Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.4, Aug. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.4.719

애플리케이션 로그를 이용한 공유 자전거 시스템의 잠금장치 해제 방법

조 준 완,[1,2†] 이 지 은,[2] 김 광 조[2‡]
[1]삼성전자, [2]KAIST

Unlocking Shared Bike System by Exploiting an Application Log

Junwan Cho,[1,2†] Jeeun Lee,[2] Kwangjo Kim[2‡]
[1]Samsung Electronics, [2]KAIST

요 약

최근 자동차, 자전거 등 '탈 것'을 공유하는 공유 모빌리티 사업 시장이 점점 커지고 있으며 많은 사업자가 다양한 서비스를 제공하고 있다. 공유 모빌리티의 보안 취약점으로 인해 정상적인 과금을 할 수 없다면 사업자는 사업을 지속 할 수 없으므로 보안에 허점이 있으면 안 된다. 하지만 공유 모빌리티 보안에 대한 인식과 이에 대한 연구는 부족한 상황이다. 본 논문에서는 국내 한 공유 자전거 업체를 선정하여 서비스 애플리케이션 로그에 노출된 보안 취약점을 분석했다. 로그를 통해 자전거 잠금장치의 비밀번호와 AES-128 알고리즘의 암호화 키를 쉽게 얻을 수 있었고, 소프트웨어 역공학 기법을 활용하여 잠금 해제를 위한 데이터 생성 과정을 확인했다. 이를 이용하여 100%의 성공률로 잠금장치를 해제하여 과금 없이 서비스를 사용할 수 있음을 보인다. 이를 통해 공유 모빌리티 사업에서 보안의 중요성을 알리고 새로운 보안 방안이 필요함을 시사한다.

ABSTRACT

Recently, there has been a growing market for shared mobility businesses that share 'transport' such as cars and bikes, and many operators offer a variety of services. However, if the fare can not be charged normally because of security vulnerability, the operator can not continue the business. So there should be no security loopholes. However, there is a lack of awareness and research on shared mobility security. In this paper, we analyzed security vulnerabilities exposed in application log of shared bike service in Korea. We could easily obtain the password of the bike lock and the encryption key of the AES-128 algorithm through the log, and confirmed the data generation process for unlocking using software reverse engineering. It is shown that the service can be used without charge with a success rate of 100%. This implies that the importance of security in shared mobility business and new security measures are needed.

Keywords: Shared bike, Dockless bike sharing system, Analyzing application log, Reverse engineering

I. 서 론

공유 경제가 4차 산업혁명의 대표적인 키워드로 주목받으면서 다양한 공유 플랫폼들이 등장하고 있다. 그 중의 하나가 '탈 것'을 공유하는 공유 모빌리티이다. 최근 공유 모빌리티 시장은 여러 사업자가 다양한 이동 수단을 고객들에게 서비스하며 적극적으로 사업을 펼치고 있다.

공유 자동차의 경우 국내에서 '카셰어링'이라는 이름으로 '쏘카'[1], '그린카'[2] 등의 사업자들이 사업을 진행하고 있다. 자동차를 구매하여 소유하는 것보다 훨씬 적은 비용으로 원하는 장소와 시간에 필요한 기간 만큼만 대여할 수 있기 때문에 사용자들의 호응

Received(04. 17. 2019), Accepted(06. 27. 2019)
† 주저자, junwan.cho@gmail.com
‡ 교신저자, kkj@kaist.ac.kr(Corresponding author)

을 얻고 있으며 사업의 성장 속도도 빨라지고 있는 추세이다. 공유 자전거는 서울시의 '따릉이'[3]처럼 지역 주민들을 위해 지자체가 자체적으로 운영하면서 서비스를 제공하거나 '모바이크'[4]와 같이 스타트업 업체가 지자체와 협약을 맺고 시민들을 대상으로 사업을 진행하는 형태로 운영되고 있다. 또한, 최근 도심 속 편리한 이동 수단으로 떠오르고 있는 전동 킥보드는 '킥고잉'[5] 등의 스타트업 업체들이 서비스를 제공하고 있다.

공유 모빌리티 서비스의 수익 모델은 사업자가 사용자에게 자동차, 자전거 등의 모빌리티를 제공하고 사용자가 해당 모빌리티를 이용할 때마다 사용 시간 혹은 거리에 따라 과금하는 형태다. 만약 정상적인 과금이 이뤄지지 않은 채 서비스가 제공된다면 사업자는 사업을 지속적으로 운영할 수 없을 것이다. 따라서 사업자는 정상 과금 된 사용자만 사용할 수 있도록 적절한 보안 장치를 모빌리티에 적용해야 한다.

본 논문에서는 위에서 언급한 공유 모빌리티 중 가장 접근성이 높은 공유 자전거의 보안에 대해 분석하고자 한다. 국내에서 사업을 운영 중인 공유 자전거 사업자 한 곳(이하 'A'사)을 선정하고 해당 사업자의 서비스 애플리케이션 로그 분석을 통해 100%의 성공률로 과금 없이 서비스를 사용할 수 있음을 보이고자 한다. 이를 통해 공유 모빌리티 사업에서 보안의 중요함을 시사한다.

본 논문의 구성은 다음과 같다. 2장에서는 공유 자전거 서비스의 구성 요소와 통신 방식을 설명한다. 3장에서는 실제로 서비스를 사용하면서 추출한 'A'사의 애플리케이션 로그를 분석한다. 4장에서는 소프트웨어 역공학 기법을 사용하여 잠금장치를 해제하는 데이터의 생성 방법을 설명한다. 5장에서는 3장과 4장에서 확인된 내용을 바탕으로 과금 없이 잠금장치를 해제하는 방법에 대하여 설명한다. 마지막 6장에서는 결론과 향후 계획을 제시하며 본 논문을 마무리한다.

