

NFC를 이용한 PC 간편 보안 솔루션

2020. 10. 23

중부대학교 정보보호학과

지도교수 : 김민수 교수님

1조: 한수연, 이연재, 강승구
도지현, 정재훈



목 차

- 조원 편성
- 주제 선정
- 구 상 도
- 추진 경과
- 개발 환경 및 개발 내용
- 개발 시스템 운영
- 결론 및 기대 효과

조원 편성

이름	역할	비고
한수연 (조장)	웹서버 구축 화면 잠금부 개발, ppt 및 보고서 작성	
이연재	웹서버 구축 화면 잠금부 개발, ppt 및 보고서 작성	
강승구	발표	
도지현	파이썬 소프트웨어 개발	취업(2020.6)
정재훈	화면 잠금부 개발	입대(2020.8)

주제 선정

NXP, 차량용 디지털 키 솔루션 공개

SE·NFC 기술 적용된 CCC 릴리즈 2 준수

권봉석 기자 | 입력: 2020/03/11 12:05 | 반도체 · 디스플레이



NXP 반도체가 SE와 NFC 칩셋 기반 자동차용 디지털 키

NXP 반도체가 SE(시큐어 엘리먼트)와 NFC 칩셋 기반 자동차용 디

이 솔루션은 전세계 주요 자동차 업체와 스마트폰 제조사 등이 승
리스 2를 준수한다.

NFC를 지원하는 스마트폰이나 스마트키, NFC 스마트카드를 자동

[테크칼럼] 스마트폰 이용한 모바일카드, 출입통제 새로운 방향 제시

| 입력: 2020-03-28 13:21



#슈프리마 #최성빈 #모바일카드 #출입통제

2019년과는 다른 2020년 모바일카드 시장 성장 기대

호환성 문제와 속도 문제, 리더보급문제 등 기존 문제점 속속 해결되고 있어

[보안뉴스=최성빈 슈프리마 기술연구소 상무] 아이폰과 안드로이드로부터 시작된 스마트폰의 출현은 전 세계적으로 IT 산업 지형을 급격하게 변화시키고 있다. 스마트폰이 출현하면서 디지털 카메라, 차량용 네비게이션, MP3, PNP 등의 산업은 이미 사라진 지 오래다. 스마트폰은 여러 장치가 가지고 있었던 각각의 기능을 하나의 기기로 통합시켜 현대의 편리함이라는 기본적인 욕구를 충족시켰다. 하지만 스마트폰이 다른 기기보다 경쟁력을 가질 수 있었던 이유는 네트워크 기반 서비스와 결합하며 이전에 누리지 못한 편리함과 새로운 가치를 제공하고 있기 때문이다.



가장 많이 본 기사 [주간]

- 1 코로나19로 뜬 화상화
- 2 인기 높은 줌, 요즘 사
- 3 [간급] 'n번방 회원 신
- 4 [단독] 부동산 중개 프
- 5 모든 버전의 윈도우에
- 6 드디어 사이버 공간에
- 7 2018년부터 시작된
- 8 '상여금 청구' 메일로
- 9 'n번방'으로 문제된 사
- 10 비너스락커 조직 단골
- 11 수천 개의 안드로이드

