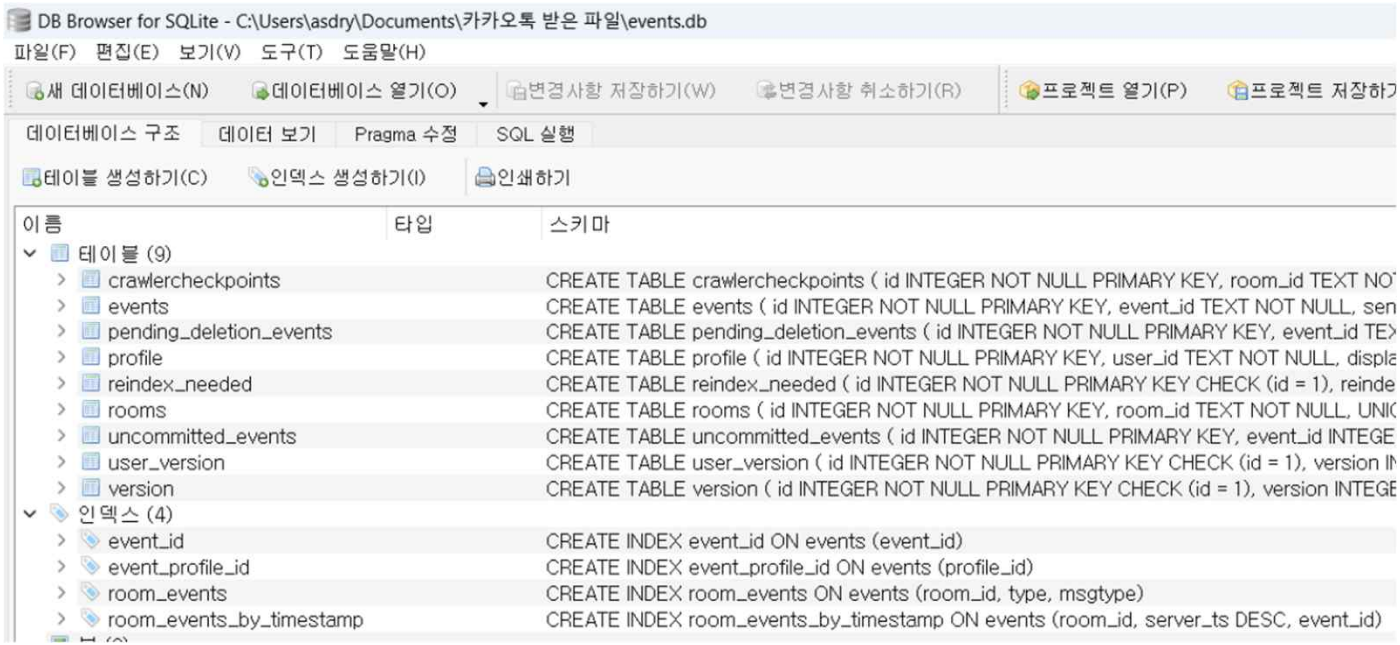
C:₩Users₩asdry₩AppData₩Roaming₩Element₩Local Storage₩*.log



02 Element

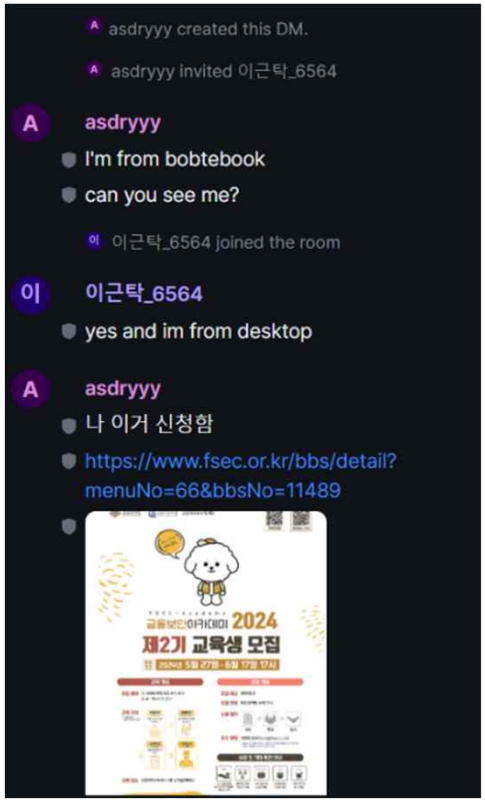
```
BOOL result = CredEnumerate(NULL, 0, &count, &credentials);
if (!result) {
    std::cerr << "Failed to enumerate credentials: " << GetLastError() << std::endl;
    return;
}

for (DWORD i = 0; i < count; i++) {
    PCREDENTIAL credential = credentials[i];

    std::wcout << L"Credential Name: " << credential->TargetName << std::endl;
    std::wcout << L"Type: " << credential->Type << std::endl;

    if (credential->UserName) {
        std::wcout << L"User Name: " << credential->UserName << std::endl;
    }

    if (credential->CredentialBlobSize > 0 && credential->CredentialBlob != NULL) {
        // The password might not be stored in plain text; handling depends on the actual use case.
        std::wcout << L"Credential Blob: " << (char*)credential->CredentialBlob << std::endl;
    }

    std::wcout << L"Persist: " << credential->Persist << std::endl;
    std::wcout << L"-------------------------------" << std::endl;
}
```