II. 배경지식

2.1 공유 자전거 서비스의 구성 요소 및 용어 설명

- 자전거(B_k): 공유 자전거 서비스의 유형은 크게 두 가지로 나눌 수 있다. 서울시 공유 자전거 '따릉이'와 같이 전용 도킹 스테이션에서 대여와 반납이 이뤄지는 것을 키오스크(kiosk) 방식이라고 한다. 반면, 도크리스(dockless) 방식은 전용 도킹 스테이션이 없으며, 어떤 곳에서든 대여와 반납이 가능한 방식이다. 도크리스 방식에서의 사용자는 전용 애플리케이션을 통해 다른 사용자가 반납한 자전거를 검색하고 자전거가 위치한 장소로 이동하여 대여한다[6]. 본 논문에서 언급하는 공유 자전거의 서비스 방식은 도크리스 방식으로 한정한다.

- 사용자(U_i): 사업자의 서비스에 접속하여 사업자의 서비스 약관에 따라 회사와 이용계약을 체결하고 회사가 제공하는 서비스를 이용한다. 본 논문에서 언급하는 사용자는 안드로이드 스마트폰을 보유하며 서비스 사용을 위해 애플리케이션 스토어에서 전용 애플리케션을 다운로드하여 설치한 상태로 가정한다.

- 사업자: 사용자와 이용계약을 체결하고 사용자에게 자전거 대여 서비스를 제공한다. 본 논문에서는 도크리스 방식의 공유 자전거 서비스를 제공하는 회사로 한정한다. 사업자는 일반적으로 업무 협약을 맺은 지자체(시, 군, 구) 지역 안에서 서비스를 제공하며, 도크리스 자전거 방식의 특성상 사용자가 서비스 제공 지역 밖에서 자전거를 반납하는 경우에는 사용자에게 추가 요금을 징수한다.

- 서버(S): 자전거 대여 서비스 제공을 위해 필요한 사업자의 장비다. 본 논문에서는 사용자의 기본 정보 및 결제 정보, 사업자가 보유하고 있는 모든 자전거의 정보 등을 저장하며, 사용자 스마트폰에 설치된 애플리케이션으로부터의 요청(request)에 대해 응답(response)할 수 있는 데이터 저장소의 의미로 사용한다.

- 잠금장치: 자전거의 도난과 무단 사용을 방지하며 과금을 위해 자전거에 설치된 장치이다. 일반적인 번호입력식 자전거 자물쇠와 유사하게 잠금 상태에서는 자전거의 바퀴가 움직이지 못하게 잠그고 있으며, 잠금장치 해제 데이터를 수신하면 잠금이 해제되어 이동할 수 있게 된다. 도크리스 방식의 공유 자전거 잠금장치에는 사용자에게 자전거의 정확한 위치를 제공

해 줄 수 있는 GPS 모듈, 잠금장치와 사용자의 스마트폰 사이 혹은 사업자의 서버와 자전거와의 통신을 위한 통신 모듈, 잠금장치 자체 충전을 위한 충전 모듈 등을 포함한다. 본 논문에서 언급하는 잠금장치는 블루투스 통신 모듈을 가지고 있는 것으로 한정한다.

- 잠금장치 해제 데이터(unlock data): 자전거의 잠금장치를 해제할 수 있는 데이터를 의미한다. unlock data는 잠금장치의 통신 모듈에 따라 블루투스를 통해 잠금장치로 전달될 수도 있고 셀룰러 네트워크를 통해 전달될 수도 있다. 본 논문에서 언급하는 unlock data는 블루투스를 통해 전달되는 것으로 한정한다.

2.2 도크리스 공유 자전거 서비스의 이용 절차

도크리스 방식의 공유 자전거 서비스를 이용하기 위한 절차는 다음과 같다 (Fig. 1).

1) $U_i \rightarrow S$: 사용자 등록 단계
사용자는 서비스 이용을 위해 전용 애플리케이션을 통해 회원 가입을 한다. 회원 가입 시 사용자는 요금 결제를 위해 일정 금액을 충전하거나 신용카드 정보를 입력해야 한다.

2) U_i: 자전거 위치 확인 단계
자전거 대여를 위해 사용자는 전용 애플리케이션을 실행하여 지도에 표시된 자전거의 위치 정보를 확인하고 가장 가까운 자전거로 이동한다.

3) $U_i \leftrightarrow B_k$: 자전거 번호 확인 단계
자전거를 식별하기 위해 모든 자전거에는 일련번호가 표시되어있다. 사용자는 특정 자전거를 사용하기 위해 애플리케이션을 통해 해당 자전거 번호를 직접 수동으로 입력하거나, 자전거에 부착된 QR코드를 스캔하여 자전거 번호를 자동으로 입력받을 수 있다.

4) $U_i \leftrightarrow S$: 사용 요청 단계
전용 애플리케이션에서는 사용자로부터 입력받은 자전거의 번호와 사용자의 정보를 함께 사업자의 서버로 보낸다. 사업자의 서버에서는 이 데이터를 검토한 뒤 사용자의 결제 수단에 문제가 없고, 선택한 자전거가 정상적으로 사용 가능한 상태라면 잠금장치 해제에 필요한 정보를 애플리케이션으로 보낸다.

