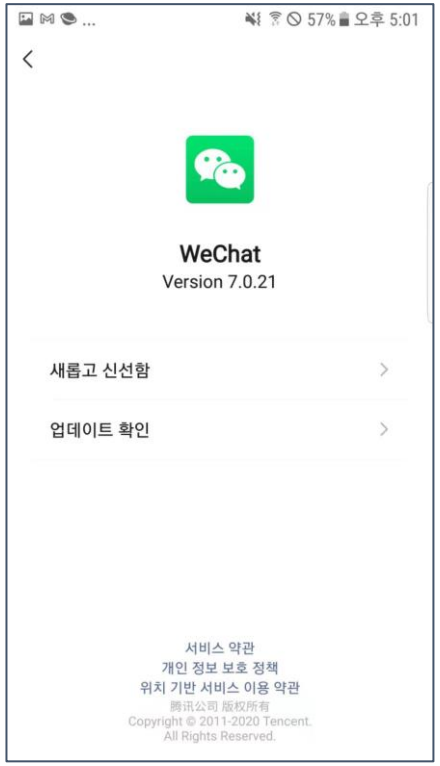
FaS
WeChat 실습

정보보안암호수학과 20182209 김수빈

WeChat 데이터 추출

루팅된 갤럭시 S6 edge



WeChat 7.0.21

루팅 후 adb pull 데이터 추출

```
C:₩Users₩kimsu₩AppData₩Local₩Android₩Sdk₩platform-tools>adb shell
zerolteskt:/ $ su
zerolteskt:/ # cd data₩data
sush: cd: /datadata: No such file or directory
2|zerolteskt:/ # cd data/data
```

adb shell 들어가기
su로 권환 획득
data/data 위치로 이동

```
zerolteskt:/data/data # ls
com.samsung.android.mdm                          com.skt.tservice.utility
com.samsung.android.messaging                    com.sktelecom.smartcard.SmartcardService
com.samsung.android.networkdiagnostic            com.tencent.mm
com.samsung.android.oneconnect                   com.trustonic.tuiservice
com.samsung.android.personalpage.service         com.wsomacp
com.samsung.android.provider.filterprovider      com.wssnps
```

com.tencent.mm 찾기

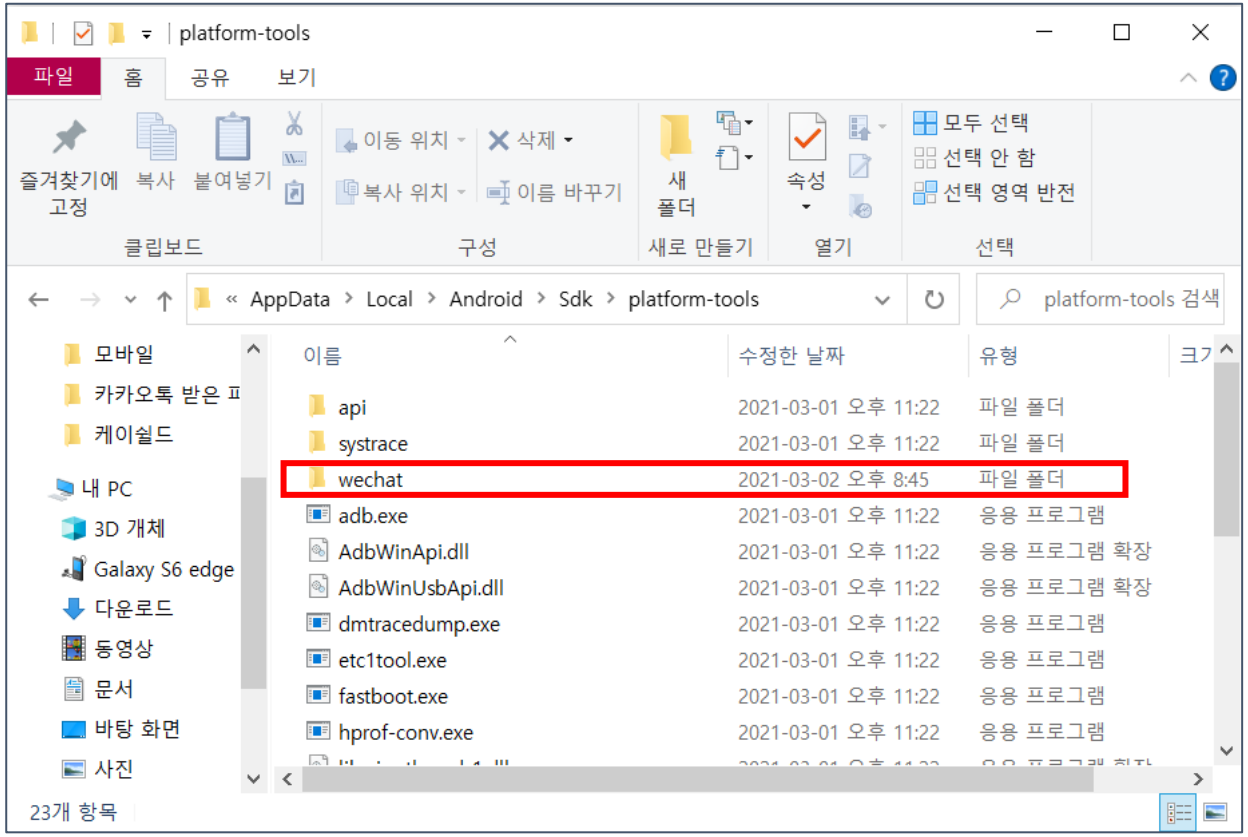
```
zerolteskt:/data/data # cd com.tencent.mm
zerolteskt:/data/data/com.tencent.mm # ls
MicroMsg     app_download_internal app_recover_lib  app_turingmm     app_xwalk_2691  cache      face_detect shaders
account.bin  app_font              app_recovery_lib app_webview      app_xwalkconfig code_cache files       shared_prefs
app_cache    app_icon_assets       app_tbs_64       app_webviewcache app_xwalkplugin databases  no_backup
app_dex      app_lib               app_textures     app_xwalk_-1     appbrand        extqbar    scan_goods
zerolteskt:/data/data/com.tencent.mm # cp -r /data/data/com.tencent.mm /sdcard/wechat
zerolteskt:/data/data/com.tencent.mm # exit
zerolteskt:/ $ exit
```