“SNS에서도 보안



취약점 경고 및 버그리포트

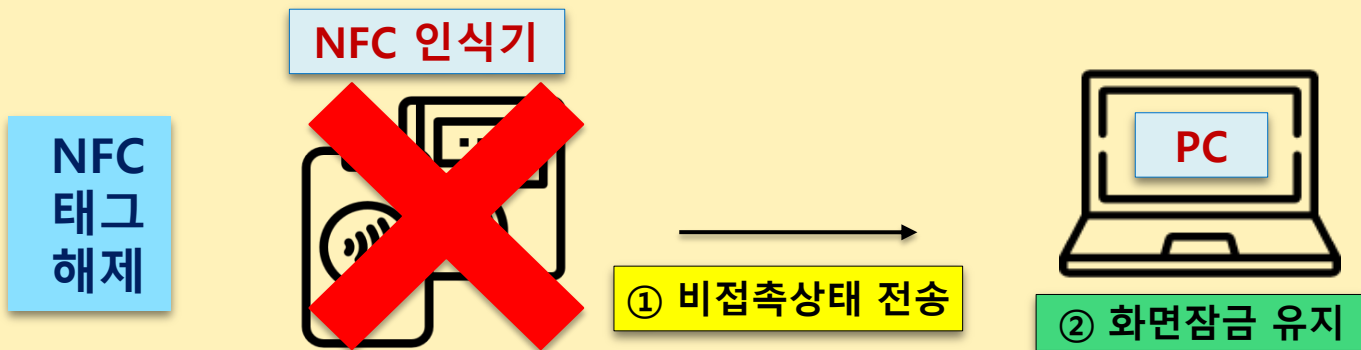
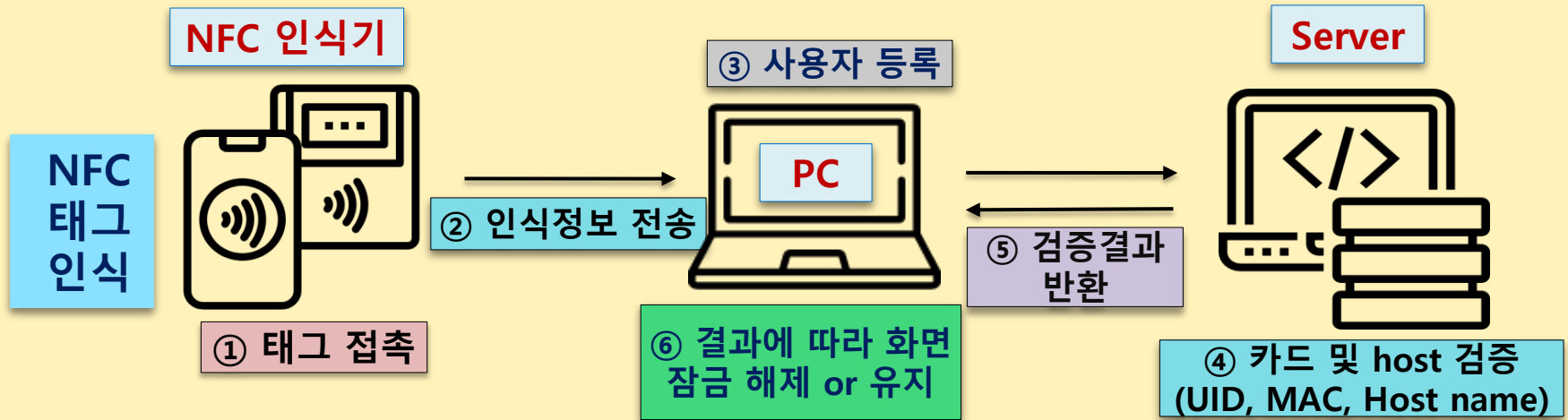
- [2020-04-08] [간급]
- [2020-04-08] 경찰청
- [2020-04-07] 2020
- [2020-04-07] '연예인
- [2020-04-07] 620만

보안 · IT산업 동향

- 모비젠, 클라우드 기반
- 시큐리온, 이성권 대표
- 링크플로우, 72억 원

NFC를 이용한 PC 간편 보안 솔루션 개발

구상도



추진 경과

추진기간(2020년) 수행 업무	3월	4월	5월	6월	7월	8월	9월	10월
	자료 조사 및 연구	■						
NFC 인식 기술 개발		■						
화면 잠금 기술 개발			■					
SSL 암호화 및 웹 서버 관리자 페이지 구축						■		
시스템 검증 및 보완							■	

개발 환경 및 개발 내용(1/9)