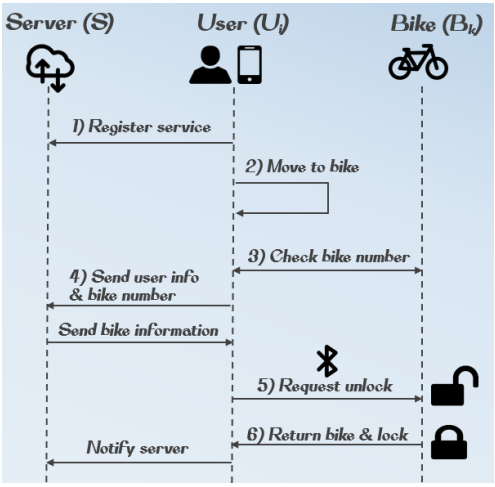

Fig. 1. Dockless bike rental procedure

5) $U_i \rightarrow B_k$: 잠금 해제 단계
애플리케이션에서는 서버가 보낸 자전거 잠금장치 해제에 필요한 정보를 확인한 뒤, 이를 조합하여 잠금 해제를 위한 unlock data를 생성한다. 이후 잠금장치와 블루투스 연결을 통해 unlock data를 발신하게 되면 잠금장치는 이를 수신하여 잠금을 해제한다.

6) $B_k \rightarrow U_i \rightarrow S$: 반납 단계
사용자는 사용이 끝나면 반납하고자 하는 장소에서 열려있는 잠금장치를 닫아 반납한다. 잠금장치는 블루투스 통신을 통해 애플리케이션에 잠금이 이뤄졌다고 알리고 애플리케이션에서는 이를 서버로 업데이트하여 반납됐음을 알린다. 이용 시간과 거리에 따라 일정 금액이 사용자에게 과금 되고 서비스가 마무리된다.

본 논문에서는 서비스 사용 절차 중 4, 5단계에서 발생하는 'A'사의 애플리케이션 로그를 분석한다. 자세한 내용은 3장에서 다룬다.

2.3 도크리스 공유 자전거 잠금장치의 통신 방식

자전거에 설치된 잠금장치를 열기 위해서는 열쇠 역할을 하는 unlock data를 잠금장치로 보내야 한다. 도크리스 공유 자전거 서비스는 전용 거치대가 없기 때문에 무선 통신 모듈을 가지고 있는 잠금장치가 필수적이다. 도크리스 방식 자전거에 설치된 잠금장치의 무선 통신 방식은 크게 두 가지로 나눌 수 있다.

첫 번째는 사업자의 서버에서 자전거로 직접 unlock data를 보내는 방법이다. 이를 위해 자전거의 잠금장치에는 셀룰러 네트워크 연결을 위한 통신 모듈이 설치되어 있고 이를 통해 서버는 스마트폰을 거치지 않고 직접 자전거와 통신하여 unlock data를 보낸다.

두 번째는 블루투스(Bluetooth)를 이용하여 스마트폰에서 자전거로 unlock data를 보내는 방법이다. 블루투스는 근거리 무선 통신 프로토콜로서, BR/EDR과 BLE(Bluetooth Low Energy)로 나뉜다[7]. BLE는 BR/EDR의 소모전력 문제를 해결한 저전력 블루투스이며, 적은 소모전력으로 장기간 이용이 필요한 경우 활용도가 높다. 따라서 도크리스 방식 공유 자전거 잠금장치에서 사용하는 블루투스는 대부분 BLE를 사용한다. 따라서 본 논문에서 언급하는 블루투스의 규격은 BLE로 한정한다.

unlock data 전송을 위해 애플리케이션에서는 자전거 잠금장치로 블루투스 연결을 시도한다. 연결 후 스마트폰과 잠금장치는 블루투스 연결을 통해 unlock data뿐만 아니라 서비스 사용에 필요한 정보를 주고받는다. 따라서 서비스 이용 시간 중에는 사용자 스마트폰의 블루투스는 연결을 유지하기 위해 항상 켜져 있어야 한다. Fig. 2는 앞에서 언급한 이용 절차 중 5단계에서 주고받는 구체적인 블루투스 데이터들의 예시이다.

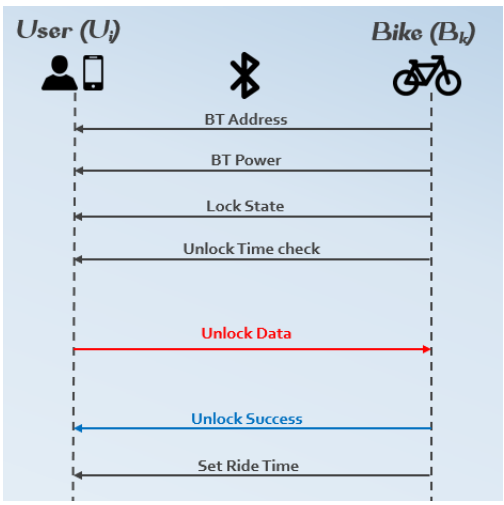

Fig. 2. Send and receive data via Bluetooth

본 논문에서 다루고 있는 'A'사의 자전거는 블루투스 통신을 이용하여 잠금을 해제하며, 대부분의 도크리스 방식의 공유 자전거 사업자는 셀룰러 통신을 이용한 방식보다 블루투스 통신을 이용하여 잠금을 해제하는 방식을 많이 사용하고 있다.

III. 애플리케이션 로그 분석

애플리케이션 로그 분석을 위해 안드로이드 스마트폰에 'A'사에서 제공하는 전용 애플리케이션을 설치하고 회원 가입을 한 뒤 정상적인 과금 환경에서 126번 자전거를 대상으로 대여와 반납을 반복하면서 실험을 진행하였다. 애플리케이션 내부의 동작 확인을 위하여 ADB(Android Debug Bridge)를 사용하여 로그를 추출하였으며, 로그의 내용은 자전거마다 다를 수 있음을 밝혀둔다.