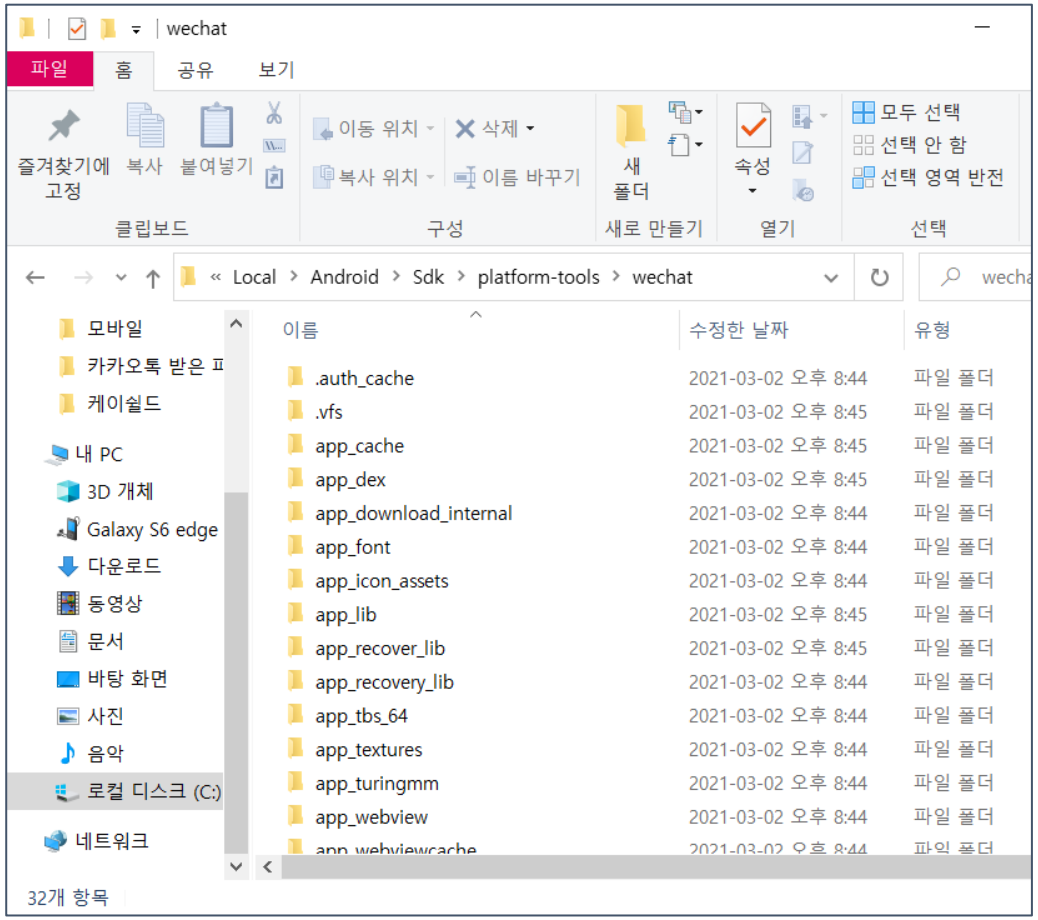
/data/data/com.tencent.mm을 보기 쉽게 /sdcard/wechat에 복사

루팅 후 adb pull 데이터 추출

adb pull 명령어로 /sdcard/wechat 데이터 획득

```
C:₩Users₩kimsu₩AppData₩Local₩Android₩Sdk₩platform-tools>adb pull /sdcard/wechat
/sdcard/wechat/: 1057 files pulled, 0 skipped. 18.1 MB/s (557409646 bytes in 29.438s)
```





루팅 후 adb pull 데이터 추출

```
C:₩Users₩kimsu₩AppData₩Local₩Android₩Sdk₩platform-tools>adb shell pm list packages -f | find "tencent"
package:/data/app/com.tencent.mm-1/base.apk=com.tencent.mm

C:₩Users₩kimsu₩AppData₩Local₩Android₩Sdk₩platform-tools>adb pull /data/app/com.tencent.mm-1/base.apk wechat.apk
/data/app/com.tencent.mm-1/base.apk: 1 file pulled, 0 skipped. 25.8 MB/s (122686963 bytes in 4.540s)
```