개발 환경

Server

Backend: Node.js

Server: CentOS 7

DB

MongoDB

Software

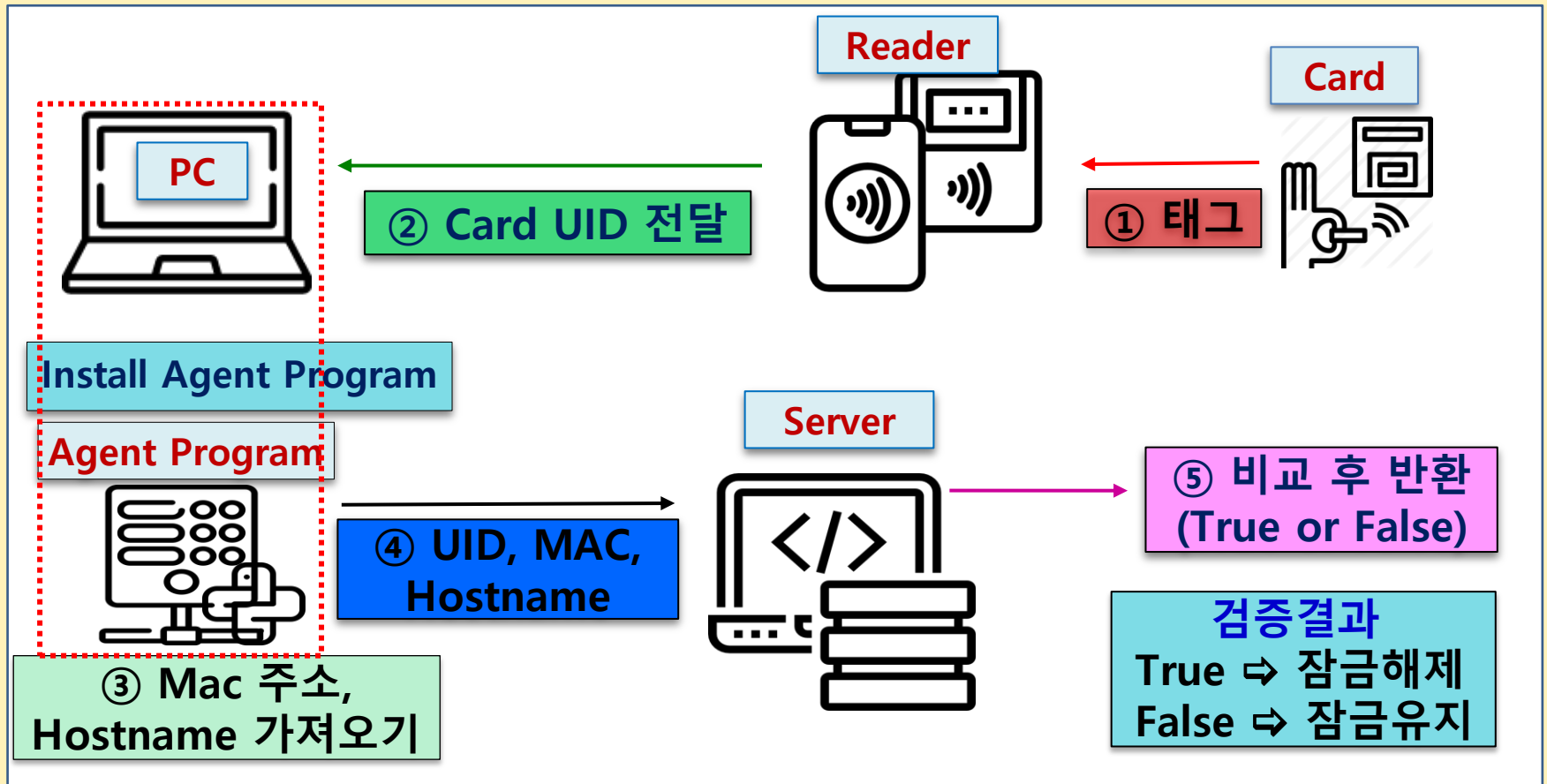
Language: Python

Module: pycard

개발 환경 및 개발 내용(2/9)