02 Element



복호화 성공

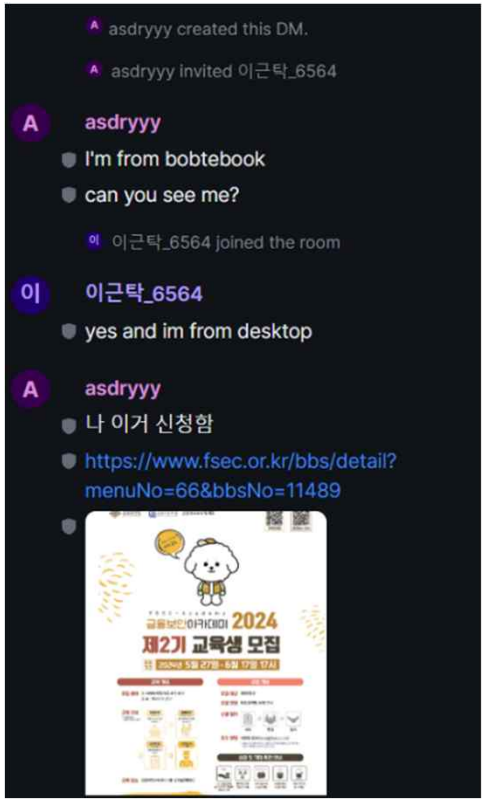
02 Element



테이블(T): profile

	id	user_id	displayname	avatar_url
	...	필터	필터	필터
1	1	@asdryyy:matrix.org	asdryyy	
2	2	@gguunnttaakk:matrix.org	이근탁_6564	

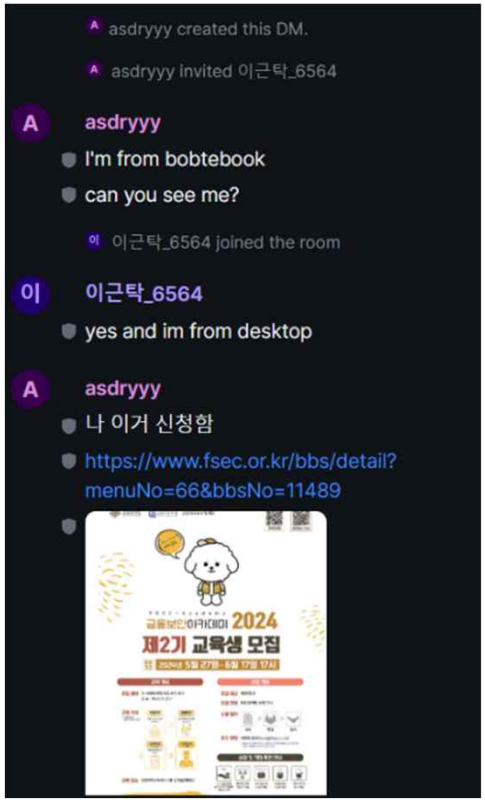
02 Element



id	event_id	sender	server_ts	room_id	type	msgtype	source	profile_id
···	필터	필터	필터	필터	필터	필터	필터	필터
1	$9ceD-···	@asdryyy:matrix.org	1718104838400	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	1
2	$MpnmNo57r6jHYx8w_wzVO6NHBdVD···	@asdryyy:matrix.org	1718104839882	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	1
3	$1JyiHjqOXLQKopuGxVLAsX7WDlm_8···	@gguunnttaakk:matrix.org	1718104890450	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	2
4	$k73lUQi4skkdsR3aw3h543xeUqxjDa7···	@asdryyy:matrix.org	1718104951854	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	1
5	$LZbKLltR62a2xV0_3rzx_aphEZXK3lU···	@asdryyy:matrix.org	1718104954200	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	1
6	$pbI7t0psYK1mHTN_UitoTxV1MQo2l8···	@asdryyy:matrix.org	1718104971539	1	m.room.message	m.image	{"algorithm":"m.megolm.v1.aes-···	1
7	$_D1Q_21qVPGBtWEjqUbT3J6XrblVCd···	@gguunnttaakk:matrix.org	1718104978470	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	2
8	$vC9a9Qlnxs_0lN3tkzsfOvzeqGoPmb···	@asdryyy:matrix.org	1718105013083	1	m.room.message	m.file	{"algorithm":"m.megolm.v1.aes-···	1
9	$U_8tmW-···	@asdryyy:matrix.org	1718105023989	1	m.room.message	m.text	{"algorithm":"m.megolm.v1.aes-···	1
10	$diBLgTIS5MPLL0Z6[illegible]							2
11	$cQZCZYpLZE0tRkkl[illegible]							2
12	$JPJbQjs1FOb7SEqbl[illegible]							2
13	$7fK7uy8DMGMZdu_[illegible]							2