3.1 사용자 인증 단계 로그

Fig. 3의 로그는 사용자 인증 단계에서 볼 수 있는 로그이다. 애플리케이션에서 사용자의 정보를 서버로 전송하면 그 결과로 해당 로그에서 보이는 정보들을 서버로부터 받게 된다. 'user_id', 'user_sn'와 같이 'A'사에서 사용자 관리를 위해 부여한 id 값을 로그를 통해 확인할 수 있다. 문제가 되는 부분은 'mobile'과 'real_name'처럼 이름과 휴대전화 번호가 그대로 로그에 노출된다는 것이다. 또한 'available_deposit', 'credit_point'와 같이 과금과 관련된 정보도 볼 수 있는데, 'card_num' 은 실

07-20 11:47:51.612 10323 11210 11210 I MainActivity:
USER_INFO==={"errorCode":0,"msg":"성공","data":{"user_id":"44[redacted]","user_sn":"80243[redacted]","mobile":"010[redacted]80","nickname":"010[redacted]80","avatar":"","deposit":"0","deposit_state":"1","available_deposit":"900","freeze_deposit":"0","freeze_recharge":"0","credit_point":"219","real_name":"조[redacted]","identification":"","verify_state":"1","available_state":"1","recommend_num":"0","has_month_card":0,"card_num":"[redacted]69","bank_account_owner":null,"bank_code":null,"bank_account_number":null,"required_deposit":"25000","user_state":3,"new_message":0,"new_coupon":0}}

Fig. 3. User verification log

제 사용자가 등록한 카드 번호의 마지막 4개 숫자를 의미한다.[1)]

디버깅 목적의 로그라면 위와 같이 개인 정보가 포함된 로그는 최대한 로그 레벨을 높여 평소에는 출력되지 않게 해야 한다.

3.2 자전거 위치 정보 로그

Fig. 4의 로그는 서버로부터 사용자 근처의 자전거 정보를 얻어온 결과를 보여준다. 위 로그를 보면 사용자는 현재 서울특별시 송파구 잠실 인근에 있으며, 주변에 215번 자전거와 126번 자전거가 있음을 알 수 있다. 특히 근처에 있는 자전거의 위도, 경도 값이 매우 상세하게 나와 있어서 현재 사용자의 위치를 쉽게 가늠해볼 수 있다. 사용자의 위치를 특정할 수 있는 정보 역시 중요한 개인 정보 중의 하나이다. 이러한 정보가 누적된다면 사용자의 이동 경로가 그대로 노출될 수 있다. 위와 같은 위치 정보 역시 로그 레벨을 조정하여 디버깅 시에만 볼 수 있도록 해야 한다.


```
07-20 11:47:51.631 10323 11210 11210 I MainActivity:
GET_BICYCLE_LOCATION_BY_BOUND===
{"errorCode":0,"msg":"success","data":[{"bicycle_id":"833","bicycle_sn":"000215","type":"1","is_hide":"0","fee":"200","last_used_time":"1531965282","region_id":"5","region_name":"잠실 인근","region_city_code":"82","region_city_ranking":"1","lock_sn":"308AE96B0","lat":"37.5229501","lng":"127.1140962","time_unit":"10분","area_code":"08201","is_limit_free":false},{"bicycle_id":"744","bicycle_sn":"000126","type":"1","is_hide":"0","fee":"200","last_used_time":"1532054860","region_id":"5","region_name":"잠실 인근","region_city_code":"82","region_city_ranking":"1","lock_sn":"308AE853C","lat":"37.5224875","lng":"127.1148505","time_unit":"10분","area_code":"08201","is_limit_free":false}]}
```


Fig. 4. Bike location Information log

3.3 잠금장치 해제 시 발생 로그

Fig. 5의 로그는 2.2절에서 언급했던 서비스 이용 절차 중 5단계에서 발생하는 로그다. 사용자 인증이 정상적으로 완료되고, 선택한 자전거가 대여 가능한 상태라면 위와 같은 로그를 볼 수 있다.

로그를 살펴보면 'errorCode'와 'msg'는 서버로


```
07-20 11:47:51.655 10323 11210 11210 I
MainActivity:
OPEN_LOCK===
{"errorCode":0,"msg":"success","data":{"lock_type":5,"mac_address":"3C:A3:08:AE:85:3C","encrypt_key":"bffd9c609b851268","password":"19dfe2","is_scenic":"0","bicycle_type":"1","order_id":22695,"is_limit_free":"0","add_time":1532054893,"bicycle_sn":"000126","lock_sn":"308AE853C","order_sn":"170585398893900433","keep_time":900,"is_month_card":"0","uuid":""}}
```


Fig. 5. Unlocking bike log

부터 문제없이 정상적으로 메시지를 받았고 자전거가 대여 가능한 상태임을 확인해준다. 'mac_address'는 자전거 잠금장치 블루투스 모듈의 물리적 주소이며 'encrypt_key'와 'password'는 잠금 해제와 관련된 내용임을 쉽게 유추할 수 있다.

그 밖에 'order_id'는 대여 요청 건에 대해 서버에서 할당한 고유 번호로 보이며 'bicycle_sn'는 대여하려는 자전거에 할당된 번호인 126번을 나타낸다. 'lock_sn'은 총 48비트의 블루투스 모듈의 주소 'mac_address'의 value 중 처음 12비트를 제외한 36비트가 value로 설정되어 있다.

