아주대학교 Apple Wallet 모바일 학생증

BLE와의 연계를 통한 보안 강화형 정적(Static) 방식의 QR Pass 모바일 신분 증명 시스템 제작

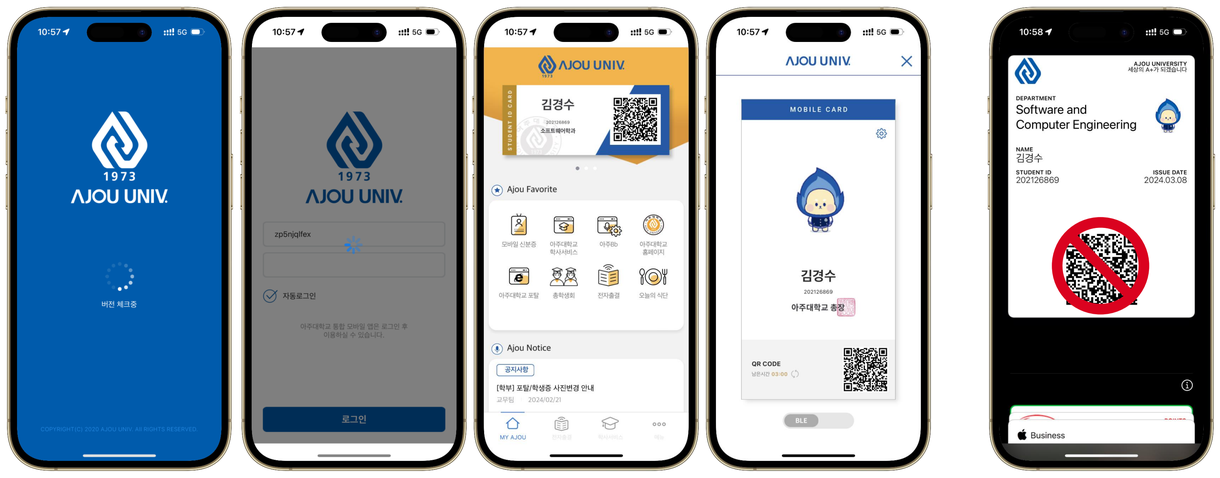
2024-1 AJOU SOFTCON

이 름 김경수 **지도교수** 강경란 교수님

프로젝트 연구 배경과 기술 아이디어 기획

기존 동적(Dynamic) 방식의 QR Pass와 그 한계점

동적 방식의 QR Pass는 일정한 유효기간을 가진 임시 QR코드를 모바일 신분 증명에 활용하는 방식입니다. 이를 통해 QR코드의 복제 및 유출에 따른 보안 문제를 방지합니다. 아주대학교의 경우에는 통합 어플리케이션을 통해 180초 동안 유효한 동적 방식의 QR코드 학생증을 스마트폰 화면에 띄워 건물 출입이나 신분 증명에 사용합니다. 그러나 이 방식은 인증시에 매번 QR코드가 변경되어야 하므로 QR코드를 발급하는 전산 서버의 부담이 크고 매 인증마다 사용자가 반드시 모바일 앱을 실행해야만 합니다. 특히, QR코드가 매번 새로 바뀌기 때문에 인증을 위해 기다리는 시간이 길어지고 QR Pass를 Apple Wallet과 같은 앱에 추가하여 사용하기 어렵습니다.



동적 방식의 QR Pass 모바일 신분 증명은 매 인증 시 마다 사용자가 별도의 어플리케이션을 직접 실행해야 하므로 Apple Wallet과 같은 앱에 추가하기 어렵습니다.

정적(Static) 방식의 QR Pass에 사용자의 Bluetooth ID를 교차 인증하여 새로고침 없이 Pass의 위·변조를 검증

정적(Static) 방식의 QR Pass는 기존 동적 방식과는 반대로 QR코드를 매번 바꾸지 않고, 이전에 발급받았던 QR코드를 그대로 신분 인증에 활용하는 것이 핵심입니다. 이는 사용자가 사용하는 모바일 디바이스의 고유한 Bluetooth ID를 미리 전산 서버에 등록해두고, 인식기가 QR코드를 감지하면 사전에 등록된 사용자의 Bluetooth ID가 근처에 존재하는지를 교차 인증합니다. 이를 통해 매 시간마다 코드를 변경하지 않고도 QR Pass의 진위 여부를 빠르고 안전하게 검증할 수 있습니다.