1. com.tencent.mm의 apk 경로 찾기
2. adb pull 명령어로 wechat.apk 추출

WeChat 데이터

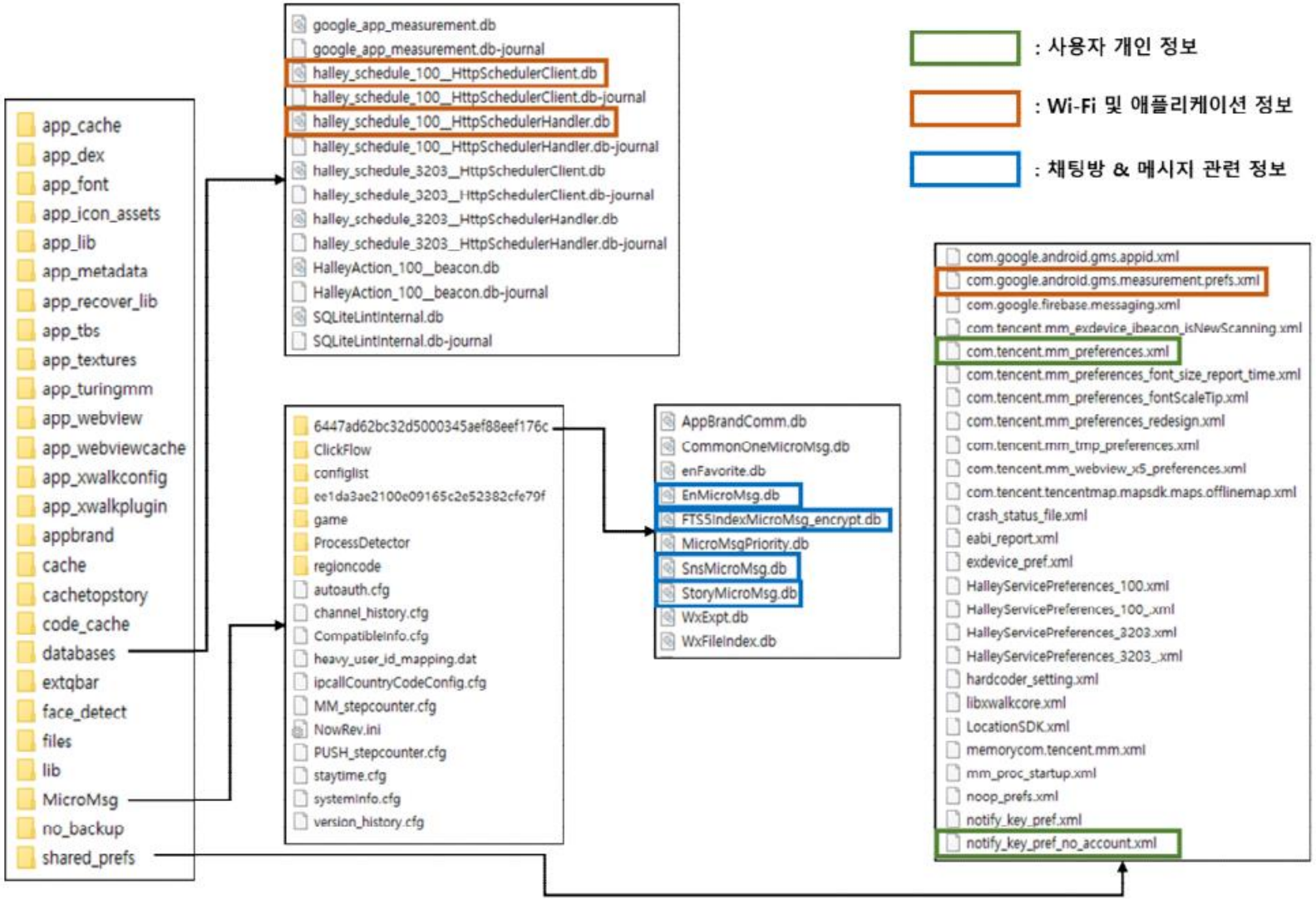


그림 3. WeChat 모바일 버전 데이터 구조 일부 (/data/data/com.tencent.mm)
Figure 3. Part of WeChat Mobile Version Data Structure (/data/data/com.tencent.mm)

WeChat 암호화 데이터

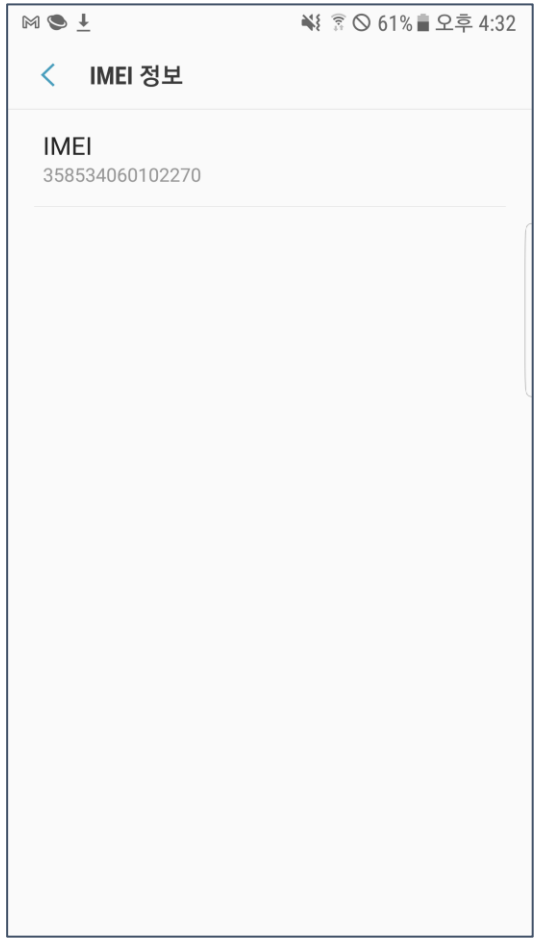
Element		PC	Mobile		
Classification		A	A	B	
				FTS5IndexMicroMsg_encrypt.db	MicroMsgPriority.db
Encryption Algorithm		AES-256-CBC	AES-256-CBC	AES-256-CBC	
Algorithm Argument	Size of Page	4,096	1,024	4,096	
	HMAC Function	O	X	O	
	Number of Iterations (Times)	64,000	4,000	64,000	
Key Generation Method		256 bits Hexadecimal String	Left7(MD5(UIN||IMEI))	FTS5IndexMicroMsg_encrypt.db	MicroMsgPriority.db
				Left7(MD5(UIN||IMEI||Account)	Left7(MD5(UIN||Account||IMEI)

A : En*.db / **B** : FTS5IndexMicroMsg_encrypt.db, MicroMsgPriority.db
O : Use, **X** : Disuse

UIN, IMEI, Account 찾기

WeChat 암호화 데이터

스마트폰에서 IMEI 정보 확인



wechat₩shared_prefs₩com.tencent.mm_preferences.xml에서
UIN, Account 확인

```
<string name="last_login_use_voice">0</string>
<string name="last_login_uin">2299112163</string>
```

```
<string name="login_weixin_username">wxid_u6dapn2wehqw12</string>
```

wechat₩shared_prefs₩system_config_prefs.xml에서 UIN확인

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <boolean name="set_service" value="false" />
    <boolean name="first_launch_weixin" value="false" />
    <string name="builtin_short_ips">6,203.205.219.196,80,hkshort.weixin.q
    <int name="default_uin" value="-1995855133" />
    <int name="launch_last_status" value="2" />
    <string name="support.weixin.qq.com">hksupport.weixin.qq.com</string>
    <int name="appbrand_video_player" value="-1" />
</map>
```

UIN 2개
→ 2299112163
→ -1995755133

'2299112163'는 '-1995755133'에서 '4294967296' (부호없는 최대 수)를 더한 값
→ 사실상 둘은 같은 값이다.

WeChat 암호화 데이터

EnMicroMsg.db

1. ** EnMicroMsg.db 암호화 알고리즘 **은 변경되지 않았습니다.이 암호화 알고리즘은 IMEI와 uin이 스 플라이 싱 된 후 계산 된 32 비트 MD5 값이며 처음 7 자리를 취합니다 (uin이 음수이면 변경할 필요가 없습니다.
2. ** IMEI 및 마지막 로그인 uin 추출 방법 **은 변경되지 않았습니다. IMEI는 CompatibleInfo.cfg 파일에 있고 마지막 로그인 uin은 system_conf_prefs.xml 파일에 있습니다.
3. WeChat 사용자 데이터 저장 디렉토리의 이름은 여전히 "mm + uin"순서로 연결된 MD5 값입니다.

→ LEFT7(MD5(IMEI + UIN))
→ UIN은 음수

FTS5IndexMicroMsg_encrypt.db

여기서 유의해야 할 점은 uin이 음수이면 직접 연결할 수 없으며 4294967296 (부호없는 최대 수)을 추가해야하며 결과 양수는 연결을위한 최종 uin으로 사용됩니다.

→ LEFT7(MD5(UIN + IMEI + ACCOUNT))
→ UIN 양수 (음수면 4294967296 더해주기)

WeChat 암호화 데이터

DB 복호화 키 구하기