개발 내용(1/8)

작동 체계

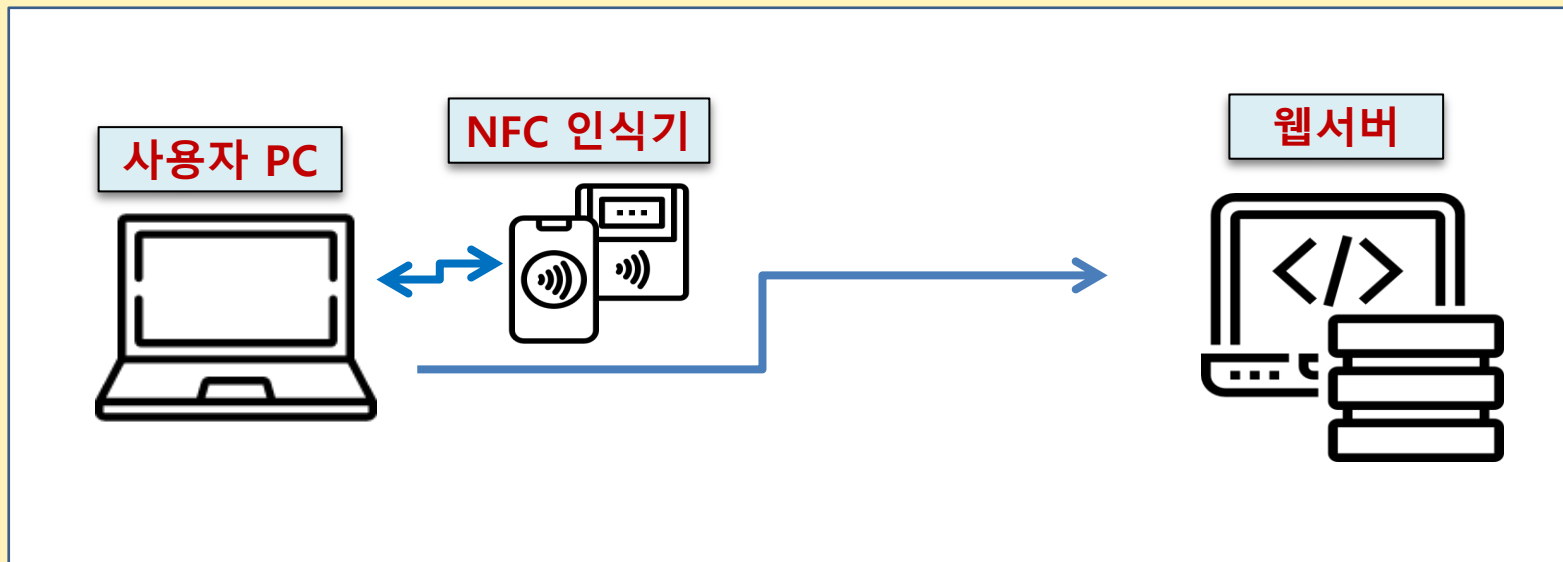


개발 환경 및 개발 내용(3/9)

개발 내용(2/8)

□ 웹서버 구축

- CentOS + Node.js 환경으로 서버 구축
- 가입회원 인증 및 회원 등록/삭제 API 개발



개발 환경 및 개발 내용(4/9)

개발 내용(3/8)

□ 회원 관리

회원등록 Prog

```
router.post("/uidadd", (req, res, next) => {
  const parsedata = {
    uid: req.body.uid,
    mac: req.body.mac,
    username: req.body.username,
    hostname: req.body.hostname,
  };

  Posttest1.getUserByUid(parsedata.uid, (err, user) => {
    if (!user) {
      Posttest1.add(parsedata);
      return res.json({
        success: true,
        msg: "신규 UID : " + parsedata.uid + "
        되었습니다.",
      });
    } else {
      return res.json({ success: false, msg:
      등록된 사용자입니다." });
    }
  });
});
```

```
router.post("/uidrm", function (req, res, next) {
```