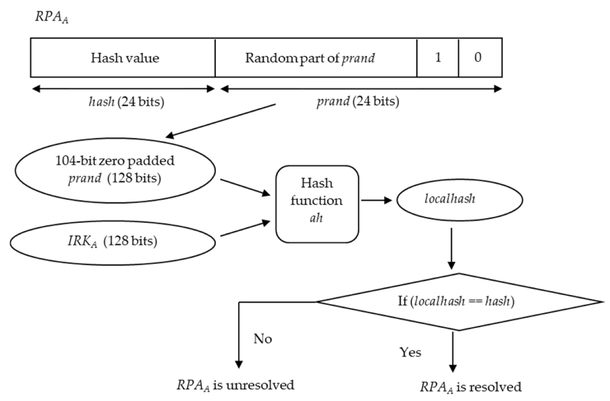
	동적(Dynamic) 방식의 QR Pass	정적(Static) 방식의 QR Pass
방식	매 인증시마다 새 QR코드 발급하여 검증	기존 QR코드와 Bluetooth ID를 교차 검증
인증 속도	느림	빠름
전산 처리	많음	적음
Apple Wallet 추가 가능 여부	불가능	가능

기존에 발급받았던 QR Pass를 즉시 불러와 신분을 인증합니다. 이 때, QR코드 이미지와 함께 사용자의 Bluetooth ID를 교차로 인증합니다.

프로젝트 연구 진행 주요 내용과 결과

디바이스의 Bluetooth ID가 랜덤으로 바뀌어도 사용자 모바일 디바이스의 근접 여부를 확인

대부분의 스마트폰은 주변 Bluetooth 네트워크에 자신의 디바이스를 홍보할 때 고정된 ID를 알리지 않고 15분에 한 번씩 자신의 Bluetooth ID를 랜덤으로 바꿉니다. 그러나 스마트폰의 Bluetooth ID가 매번 변경되어도, 한번 페어링을 진행했던 디바이스는 별도의 추가 절차 없이 매번 자동으로 연결됩니다. 본 프로젝트는 이 원리를 적극 활용하였습니다. Bluetooth 페어링 단계 시에 각 디바이스들은 서로 AES-128bit 암호화로 인증 가능한 비밀 키를 주고 받습니다. 이를 통해 Bluetooth 디바이스의 ID가 매번 변경되어도 근처에 특정 디바이스가 근접해 있는지 여부를 검증할 수 있습니다.




```python
from Crypto.Cipher import AES
import binascii

# IRK와 RPA를 정의합니다.
irk_hex = "45966f53116397456b476b2e16e4dbd8"
rpa_hex = "643b1fd44086".replace(__old: ':', __new: '')

# IRK와 RPA를 바이트 배열로 변환합니다.
irk = binascii.unhexlify(irk_hex)
rpa = binascii.unhexlify(rpa_hex)

# RPA에서 prand와 hash를 분리합니다.
prand = rpa[:3]
rpa_hash = rpa[3:]

# prand 앞에 0을 붙여서 16바이트로 만듭니다. BLE 규격에 따라 prand는 24비트이고, 나머지는 8으로 채웁니다.
padding = b'\x00' * 13
prand_padded = padding + prand

# AES-128 ECM 암호화를 수행합니다.
cipher = AES.new(irk, AES.MODE_ECB)
encrypted = cipher.encrypt(prand_padded)

# 암호화된 결과의 하위 3바이트를 추출하고, RPA의 hash와 비교합니다.
calculated_hash = encrypted[-3:]

if calculated_hash == rpa_hash:
    print("RPA는 해당 IRK에 의해 생성된 것입니다.")
else:
    print("RPA는 해당 IRK에 의해 생성되지 않았습니다.")
```

실현 가능성 검증을 위한 입·출입 통제 시스템의 동작 과정 분석 및 적용

본 프로젝트에서 제안한 기술을 실제 환경에 적용하고 아이디어의 실현 가능성을 검토하는 것은 매우 중요한 과정이라 할 수 있습니다. 이번 프로젝트에서는 단순한 기술 제안을 넘어서, 이 기술이 현재 아주대학교에서 도입해 사용 중인 입·출입 통제 시스템과 호환 가능하도록 프로토타입을 직접 구현하였습니다. 이를 위해 S사의 입·출입 통제 하드웨어가 사용되었습니다. 실제 입·출입 통제장치를 가져와 시스템의 동작 메커니즘을 하드웨어단에서 분석하고, 이를 별도의 임베디드 보드에서 모조하여 QR Pass 인식기가 카드리더기와 같은 별도의 인식기 역할을 할 수 있도록 적용하였습니다.

* Apple Wallet is trademarks of Apple Inc., registered in the U.S. and other countries and regions.