```python
import hashlib

UIN = '2299112163'
IMEI = '358534060102270'
Account = 'u6dapn2wehqw12'

En_key = UIN+IMEI
FTS_key = UIN+IMEI+Account
Priority_key = UIN+Account+IMEI

En = hashlib.md5(En_key.encode()).hexdigest()
FTS = hashlib.md5(FTS_key.encode()).hexdigest()
Priority = hashlib.md5(Priority_key.encode()).hexdigest()

print('En*.db key :', En[:7])
print('FTS5IndexMicroMsg_encrypt.db key :', FTS[:7])
print('MicroMsgPriority.db key :', Priority[:7])
```

```
En*.db key : 7835ba0
FTS5IndexMicroMsg_encrypt.db key : 3280741
MicroMsgPriority.db key : e7e2d1d
```

```python
import hashlib

UIN1 = '-1995855133'.encode()
UIN2 = '2299112163'.encode()
IMEI = '358534060102270'.encode()
Account = 'wxid_u6dapn2wehqw12'.encode()

En_key = IMEI+UIN1
FTS_key = UIN2+IMEI+Account
Priority_key = UIN2+Account+IMEI

En = hashlib.md5(En_key).hexdigest()
FTS = hashlib.md5(FTS_key).hexdigest()
Priority = hashlib.md5(Priority_key).hexdigest()

print('En*.db key :', En[:7])
print('FTS5IndexMicroMsg_encrypt.db key :', FTS[:7])
print('MicroMsgPriority.db key :', Priority[:7])
```

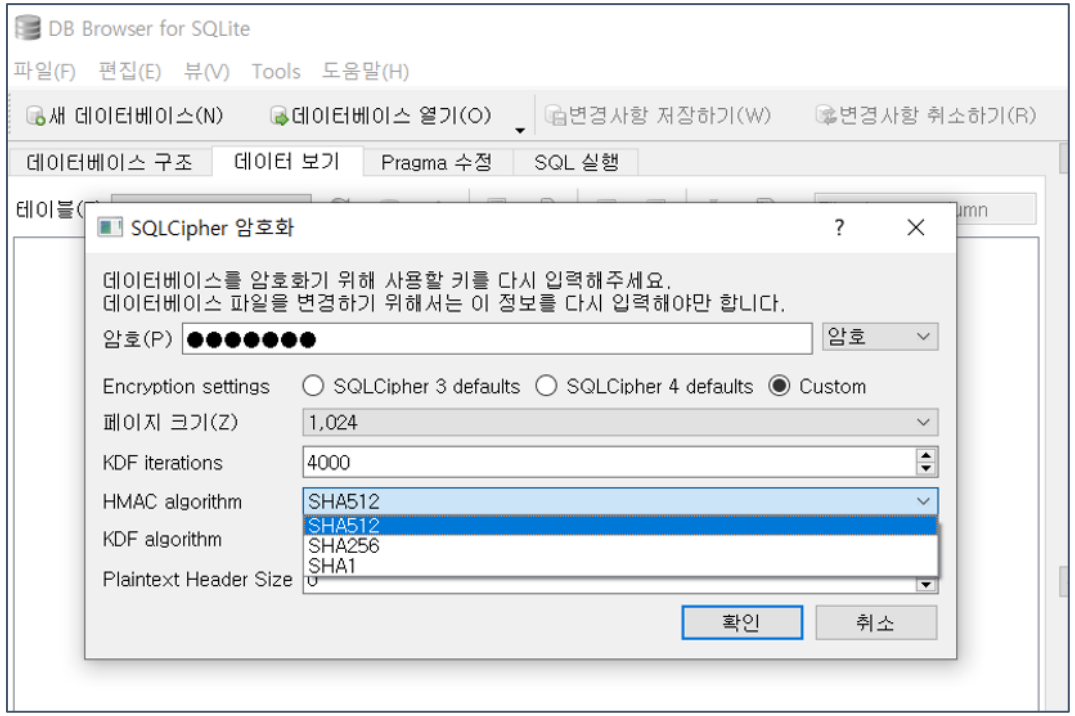
```
En*.db key : 6b2a05c
FTS5IndexMicroMsg_encrypt.db key : a1cb3a8
MicroMsgPriority.db key : fd93892
```

WeChat ‘EnMicroMsg.db’ 복호화

A : EnMicroMsg.db

A
AES-256-CBC
1,024
X
4,000
Left7(MD5(UIN||IMEI))

Size of Page : 1024
HMAC : x
iterator : 4000
key : 6b2a05c



DB Brower for SQLCipher에서는
HMAC이 없도록 선택할 수 없었다.

WeChat 'EnMicroMsg.db' 복호화

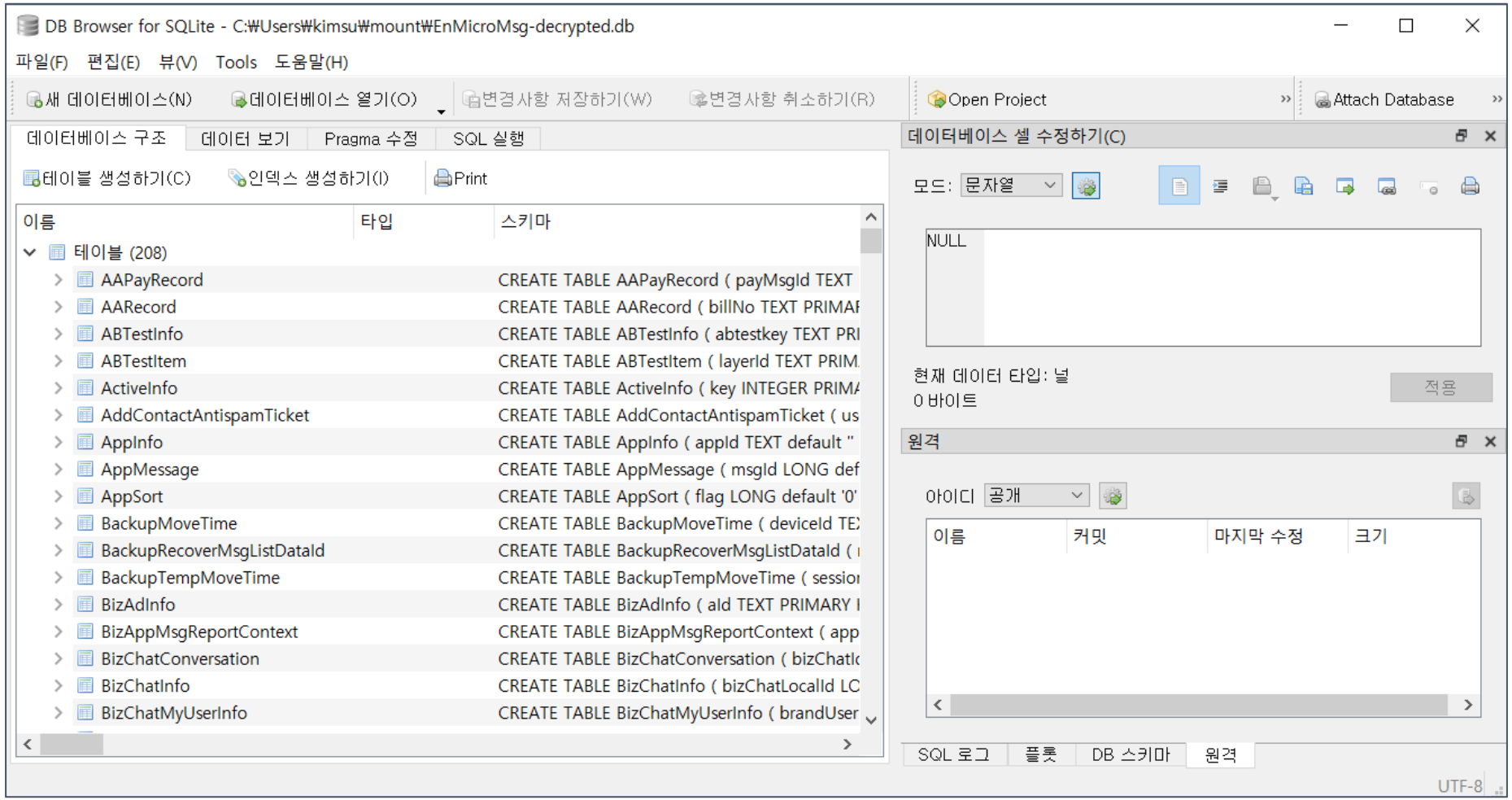
```
from pysqlcipher3 import dbapi2 as sqlite