회원삭제 Prog

회원인증 Prog

```
router.post('/postest', (req, res, next) => {
  const parsedata = {
    'uid': req.body.uid,
    'mac': req.body.mac,
    'hostname': req.body.hostname
  };

  if (req.body.uid == undefined || req.body.mac == undefined || req.body.hostname == undefined) {
    return res.json({ success: false, msg: 'Fail' });
  }

  Posttest.getUser(parsedata.uid, parsedata.mac, parsedata.hostname, (err, user) => {
    if (!user) {
      console.log("matching fail");
      return res.json({ success: false, msg: 'Compare Fail' });
    }
  });
});
```

회원 등록/삭제 및 가입회원 인증체제 구현

개발 환경 및 개발 내용(5/9)

개발 내용(4/8)

□ 관리자 페이지

선택 관리자: 명령 프롬프트

https 포팅(보안 강화)

```
C:\Users\dlakd\sooyeonjae>heroku login
heroku: Press any key to open up the browser to login or q to e
Opening browser to https://cli-auth.heroku.com/auth/cli/browser
BQAAAAACZAAEZGFOYWAAAAOMjlyLjEwOS42Mi4yNDFkAAZzaWduZWVudWZWRuBgBCGeTM
Logging in... done
Logged in as @naver.com

C:\Users\dlakd\sooyeonjae>git init
Reinitialized existing Git repository in C:/Users/dlakd/sooyeon

C:\Users\dlakd\sooyeonjae>heroku git:remote -a sooyeonjae
set git remote heroku to https://git.heroku.com/sooyeonjae.git
```

회원 목록 frontend

```
<h2 class="page-head">회원 목록 frontend</h2>
<table class="table table-striped">
  <thead class="thead-dark">
    <tr>
      <th>uid</th>
      <th>mac</th>
      <th>hostname</th>
      <th>username</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let user of users">
      <td>{{ user.uid }}</td>
      <td>{{ user.mac }}</td>
      <td>{{ user.hostname }}</td>
      <td>{{ user.username }}</td>
    </tr>
  </tbody>
</table>
```

회원 목록 호출

```
router.get("/testest", (req, res) => {
  Postest1.getAll((err, users) => {
    if (err) throw err;
    res.json(users);
  });
});
```

등록 회원 카드 목록을 호출하는 관리자 페이지 구현

개발 환경 및 개발 내용(6/9)

개발 내용(5/8)

□ NFC 인식 기술 개발

```
from smartcard.System import readers
from smartcard.util import toHexString
from smartcard.
from smartcard.
```

모듈 호출

카드인식 Prog

```
def work():
    r = readers()

    if len(r) < 1:
        print ("error: No readers available!")
        sys.exit()
    else :
        reader = r[0]
        print ("Using: ", reader)

    connection = reader.createConnection()

    while(True):
        try:
            connection.connect()
            cmdMap = {
                "getuid":[0xFF, 0xCA, 0x00, 0x00, 0x00]
            }
            data, sw1, sw2 = connection.transmit( cmdMap['getuid'] )
            uid = toHexString(data)
            checkUid(uid)
            sleep(1)
        except:
            print("No Card")
            window.deiconify()
```

Python 'pyscard' 모듈을 사용하여 NFC 데이터 획득/활용

개발 환경 및 개발 내용(7/9)

개발 내용(6/8)

□ 화면 잠금 기술 개발

- 파이썬 tkinter(GUI) , pyautogui(키보드제어) 모듈 활용
- 서버에서 MAC, Hostname, UID가 존재하는지 확인
- 이미지파일을 전체화면, 항상 위 설정 후 Key disable을 통해 화면 잠금
- 강제 종료할 수 있는 단축키(Esc, Alt-F4, 작업관리자)를 막아 보안 강화