사용자 인증 결과 로그(Fig. 3) 및 자전거 위치 정보 로그(Fig. 4)와 마찬가지로 잠금장치 해제 시 발생하는 로그도 역시 평문으로 되어있다. 이 중에서 자전거 잠금장치 해제와 연관성이 매우 높아 보이는 'encrypt_key'와 'password'에 대해 구체적으로 살펴볼 필요가 있다.

3.3.1 'encrypt_key'의 의미와 용도

Fig. 5의 로그에서 보이는 'encrypt_key'는 일반적인 보안 알고리즘의 암호화 키(encryption key)의 용도와 마찬가지로 평문으로 된 메시지를 암호화하거나 암호화된 메시지를 복호화할 때 사용되는 것으로 추정할 수 있다. 총 16바이트(128비트) 길이의 'encrypt_key' 는 Table 1과 같이 표현할 수 있다.

Table 1에서 확인할 수 있는 'encrypt_key'의

Table 1. Conversion table of 'encrypt_key'

type	value
ASCII	bffd9c609b851268
Hexdecimal	62 66 66 64 39 63 36 30 39 62 38 35 31 32 36 38
Decimal	98 102 102 100 57 99 54 48 57 98 56 53 49 50 54 56

1) 개인 정보 보호를 위하여 로그 중 일부 내용을 가림.

```
07-20 11:47:51.656 10323 11210 11210 I
MainActivity: key=[98, 102, 102, 100, 57, 99, 54,
48, 57, 98, 56, 53, 49, 50, 54, 56]====mima=[49,
57, 100, 102, 101, 50]
```

Fig. 6. Decimal value of 'encrypt_key'

Decimal value는 Fig. 6의 'key'와 일치한다.

이처럼 'encrypt_key'가 ASCII 및 Decimal의 형태로 로그에 반복적으로 노출되고 있는 것을 확인할 수 있다. 한편, 'encrypt_key'의 용도를 알기 위해 아래 Fig. 7의 로그를 확인해 볼 필요가 있다.

잠금 해제 과정 중 자전거의 잠금장치는 스마트폰으로 여러 번에 걸쳐 데이터를 보내는데 Fig. 7의 로그는 스마트폰이 자전거로부터 첫 번째 데이터를 수신할 때의 로그이다. 'Send Bluetooth Data'는 자전거로부터 수신한 데이터이며 'decryptString'은 자전거가 보낸 'Send Bluetooth Data'를 복호화한 값으로 추측해 볼 수 있다.

Fig. 7의 'Send Bluetooth Data'의 value와 'decryptString'의 value 사이의 관계를 확인하기 위해 AES 알고리즘의 cipher tool을 이용했다[8].

Table 2는 'Input text' 항목에 'Send Bluetooth Data'의 value를 입력하고, Table 1에서 확인한 'encrypt_key'의 Hexdecimal value를 AES-128 알고리즘의 복호화 키로 사용하여 복호화한 결과이다. 복호화 결과인 'Decrypted Text'는 Fig. 7의 'decryptString' 값과 일치한다.

따라서 로그에 노출된 'encrypt_key'는 자전거의 잠금장치와 스마트폰 사이에서 전달 되는 데이터를 암호화 및 복호화하기 위해 사용되는 AES-128 알고리즘의 암호화 키라는 것이 확인되었다.

```
07-20 11:46:56.570 10323 11210 11292 I
BluetoothLeService: Send Bluetooth Data :
3f12ec2a4dd7af39a11f536e1d1d2399
07-20 11:46:56.575 10323 11210 11210 I BLE_INFO:
decryptString=060204A308AE85303030303030303030
07-20 11:46:56.591 10323 11210 11210 I BLE_INFO:
Get Bluetooth Token : a3 08 ae 85
```

Fig. 7. Example of received data from bike

Table 2. Result of decrypting 'Send Bluetooth Data' by using 'encrypt_key' as an encryption key in AES-128

Input Text	3f12ec2a4dd7af39a11f536e1d1d2399
Key	62666664396336303962383531323638
Decrypted Text	060204A308AE85303030303030303030

3.3.2 'password'의 의미와 용도

Fig. 5의 'password'는 자전거의 잠금장치를 해제할 수 있는 비밀번호 역할을 한다는 것을 직관적으로 알 수 있다. 3.3.1절에서 다룬 'encrypt_key'와 같은 방법으로 로그에 있는 'password'의 value '19dfe2'는 Table 3과 같이 표현할 수 있다.

'password'의 Decimal values는 Fig. 6의 'mima'와 일치한다. 따라서 'password'와 'mima'는 같은 목적으로 사용되며, 자전거 잠금장치의 비밀번호로 사용된다고 추정할 수 있다.[2)]

지금까지의 애플리케이션 로그 분석만으로도 unlock data에 필수적으로 포함되는 비밀번호를 알아냈고, unlock data 생성 후 자전거의 잠금장치로 발신하기 전 암호화할 때 쓰이는 AES-128 알고리즘의 암호화 키를 알아낼 수 있었다. 다음 4장에서는 unlock data가 어떻게 만들어지는지에 대해 구체적으로 알아보기로 한다.

Table 3. Conversion table of 'password'

type	value
ASCII	19dfe2
Hexdecimal	31 39 64 66 65 32
Decimal	49 57 100 102 101 50

IV. 애플리케이션 역공학 분석

unlock data가 어떻게 생성되는지 파악하기 위해 'A'사의 애플리케이션이 설치된 안드로이드 스마트폰에서 ADB를 사용하여 apk(android package kit)를 추출한 뒤 소프트웨어 역공학 도구를 사용하여 분석을 진행하였다[9]. 본문에 제시한 코드들은 설명을 위해 꼭 필요한 부분만 일부 발췌한 것임을 밝혀둔다.