conn = sqlite.connect( "EnMicroMsg.db" )
c = conn.cursor()
c.execute( "PRAGMA key = '6b2a05c';" )
c.execute( "PRAGMA cipher_use_hmac = OFF;" )
c.execute( "PRAGMA cipher_page_size = 1024;" )
c.execute( "PRAGMA kdf_iter = 4000;" )
c.execute( "ATTACH DATABASE 'EnMicroMsg-decrypted.db' AS wechatdecrypted KEY '';" )
c.execute( "SELECT sqlcipher_export( 'wechatdecrypted' );" )
c.execute( "DETACH DATABASE wechatdecrypted;" )
c.close()
```

Size of Page : 1024
HMAC : x
iterator : 4000
key : 6b2a05c

pysqlcipher3를 이용해 hmac을 off로 설정할 수 있다.
각 매개변수를 입력하면 db가 복호화가 되고,
복호화된 db를 EnMicroMsg-decrypted.db에 저장했다.

WeChat ‘EnMicroMsg.db’ 복호화



EnMicroMsg-decrypted.db

WeChat 'FTS5IndexMicroMsg_encrypt.db' 복호화

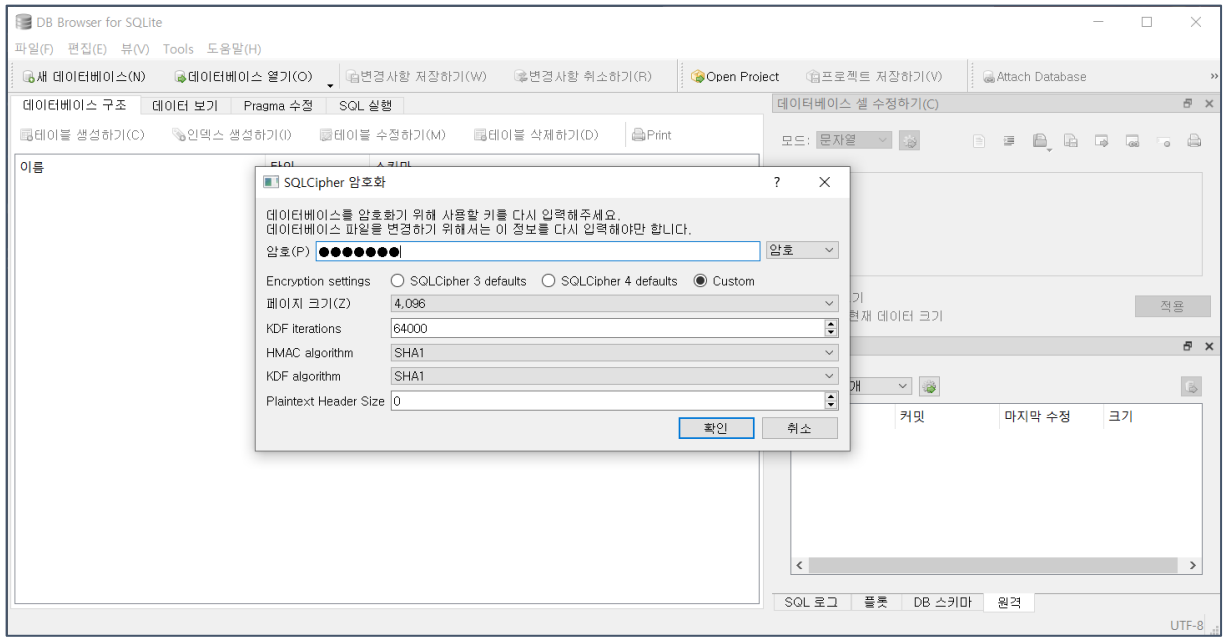
FTS5IndexMicroMsg_encrypt.db

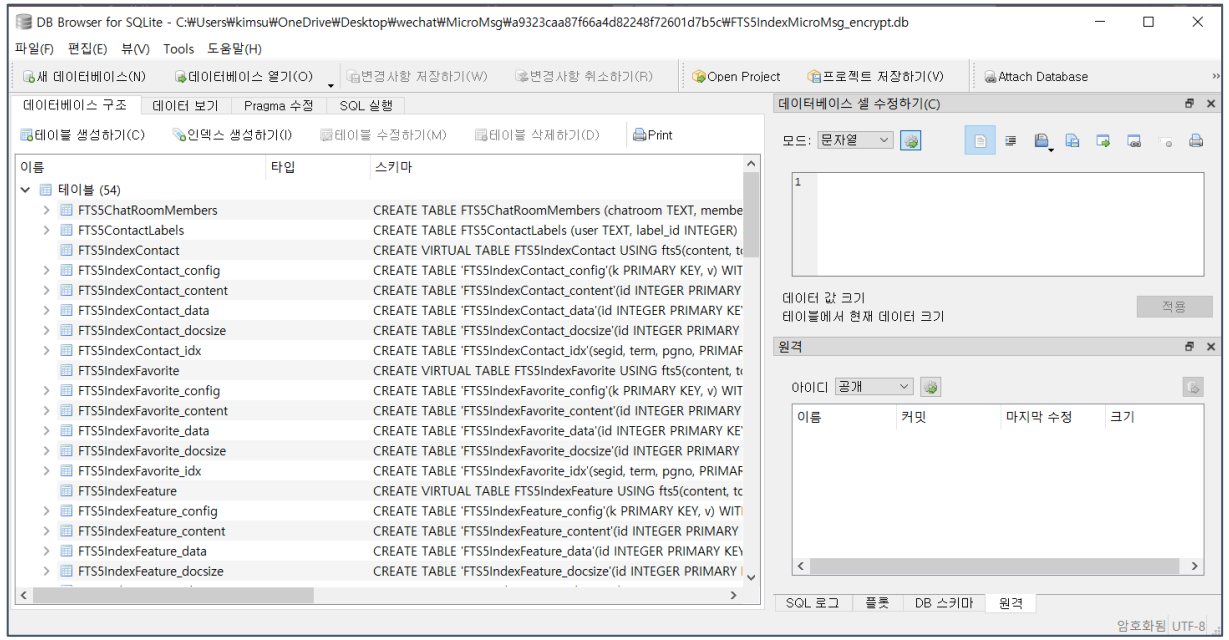
B	
FTS5IndexMicro Msg_encrypt.db	MicroMsg Priority.db
AES-256-CBC	
4,096	
O	
64,000	
FTS5IndexMicro Msg_encrypt.db	MicroMsg Priority.db
