한국사물인터넷학회논문지 Vol. 2, No. 3, pp. 25-30, 2016 http://dx.doi.org/10.20465/KIOTS.2016.2.3.025

NFC와 OPT를 활용한 아두이노 다중 인증 시스템

이범진, 김주성, 양기열, 양우범, 최창원*
한신대학교 컴퓨터공학부

Arduino Multi-Authentication System using NFC and OTP

BeomJin Lee, JuSeong Kim, GiYeol Yang, WooBeom Yang, Changwon Choi*
Division of Computer Engineering, Hanshin University

요 약 기존의 NFC 인증 시스템의 문제점은 고정된 킷값 사용 시 발생하는 보안적인 문제들과 NFC 인증 시스템에 별도의 NFC 태그를 사용해야 하는 불편함이었다. 본 논문에서는 이러한 문제점을 해결하기 위해 기존의 킷값을 발생하는 알고리즘에 OTP를 병행하여 처리하도록 하여 보안성을 높였고, 통합된 안드로이드 어플리케이션을 개발하여 사용자의 편의성을 높였다. 또한 이러한 인증 기술을 아두이노에 적용하여 향후 사물인터넷(IoT) 환경의 보안 기술과 인증 시스템 기술의 활용 방향을 제시하였다.

주제어 : 사물인터넷, 인증 시스템, OTP, 아두이노, NFC

Abstract The issued problem in the existed NFC authentication system is the security problem caused by the fixed key and the inconvenience occurred by the extra NFC tag. This paper aims to enhance the security level through OTP system added up the key generation mechanism and the convenience level by the integrated Android App. This proposed authentication system on Arduino will be widely applied to IoT security and will be used on diverse applications.

Key Words : IoT, Authentication system, OTP, Arduino, NFC

1. 서론

1.1 연구 배경 및 목적

IT 분야에서 보안 이슈는 과거부터 현재까지 계속 중요하게 여겨져 왔으며. 정보는 자산이기 때문에 그에 맞는 위험 관리가 대두되어왔다. 하지만 IT는 발전과 적용이 빠른 분야이기 때문에 보안 기술이 증대 됨에도 불구하고 다양한 공격에 의해 개인 정보들이 누출되곤 한다. 특히 사물인터넷 환경에서의 보안 관리는 더욱 중요시되고 있다. 이에 대한 원인으로 고정된 키를 사용하는 기존의 인증 시스템을 들 수 있다. 최근 많은 분야에서 활용되고 있는 NFC인 경우도 고정 킷값을 이용한 응용이며 보안성에 대한 위험 요소를 내포하고 있다. 따라서 보안성이 강화된 시스템이 사물인터넷 환경에서 더욱 필요한 상황이 되었다[1].

1.2 연구 내용

본 논문에서는 이를 해결하기 위해 고정된 키를 사용하는 인증 시스템이 아닌 OTP(One Time Password) 기능으로 다중 인증하여 사용자에게 안전성과 효율성을

본 논문은 한신대학교 2016년도 2학기 컴퓨터공학부 종합설계2 프로젝트 결과물입니다.
*교신저자 : 최창원(won@hs.ac.kr)
접수일 2016년 10월 12일 심사완료 2016년 11월 2일

제공한다. 이를 위해 시간 기반의 OTP를 사용하여 시간이 경과됨에 따라 OTP의 값이 수시로 바뀌도록 설계하였다. 이와 같은 OTP와 NFC의 기능을 결합하여 편의성과 보안성을 가지는 다중 인증 시스템을 개발하였다.

본 논문은 1장에서 연구의 목적과 내용을 기술하고 2장에서 관련 기술에 대한 내용을 간략히 설명한다. NFC와 OTP를 이용한 다중 인증 시스템의 설계를 3장에서 구체적으로 제시하며 개발된 결과의 보안성 강화와 활용분야를 4장에서 기술하고 5장에서 본 논문의 결론을 기술한다.

2. 관련 기술

2.1 NFC(Near Field Communication)

근거리 무선 통신(Near Field Communication, NFC)은 전자태그(Radio Frequency IDentification, RFID) 기술 중 하나로 13.56MHz의 주파수 대역을 사용하는 비접촉식 통신 기술이다. 2003년에 ISO/IEC 표준으로서 승인되었다. 10cm 이내의 아주 가까운 거리의 무선 통신을 하기 위한 기술로 2016년 현재 지원되는 데이터 통신 속도는 424Kbps다.

통신거리가 짧기 때문에 상대적으로 보안이 우수하고 가격이 저렴하여 차세대 근거리 통신 기술로서 주목받고 있다. 블루투스나 지그비 등 경쟁기술에 비해 보안성과 편의성이 뛰어나다. 데이터 읽기와 쓰기 기능을 모두 사용할 수 있으며 교통요금, 티켓요금, 슈퍼마켓이나 일반 상점 등에서의 요금 지불뿐만 아니라 여행 정보 전송이나 출입통제 잠금장치 등 일상생활의 여러 서비스에서 사용할 수 있어 적용 범위가 광범위하다는 것 역시 장점이다.