WM_DELETE_WINDOW



```
window.bind('<Key>', deny_none_alnu)  
window.protocol("WM_DELETE_WINDOW", do_exit)
```

프로그램의 창이 닫힐 때 발생하는 윈도우 메시지를 감지



키보드가 눌러있으면 무시 & 아무 행동도 하지 않게 함

개발 환경 및 개발 내용(8/9)

개발 내용(7/8)

□ 화면 잠금 유지/해제

```
import tkinter as tk
from tkinter import *
import string
from PIL import Image
import pyautogui
```

```
#global variable
window = tk.Tk()
window.title("LOCKER")
window.wm_
key_is_pres
```

잠금화면 출력부

```
#User Data
hostname = socket.gethostname()
mac = psutil.net_if_addrs()
mac = mac['이더넷'][0][1]
```

```
wall = Phot
```

```
def checkUid(uid):
```

```
global window
```

```
params = {"uid": uid , "mac": mac, "hostname": hostname }
```

```
hed = {'Content-Type': 'application/json'}
```

```
res = requests.request('post', 'http://sooyeonjae.com/users/postest', header
print(res.text)
```

```
resjson = json.loads(res.text)
```

잠금 해제 Prog

```
Postest1.getUser(
    parsedata.uid,
    parsedata.mac,
    parsedata.hostname,
    (err, user) => {
        if (!user) {
            console.log("matching fail");
            return res.json({ success: false, msg: "Compare
            Fail" });
        } else {
            console.log("matching success");
```

회원인증 Prog

```
len(r) < 1:
    print ("error: No
    sys.exit()
se :
    reader = r[0]
    print ("Using: ", reader)
```

잠금 유지 Prog

전체 잠금화면으로 설정 후 서버 3종 검증을 통해 잠금 유지/해제

개발 환경 및 개발 내용(9/9)

개발 내용(8/8)

□ 강제 종료 막기(보안통제)

```
t1 = Thread(target=work)
t1.start()

window.attributes('-fullscreen', True)
window.bind('<Key>', deny_none_alnu)
window.protocol("WM_DELETE_WINDOW", do_exit)
window.mainloop()
t1.join()
```

강제 종료 막기 및 화면잠금 Prog

```
def do_exit(event):
    global window
    global key_is_pressed
    if key_is_pressed:
        print("No")
        key_is_pressed = False

def deny_none_alnu(event):
    if event.char in string.printable:
        print(event, "denied!!!")
        key_is_pressed = True
```

```
<KeyPress event state=0x40000 keysym=Win_L keycode=91 x=911 y=388> denied!!!
<KeyPress event keysym=Control_L keycode=17 x=1163 y=403> denied!!!
<KeyPress event state=Control keysym=Alt_L keycode=18 x=1163 y=403> denied!!!
<KeyPress event state=Control|0x20000 keysym=Alt_L keycode=18 x=1163 y=403> denied!!!
```

강제 종료할 수 있는 단축키(Esc, Alt-F4, 작업관리자)를 막아 보안 강화

개발 시스템 운영(1/6)

사용자 등록/삭제

The image displays three overlapping screenshots of a Python 3.7.9 Shell window, illustrating the process of user registration and deletion. Each screenshot is labeled with a green box: '초기 화면' (Initial Screen), '사용자 등록 화면' (User Registration Screen), and '사용자 삭제 화면' (User Deletion Screen).

초기 화면 (Initial Screen): Shows the Python 3.7.9 Shell interface with the prompt 'Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32'. The prompt is followed by 'Type "help" for more information.' and '>>>'. The output shows '=====
or rm.py ==
Using: ACS ACR122U PICC Interface 0
UID : CC 96 F3 01
Status words: 90 00
Status: The operation completed successfully.'