Left7(MD5(UIN ||IMEI||Account)	Left7(MD5(UIN|| Account||IMEI)

Target file	Page size	HMAC	KDF	Iteration	PassPhrase
FTS5 Index database	4,096	SHA1	PBKDF2-HMAC-SHA1	64,000	LEFT7 (MD5(UIN||IMEI||WeChat ID))

Size of Page : 4096
HMAC : SHA1
KDF : SHA1
iterator : 64000
key : a1cb3a8

WeChat ‘FTS5IndexMicroMsg_encrypt.db’ 복호화





EnMicroMsg-decrypted.db

WeChat ‘MicroMsgPriority.db’ 복호화

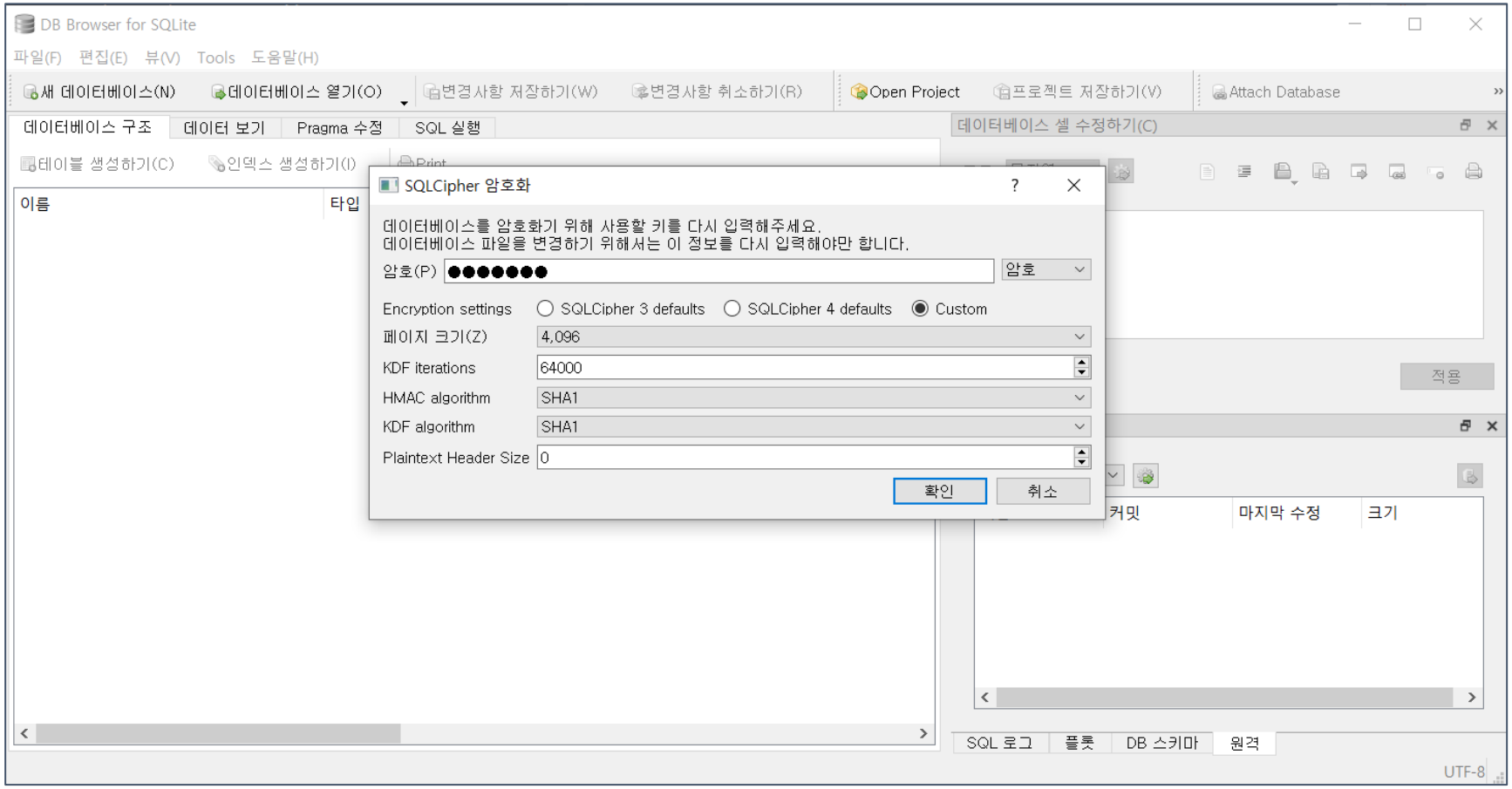
MicroMsgPriority.db

B	
FTS5IndexMicro Msg_encrypt.db	MicroMsg Priority.db
AES-256-CBC	
4,096	
O	
64,000	
FTS5IndexMicro Msg_encrypt.db	MicroMsg Priority.db
Left7(MD5(UIN ||IMEI||Account)	Left7(MD5(UIN|| Account||IMEI)

Size of Page : 4096
HMAC : SHA1
KDF : SHA1
iterator : 64000
key : fd93892

→ FTS5IndexMicroMsg_encrypt.db와 같다.

WeChat ‘MicroMsgPriority.db’ 복호화



→ 복호화 실패

WeChat 사용자 정보

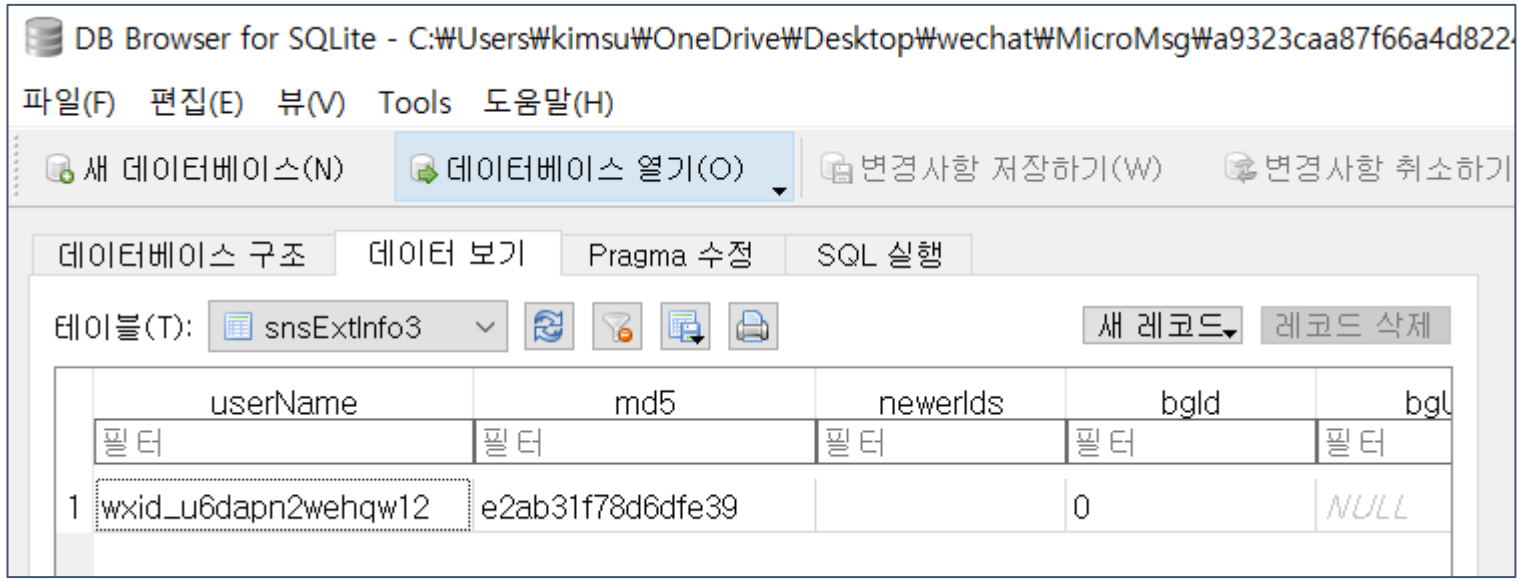
wechat₩shared_prefs₩**com.tencent.mm_preferences.xml**


```xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="last_login_nick_name">김수빈</string>
    <long name="MMTempKeyStepLogger-Last-Clean-Time" value="1614684398051" />
    <string name="reg_last_exit_ui">L200_100</string>
    <boolean name="dynamic_bg_will_crash" value="false" />
    <string name="last_login_use_voice">0</string>
    <string name="last_login_uin">2299112163</string>
    <long name="__MID_LAST_CHECK_TIME__" value="1614684409430" />
    <int name="Main_top_marign" value="96" />
    <string name="__MTA_DEVICE_INFO__">xjgD6zAYj0Q30xaD15xF4HaSIpN46Hm8SnAJpFaX2PhicUBmi+vXgrYOHzb7T3EJQUNOx+dWSUw9ECJDrA7QZwQMI605uJjDULP1ESbz5ukj9qASe9
    <boolean name="dynamic_bg_init_crash" value="false" />
    <string name="login_user_name">+8201041162641</string>
    <long name="check_trim_time" value="1614684396" />
    <string name="__MTA_DEVICE_INFO_CHECK_ENTITY__">{&quot;ts&quot;:1614684409474,&quot;times&quot;:0,&quot;mfreq&quot;:100,&quot;mdays&quot;:3}</string>
    <int name="SP_PERMISSION_HAD_REQUEST_PERMISSION_UID" value="10196" />
    <boolean name="Main_need_read_top_margin" value="false" />
    <string name="last_login_bind_mobile">+821041162641</string>
    <boolean name="SP_PERMISSION_HAD_REQUEST_PERMISSION_STORAGE" value="true" />
    <string name="login_weixin_username">wxid_u6dapn2wehqw12</string>
    <int name="com.tencent.mm.compatible.util.keybord.height" value="1128" />
    <int name="last_reportdevice_clientversion" value="654316867" />
    <int name="last_reportdevice_channel" value="1" />
    <boolean name="Main_ShortCut" value="true" />
    <string name="last_bind_info">4</string>
    <int name="dynamic_bg_init_start_point_count" value="0" />
    <boolean name="login_upload_contacts_already_displayed" value="true" />
    <boolean name="isLogin" value="true" />
```


현 사용자의 WeChat ID, 가입 시 등록한 전화번호와 이름

WeChat 채팅방

wechat₩MicroMsg₩a9323caa87f66a4d82248f72601d7b5c₩**MicroMsg.db**



snsEctInfo3 테이블은 대화 상대방의 WeChat ID 정보 포함
나와의 채팅이어서 내 ID만 있다.

WeChat 채팅방

wechat₩MicroMsg₩a9323caa87f66a4d82248f72601d7b5c₩**FTS5IndexMicroMsg_encrypt.db**

FTS5MetaContact 테이블의 docid 필드

데이터베이스 구조 | 데이터 보기 | Pragma 수정 | SQL 실행

테이블(T): FTS5MetaContact