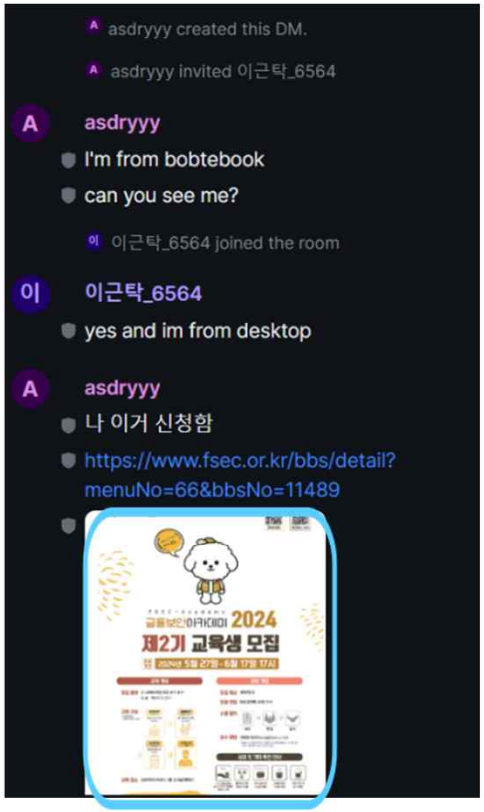

```
{"algorithm":"m.megolm.v1.aes-sha2","content"
:{"body":"I'm from bobtebook","m.mentions":{},
"msgtype":"m.text"},"curve25519Key":
"8OsUwq1BC8DeyuYrXN7cSfpcCz6Iv0pMoRYVKpOaQGY",
"ed25519Key":
"jzwKWOZDfMkjiqCucgmk+EYc9cJpLyA/uuuFF8AEdH0",
"event_id":
"$9ceD-BC9jmdM6vF8D2DXfvVF5bs7QOFA2AQ2_qOqt7M"
,"forwardingCurve25519KeyChain":[],
```

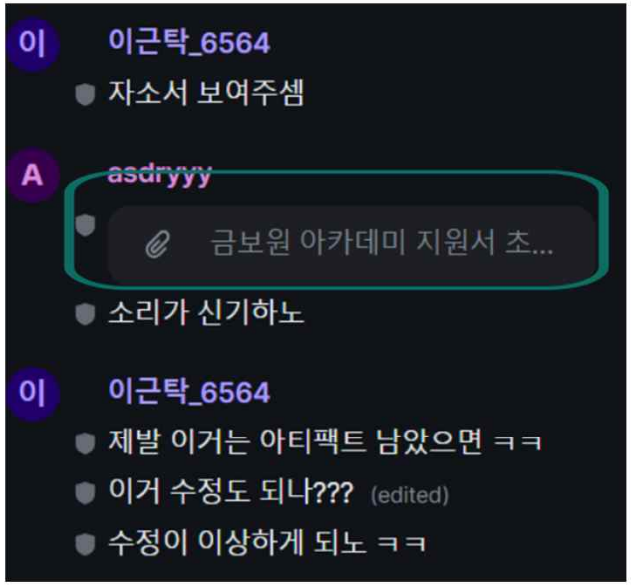
02 Element



{"algorithm":"m.megolm.v1.aes-sha2","content":{"body":"I'm from bobtebook","m.mentions":{},"msgtype":"m.text"},"curve25519Key":"8OsUwq1BC8DeyuYrXN7cSfpcCz6Iv0pMoRYVKpOaQGY","ed25519Key":"jzwKWOZDfMkjiqCucgmk+EYc9cJpLyA/uuuFF8AEdH0","event_id":"$9ceD-BC9jmdM6vF8D2DXfvVF5bs7QOFA2AQ2_qOqt7M","forwardingCurve25519KeyChain":[],"origin_server_ts":1718104838400,"room_id":"!noZGlLvRzUpswsyzmF:matrix.org","sender":"@asdryyy:matrix.org","type":"m.room.message","unsigned":{}}