사용자 등록 화면 (User Registration Screen): Shows the same shell interface. The user enters '1.등록' (1. Register) at the prompt. The output shows '=====
or rm.py ==
Using: ACS ACR122U PICC Interface 0
UID : CC 96 F3 01
Status words: 90 00
Status: The operation completed successfully.'

사용자 삭제 화면 (User Deletion Screen): Shows the same shell interface. The user enters '2.삭제' (2. Delete) at the prompt. The output shows '=====
or rm.py ==
Using: ACS ACR122U PICC Interface 0
UID : CC 96 F3 01
Status words: 90 00
Status: The operation completed successfully.'

On the left side of the screenshots, there are three red-bordered boxes containing the following text:

- 1.등록
- 2.삭제
- 기능 선택 :

These boxes are positioned to show the user's input corresponding to the actions being performed in the shell.

개발 시스템 운영(2/6)

초기 잠금 화면

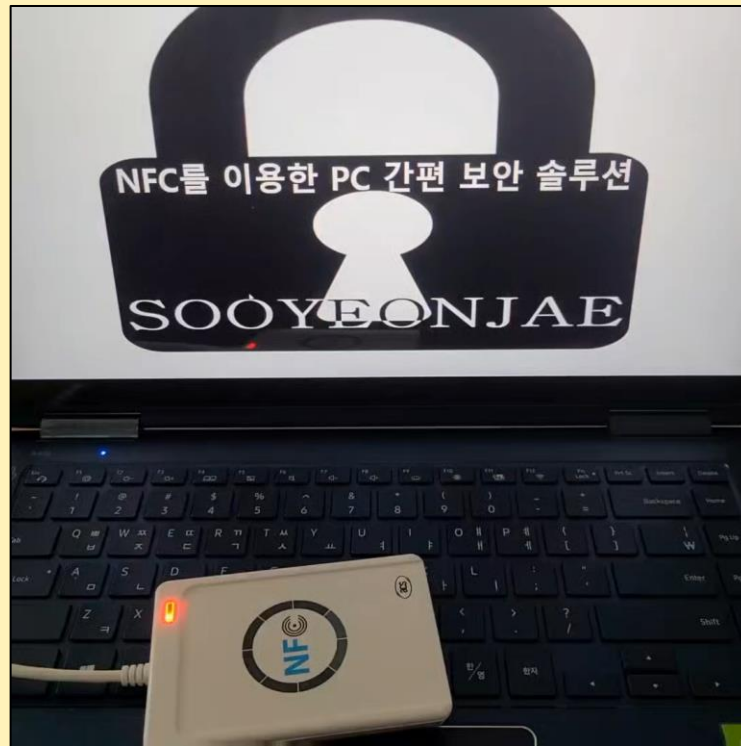


개발 시스템의 초기 화면 잠금 상태 로고

개발 시스템 운영(3/6)

화면 잠금 해제

□ 접촉 시범(등록자)