	docid	type	subtype	entity_id	aux_index	timestamp	status
	필터	필터	필터	필터	필터	필터	필터
1	1	131072	15	17	notifymessage	0	0
2	2	131072	1	17	notifymessage	0	0
3	6	131072	1	21	wxid_u6dapn2wehqw12	1614684444572	0
4	7	131076	15	22	weixin	1614685010931	0
5	8	131076	1	22	weixin	1614685010931	0

FTS5IndexContact_content 테이블의 id 필드

데이터베이스 구조 | 데이터 보기 | Pragma 수정 | SQL

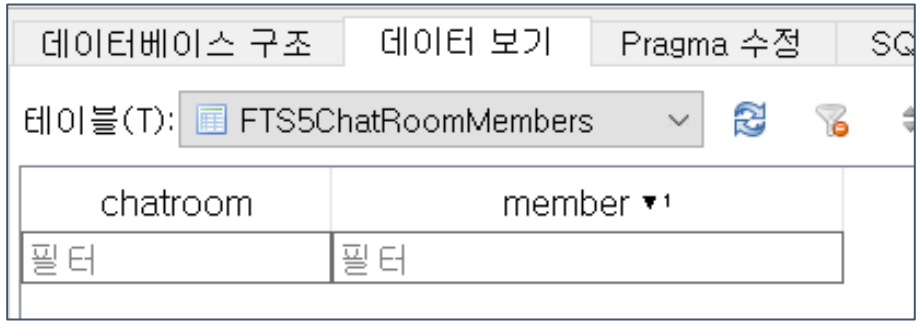
테이블(T): FTS5IndexContact_content

	id	c0
	필터	필터
1	1	notifymessage
2	2	서비스 알림
3	6	김수빈
4	7	WeChat
5	8	WeChat Team

두 필드를 비교해 일대일 대화한 상대방의 WeChat ID를 알 수 있다.

WeChat 채팅방

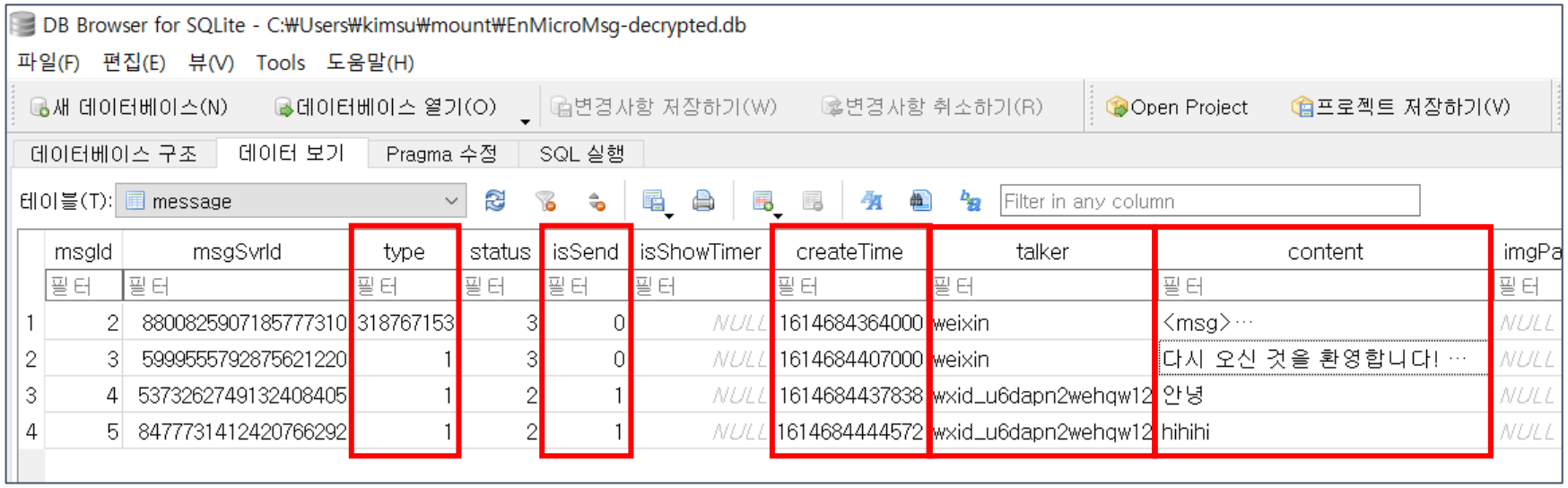
wechat₩MicroMsg₩a9323caa87f66a4d82248f72601d7b5c₩**FTS5IndexMicroMsg_encrypt.db**



단체 채팅방의 경우, FTS5ChatRoomMembers 테이블에
단체 채팅방에 참여한 모든 사람의 WeChat ID가 저장

WeChat 채팅

wechat₩MicroMsg₩a9323caa87f66a4d82248f72601d7b5c₩**EnMicroMsg.db**



	msgId	msgSvrId	type	status	isSend	isShowTimer	createTime	talker	content	imgPa
	필터	필터	필터	필터	필터	필터	필터	필터	필터	필터
1	2	8800825907185777310	318767153	3	0	NULL	1614684364000	weixin	<msg>…	NULL
2	3	5999555792875621220	1	3	0	NULL	1614684407000	weixin	다시 오신 것을 환영합니다! …	NULL
3	4	5373262749132408405	1	2	1	NULL	1614684437838	wxid_u6dapn2wehqw12	안녕	NULL
4	5	8477731412420766292	1	2	1	NULL	1614684444572	wxid_u6dapn2wehqw12	hihihi	NULL

표 4. 유형에 따른 행위 의미
Table 4. Meaning of Action by Type

Type	Meaning of Action
1	Text Message
3	Transition Photo
48	Sharing Location
49	Sharing File
50	Voice Call / Video Call
-1879048186	Sharing Live Location
10000	End Sharing Live Location Session

행위 유형 , 발신 여부 , 수·발신 시각 정보, 대화 상대방의 WeChat ID, 메시지 내용 획득

WeChat 채팅

wechat₩MicroMsg₩a9323caa87f66a4d82248f72601d7b5c₩**FTS5IndexMicroMsg_encrypt.db**

데이터베이스 구조 | 데이터 보기 | Pragma 수정

테이블(T): FTS5IndexMessage_content

	id	c0
	필터	필터
1	1	안녕
2	2	hihihi

데이터베이스 구조 | 데이터 보기 | Pragma 수정 | SQL 실행

테이블(T): FTS5MetaMessage

Filter in any colum

	docid	type	subtype	entity_id	aux_index	timestamp	status	talker
	필터	필터	필터	필터	필터	필터	필터	필터
1	1	65536	41	4	wxid_u6dapn2wehqw12	1614684437838	0	wxid_u6dapn2wehqw12
2	2	65536	41	5	wxid_u6dapn2wehqw12	1614684444572	0	wxid_u6dapn2wehqw12

FTS5IndexMessage_content 테이블에
메시지 내용을 포함

FTSMetaMessage 테이블을 참조하여 메시지의
수·발신자의 WeChat ID와 시간 정보를 획득