02 Element

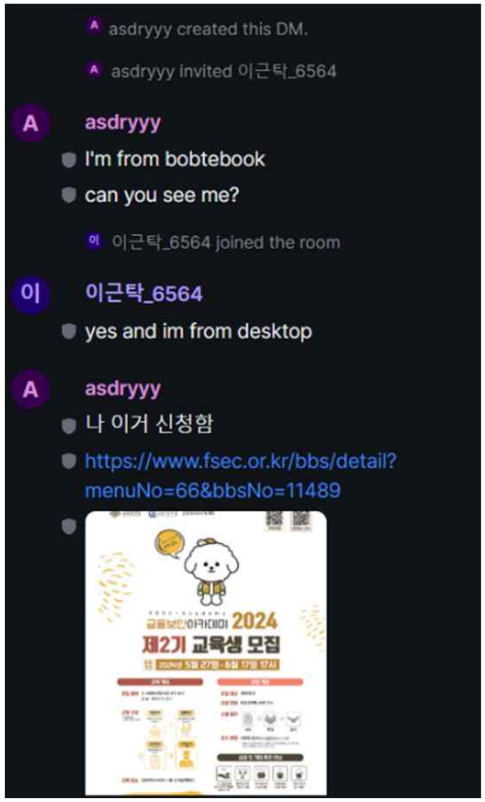


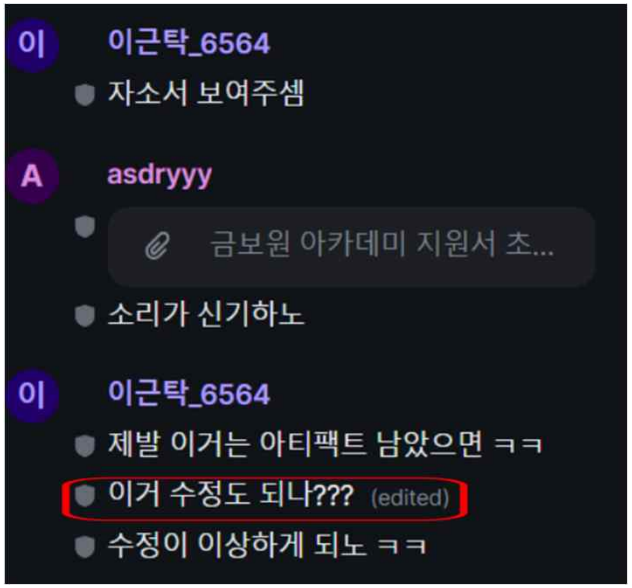


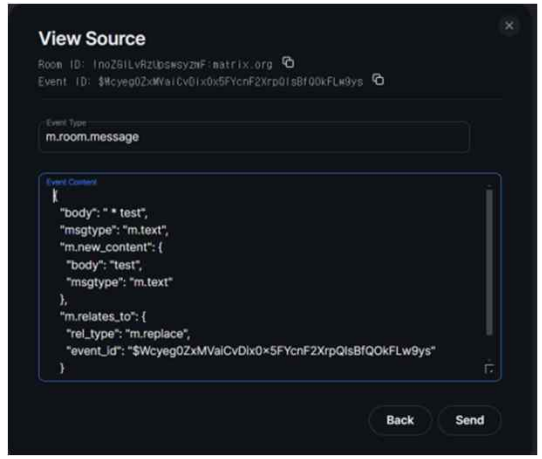
"body":"이미지 파일명",
"msgtype":"m.image"
url : "mxc://matrix.org/ryJRYMzPVWPuMDCcgPGosOar"
sha-256 hash

"body":"파일명",
"msgtype":"m.file"
url : "mxc://matrix.org/bAjmnIciHUxDqmmsHZaNtiio"
sha-256 hash

02 Element








```
{"algorithm":"m.megolm.v1.aes-sha2","content"
:{"body":" * 이거 수정도 되나??","m.new_content"
:{"body":"이거 수정도 되나???","msgtype":"m.text"
},"m.relates_to":{"event_id":
"$cQZCZYpLZE0tRkkI6nhppdNXiti735ER9RITB_-rRX0"
,"rel_type":"m.replace"},"msgtype":"m.text"},
"curve25519Key":
```

02 Element – 한계점

1. 로그아웃 되어 있으면 획득 불가능 (Events.db X)
2.
2.
2. 로그아웃 후 다시 로그인하면 로그인 이전의 대화 내역은 소실됨
3.
3.
3. 크레덴셜을 획득하려면 Windows API를 써야하기 때문에 라이브포렌식을 해야함