4.1 블루투스 토큰 생성 코드

unlock data 생성 코드를 살펴보기 전에 블루투

2) 密碼(mìmǎ)는 중국어로 '암호', '비밀번호'라는 의미다.

스 토큰 생성 과정에 대하여 확인하고자 한다.

앞서 'encrypt_key'의 용도를 알기 위해 자전거의 잠금장치로부터 수신한 첫 번째 데이터를 Fig. 7에서 살펴보았다. 해당 로그의 하단부에는 Fig. 8의 로그도 확인할 수 있다.

```
07-20 11:46:56.591 10323 11210 11210 I BLE_INFO:
Get Bluetooth Token : a3 08 ae 85
```

Fig. 8. Getting Bluetooth token log

RFC 2828에 정의된 보안 토큰의 의미는 암호화 정보를 제공하며 사용자가 제어 가능한 이동 가능한 물리적인 디바이스를 말한다[10]. 하지만 'A'사에서 정의한 '블루투스 토큰'의 의미는 unlock data 생성을 위한 재료의 의미로 보인다.

Fig. 8의 로그로 확인할 수 있는 사실은 자전거의 잠금장치로부터 수신하는 첫 번째 데이터를 통해 잠금장치의 블루투스 주소를 알 수 있으며, 애플리케이션에서는 이를 가공하여 '블루투스 토큰'으로 사용한다는 점이다. 실험대상인 126번 자전거의 경우 'a3 08 ae 85'가 '블루투스 토큰'으로 추출됐다. 아래 Table 4를 보면 잠금장치의 블루투스 주소(Fig. 5의 'mac_address') 맨 처음 1바이트와 가장 마지막 1바이트를 제외한 4바이트를 '블루투스 토큰'으로 사용한다는 것을 알 수 있다.

Table 4. Bluetooth address and token

Bluetooth address	Bluetooth token
3C:**A3:08:AE:85**:3C	**a3 08 ae 85**

Fig. 9는 '블루투스 토큰' 생성 코드이다. 자전거로부터 받은 데이터를 복호화하고, 복호화된 데이터가 만약 '0602'로 시작한다면 해당 데이터의 4번째 바이트부터 7번째 바이트까지를 배열 z에 할당하여 '블루투스 토큰'으로 사용하겠다는 의도이다.[3)]

```
LogUtil.info("BLE_INFO", "decryptString=" + mainActivity.az);[[
if (mainActivity.az.startsWith("0602")) {
    if (a != null && a.length == 16 && a[0] == (byte) 6 && a[1] == (byte) 2) {
        z[0] = a[3];
        z[1] = a[4];
        z[2] = a[5];
        z[3] = a[6];
```

Fig. 9. Code to create Bluetooth token

3) 소프트웨어 역공학으로 확인한 코드이므로 실제 변수명과 차이가 있다.

이 '블루투스 토큰'은 다음절에서 설명할 unlock data 생성을 위한 중요한 재료로 사용된다.

4.2 unlock data 생성 코드

Fig. 10은 잠금장치가 해제될 때 발생하는 로그이다. 'unlock'이라는 용어와 로그가 출력되는 시점으로 추측해보면, 'unlock' 뒤의 내용은 애플리케이션에서 자전거의 잠금장치로 보내는 실제 unlock data라고 생각해 볼 수 있다.

Fig. 11과 Table 5를 참고하면 unlock data가 어떤 방법으로 구성되는지 알 수 있다.

코드를 보면 처음 3바이트인 '5, 1, 6'은 하드 코딩으로 값이 고정되어 있음을 알 수 있다. 4~9번째 바이트는 3.3.2절에서 확인한 'password'의 Decimal 값이며, 10~13번째 바이트는 4.1절에서 확인한 '블루투스 토큰' 값을 byte array로 저장한 뒤 string으로 출력한 값이다. 14~16바이트인 '0, 0, 0'은 하드 코딩된 값이다. 이를 간단히 정리하면 Fig. 12과 같다.

위 코드를 통해 생성된 데이터는 아직 AES-128 알고리즘으로 암호화하기 이전의 데이터이다.

앞서 사용한 AES cipher tool을 사용하여 위 데이터를 Input text로 입력하고 3.3.1절에서 확인한 'encrypt_key'를 암호 키로 사용해서 AES-128 알고리즘으로 암호화하면 Table 6의 결과를 얻을

```
07-20 11:47:57.675 10323 11210 11210 I BLE_INFO:
BLE_UNLOCK   handlerBleSend==2   unlock[5, 1, 6,
49, 57, 100, 102, 101, 50, -93, 8, -82, -123, 0, 0,
0]
```

Fig. 10. Unlock log

```
MainActivity.Q = new byte[]
{(byte) 5, (byte) 1, (byte) 6,

MainActivity.A[0], MainActivity.A[1], MainActivity.A[2], MainActivity.A[3],
MainActivity.A[4], MainActivity.A[5],

MainActivity.z[0], MainActivity.z[1], MainActivity.z[2], MainActivity.z[3],

(byte) 0, (byte) 0, (byte) 0};
LogUtil.info(
    ("BLE_INFO", "BLE_UNLOCK  handlerBleSend==2  unlock" + Arrays.toString(MainActivity.Q));
```

Fig. 11. Code to create unlock data

Table 5. The structure of unlock data

byte	value	description
1~3	5, 1, 6	fixed value
4~9	49, 57, 100, 102, 101, 50	password (mima)
10~13	-93, 8, -82, -123	Bluetooth token
14~16	0, 0, 0	fixed value

‘050106’ + Password(mima) + ‘Bluetooth token’ + ‘000000’

* Password(mima) : 31 39 64 66 65 32 (Hex)
* Bluetooth token : A3 08 AE 85 (Hex)

‘050106’ + ‘31 39 64 66 65 32’ + ‘A3 08 AE 85’ + ‘000000’

➔ 0501063139646666532A308AE85000000 (Hex)

Fig. 12. Example of constructing unlock data

수 있다.