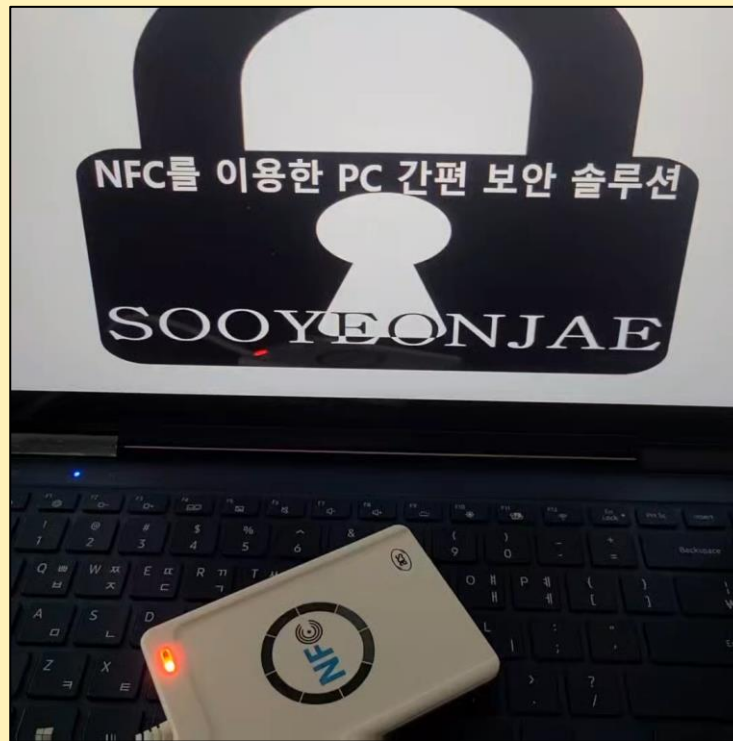
등록 사용자

카드 인식 및 등록된 사용자를 확인 후 화면 잠금 해제

개발 시스템 운영(4/6)

화면 잠금 유지

□ 접촉 시범(미등록자)



미등록 사용자

카드 인식 및 미등록된 사용자일 경우 화면 잠금 상태 유지

개발 시스템 운영(5/6)

사용자 인증 결과(콘솔창)

```
*Python 3.7.9 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\dlak\Desktop\정보보호프로젝트설계실습2\아카이브\200618
finally.py =====
Using: ACS ACR122U PICC Interface 0
{"success":false,"msg":"Compare Fail"}
```

미등록 사용자

```
node -- 회원인증 화면
dojihyeon@dojihyeon-ui-MacBookAir node % node app.js
Server started on port 80!
Connected to database mongodb://sooyeonjae:1whghkd1x1d@ds361998.mla
sooyeonjae
matching succes
matching fail
```

```
Shell*
Debug Options Window Help
(tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit
in32
copyright", "credits" or "license()" for more information.
: C:\Users\dlak\Desktop\정보보호프로젝트설계실습2\아카이브\200618
====
CR122U PICC Interface 0
ie,"msg":"등록된 사용자 입니다."}
```

등록 사용자

카드의 인식 및 등록된 사용자 여부를 확인 후 인증 결과 출력

개발 시스템 운영(6/6)

관리자 페이지

The screenshot shows a web browser with three tabs. The top tab is the main page (메인 페이지) at https://sooyeonjae.herokuapp.com. The middle tab is the admin login page (관리자 로그인 페이지) at sooyeonjae.herokuapp.com/login. The bottom tab is the admin page (등록 회원 리스트) at sooyeonjae.herokuapp.com/testtest. The admin page displays a table of registered cards with columns for uid, mac, hostname, and username.

uid	mac	hostname	username
2C 6B FB 01	f8:63:3f:14:c9:e0	윤잉	yony
CC 96 F3 01	3c:f0:11:75:4a:f4	DESKTOP-YJL	yeonjae
3C 76 EB 01	3c:f0:11:75:4a:f4	DESKTOP-YJL	antlee

관리자 페이지에서 등록된 카드 및 사용자 확인 관리

결론 및 기대 효과

□ 결 론

- 간편 로그인 방식과 NFC 카드 태그를 활용, 기존의 비밀번호 입력방식 보다 보안이 강화된 PC 간편 보안 솔루션을 개발하는데 성공
- 자료 수집 및 연구 등 사전 철저한 준비와 팀원간 역할 분담으로 차질 없이 시스템을 개발

※ 인원 축소 등에도 불구하고 보안요건을 충족하는 시스템 개발에 최선

□ 기대 효과

- 미등록 사용자의 출입 방지 등 외적 위험요소를 일차적으로 예방하면서 보안성이 향상된 시스템으로 활용 기대
- NFC 인식 기술 등 기술의 이해도를 증진시키면서 프로젝트를 진행, 실무 기술 역량을 배양하는 계기. 끝.

Q & A

감사합니다