암호화 결과 총 16바이트 길이의 최종 unlock data(Table 6의 Decrypted Text)를 얻었다. 블루투스를 통해 위 unlock data를 자전거로 발신하기만 하면 잠금장치가 해제된다.

다음 장에서는 unlock data를 자전거로 보내는 구체적인 방법에 대하여 설명하도록 한다.

Table 6. Result of encrypting unlock data

Input Text	0501063139646665 32A308AE85000000
Key	62666664396336303962383531323638
Encrypted Text	47204F0B93D748FC57424B18AD9F62D2

V. unlock data 발신 방법

스마트폰에 설치된 ‘A’사의 애플리케이션은 자전거와 블루투스 통신 시 클라이언트 역할을 하고, 자전거의 잠금장치는 서버 역할을 한다. 클라이언트에서는 블루투스 스캔을 하여 장치를 검색하고, 연결되면 서버로 데이터를 write할 수 있으며 서버로부터 데이터를 받을 수도 있다.

unlock data를 발신하기 위해 ‘A’사의 애플리케이션과 동일하게 클라이언트 역할을 할 수 있는 애플리케이션인 ‘nRF Connect for Mobile’을 앱 스토어에서 설치하여 실험을 진행하였다[11].

5.1 잠금장치와 블루투스 연결

애플리케이션을 실행시켜 블루투스 스캔 후 검색된 자전거와 연결하면 Fig. 13의 화면을 볼 수 있다. 목록에서 잠금장치가 가지고 있는 서비스를 확인할 수 있는데, 블루투스 사양에 정의된 규격화된 서비스를 제외하면 두 개의 커스텀 서비스(unknown service)가 있는 것을 확인할 수 있다. 이는 블루투

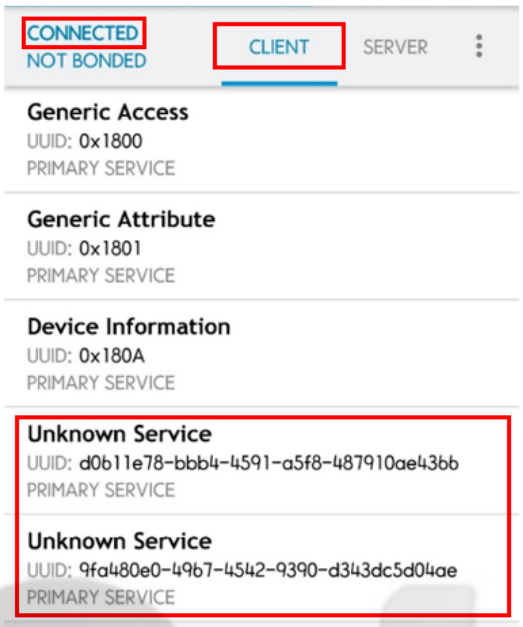

Fig. 13. Connecting to bike and Checking custom service

스 규격에 정의되어있지 않는 서비스를 의미하고 장치 제조사에서 추가한 서비스이다. 따라서 잠금장치 제조사마다 목록의 내용은 다를 수 있다. 서버(잠금장치)는 이 커스텀 서비스를 이용하여 클라이언트(애플리케이션)에 write 권한을 부여하고 클라이언트가 서버로 직접 데이터를 write 할 수 있게 만들 수 있다.

정리하면, 애플리케이션은 잠금장치가 제공하는 커스텀 서비스를 이용하여 unlock data를 write 함으로서 잠금장치로 전달한다.

5.2 unlock data 쓰기

두 가지의 커스텀 서비스 중 ‘9fa480e0’의 상세정보를 보면 properties에 write 권한이 있는 것을 Fig. 14를 통해 알 수 있다. 서버는 해당 서비스를 통해 클라이언트 애플리케이션이 서버로 데이터를 write 할 수 있는 권한을 부여한다는 의미이다.

write 버튼을 누르면 Fig. 15의 팝업 화면이 발생하고, ‘write value’ 항목에 Table 6에서 확인했던 암호화된 unlock data를 입력한 뒤 send 버튼을 누르면 자전거의 잠금장치가 풀리게 된다.

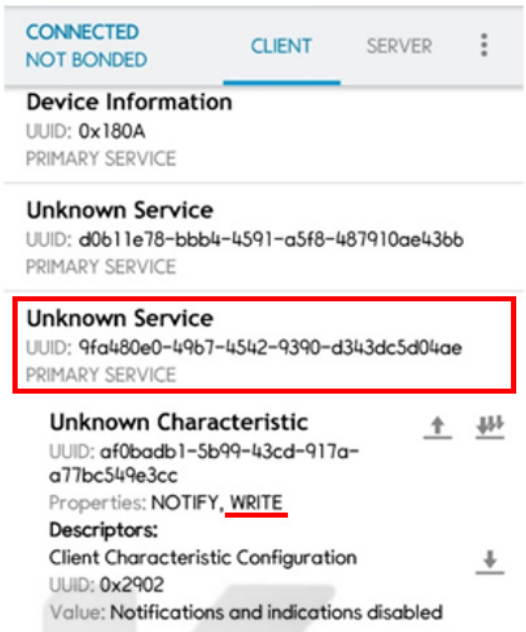

Fig. 14. Checking write permission

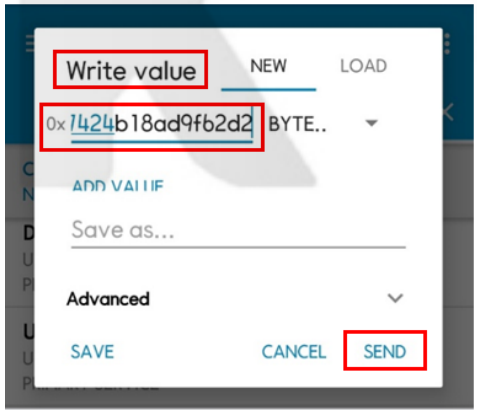

Fig. 15. Writing unlock data

VI. 결론 및 향후 계획

본 논문에서는 공유 자전거 사업자인 'A'사의 서비스 애플리케이션 로그로 사용자의 전화번호 및 위치 정보가 평문으로 노출되어 개인 정보 침해의 위험이 있음을 확인하였다. 또한, 애플리케이션의 로그 분석을 통해 자전거 잠금장치의 비밀번호 및 데이터 전송 시 사용되는 AES-128 알고리즘의 암호화 키를 알아냈고, 소프트웨어 역공학 기법을 통해 unlock data 생성 과정을 파악했다. 최종적으로 암호화 된 unlock data를 앱 스토어에서 쉽게 구할 수 있는 앱을 이용하여 자전거로 발신하면 잠금장치가 해제되는 것을 확인했다. 본 논문에서 제시한 방법은 특정 번호의 자전거에 한하지 않고 'A'사의 모든 자전거에 적용할 수 있는 방법이다.

'A'사 공유 자전거의 잠금장치는 자전거마다 항상 고정된 암호화 키와 unlock data를 사용하기 때문에 특정 자전거의 unlock data를 알아내면 반복적으로 과금 없이 사용할 수 있다. 따라서 사용 후 반납이 이뤄질 때마다 암호화 키와 unlock data를 변경할 필요성이 있다.

향후 후속 연구를 통해 'A'사의 공유 자전거 외에 다른 공유 자전거 사업자와 자동차, 전동 킥보드 등 다른 공유 모빌리티 사업자의 보안에도 유사한 문제가 있는지 점검하고, 새로운 보안 방법을 제안할 예정이다.

본 논문에서 다룬 모든 분석내용은 2018년 8월 8일 'A'사에 통보하였음을 알려둔다.

References

[1] SOCAR, "socar" https://www.socar.kr/, Apr. 2019.
[2] Green Car, "green car" https://www.greencar.co.kr/, Apr. 2019.
[3] Seoul Bike, "seoul bike" https://www.bikeseoul.com/, Apr. 2019.
[4] Mobike, "mobike" https://mobike.com/kr/, Apr. 2019.
[5] KICKGOING, "kickgoing" https://kickgoing.io/, Apr. 2019.
[6] Mengting Wang, "Share Bike Use of Chinese Consumers," Master Thesis, Seoul National University, Feb. 2018.
[7] Bluetooth SIG. Bluetooth Core Specification Version 5.1 [Internet]. Available: https://www.bluetooth.com/ko-kr/specifications/bluetooth-core-specification/, Apr. 2019.
[8] Online Domain Tools, "online domain tools" http://aes.online-domain-tools.com/, Apr. 2019.
[9] APK Decompile Online, "apk decompile online" https://www.apkdecompilers.

com/, Apr. 2019.
[10] IETF, "Internet Security Glossary," https://tools.ietf.org/html/rfc2828, Apr. 2019.
[11] Google Play, "nrf connect for mobile" https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=ko, Apr. 2019.

〈저자소개〉

조 준 완 (Junwan Cho) 정회원
2009년 2월: 동국대학교 컴퓨터공학과 졸업
2009년 3월~현재: 삼성전자 무선사업부 책임 연구원
2018년 3월~현재: KAIST 전산학부 소프트웨어대학원 석사과정
〈관심분야〉 정보보호, 통신공학

이 지 은 (Jeeun Lee) 학생회원
2013년 2월: KAIST 물리학과 졸업
2013년 3월~현재: KAIST 전산학부 석박사통합과정
〈관심분야〉 포스트 양자 암호, 양자 암호, 안전성 증명

김 광 조 (Kwangjo Kim) 종신회원
1980년 2월: 연세대학교 전자공학과 졸업
1983년 8월: 연세대학교 전자공학과 석사
1991년 3월: 요코하마 국립대 전자공학과 박사
1998년 1월~2006년 8월: 한국정보통신대학교 정보공학부 교수
2000년 1월~2004년 12월: 세계암호학회(IACR) 상임이사(BoD)
2005년 3월~2005년 12월: MIT/UCSD 방문교수
2006년 8월~2009년 2월: 한국정보통신대학교 공학부장
2009년 1월~2009년 12월: 한국정보보호학회 회장
2009년 3월~현재: KAIST 전산학부 정보보호대학원 교수
2012년 1월~2012년 8월: UAE 칼리파대학 방문 교수
현재: IFIP TC-11 한국 대표, 세계암호학회(IACR) 석학회원(Fellow) 및 선정위원, 한국정보보호학회 명예회장, MDPI Cryptography 주편집자, IEEE Trans. on Dependable and Secure Computing 부편집자, IEICE Trans. on Fundamentals 부편집자, ASIACRYPT2020 General Chair 등
〈관심분야〉 암호 및 정보보호 이론과 응용