2.2 OTP(One Time Password)

OTP를 이용하기 전에는 사용자 아이디 및 암호를 이용한 방식이 주를 이루었고 대부분 한번 인증 값이 유출되고 나면 인증 값을 바꾸기 전에는 해킹을 당하기 쉽다는 문제점이 있다. 이를 보완하기 위한 인증 수단으로 보안카드가 있지만, 이 역시 정해진 수십여 개의 번호를 사용할 뿐이므로 여전히 유출에 의한 위험성을 가지고 있다. 이에 따라 한 번 생성되면 한 번에 한해서만 유효한 OTP 방식을 도입하게 되었다.

OTP는 무작위로 생성되는 난수의 일회용 패스워드를 이용하는 사용자 인증 방식이다. 보안을 강화하기 위하여 도입한 시스템으로, 로그인 할 때마다 일회성 패스워드를 생성하여 동일한 패스워드가 반복해서 사용됨으로 발생하는 보안상의 취약점을 극복하기 위해 도입되었다.

본 연구에서는 시간과 암호화 해쉬 함수를 결합하여 일회용 비밀번호를 생성하는 시간 동기화 방식을 사용하였다. 같은 시간에 동일한 비밀 키로 생성한 키는 같은 값을 갖는다. 임의의 입력 값이 필요하지 않다는 점에서 사용이 간편하고 클라이언트가 서버와 통신해야 하는 횟수가 비교적 적다[3][4]. 또한 클라이언트에서는 시각 정보를 이용해 OTP를 생성하므로 스마트폰 등의 모바일 기기도 클라이언트로 사용되기 적합하다는 점 역시 비용 절감 적인 측면에서도 장점이다.

2.3 안드로이드(Android)

안드로이드는 리눅스(Linux)란 오픈소스 운영체제의 소스코드를 기반으로 개발된 모바일 디바이스용 운영체제로써, 개발자는 JAVA를 이용하여 응용프로그램을 개발할 수 있으며, 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다.

또한 안드로이드 소프트웨어 개발 키드(SDK)를 통해 응용 프로그램을 개발하기 위해 필요한 각종 도구들과 API를 제공한다.

안드로이드는 리눅스 커널 위에서 동작하며, 다양한 안드로이드 시스템 구성 요소에서 사용되는 C/C++ 라이브러리들을 포함하고 있다. 안드로이드는 기존의 자바 가상 머신과는 다른 가상 머신인 달빅 가상 머신을 통해 자바로 작성된 응용 프로그램을 별도의 프로세스에서 실행하는 구조로 되어 있다.

2.4 Arduino

아두이노는 2005년 이탈리아의 IDII(Interaction Design Institutelvera)에서 하드웨어에 익숙지 않은 학생들이 자신들의 디자인 작품을 손쉽게 제어할 수 있도록 하기 위해 고안되었다[5].

아두이노는 오픈 소스를 지향하는 마이크로 컨트롤러(micro controller)를 내장한 기기 제어용 기판. 컴퓨터 메인보드의 단순 버전으로 다수의 스위치나 센서로부터 값을 받아들여 LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어 낼

수 있다. 자유 소프트웨어 운동에서 출발한 오픈 소스라는 개념을 하드웨어 부문까지 확산시킨 것이다.

3. 시스템 설계

3.1 시스템 구성

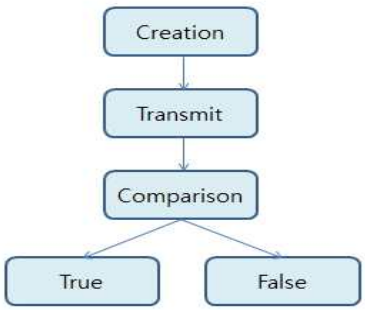
시스템 구성은 [그림 3.1]과 같이 OTP를 생성 하는 안드로이드 폰과 서버, 안드로이드에서 서버로 값을 전송하는 중간 전송자의 역할과 결과를 표시해주는 아두이노로 구성되어있다.

아두이노는 NFC Shield, WIFI Shield, 서보모터 등의 모듈로 구성되어 안드로이드 단말기에서 생성된 OTP 토큰을 NFC Shield로 받아 WIFI Shield를 통해 무선통신으로 서버에 전송을 한다. 또 서보모터의 동작으로 결과값을 표현한다. 안드로이드 단말기의 NFC 기능을 이용하여 생성된 OTP 토큰을 아두이노로 전송을 한다.

서버는 서버자체의 알고리즘을 통해 생성된 OTP를 저장하기 위한 오라클 데이터베이스가 구성되어 있다.

[그림 3.1] 시스템 구성
[Fig. 3.1] System Components

3.2 시스템 흐름도



[그림 3.2] 시스템 흐름도
[Fig. 3.2] System Flow

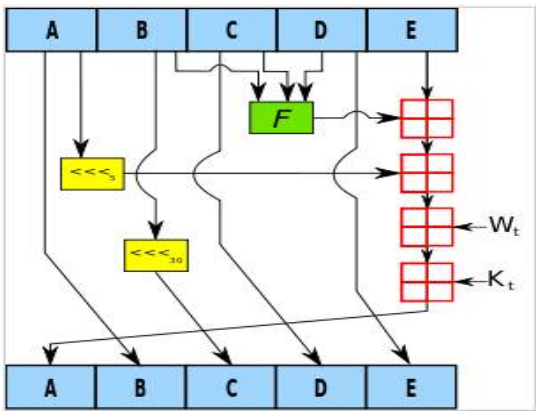
[그림 3.2]는 NFC와 OTP를 활용한 다중 인증시스템의 데이터 처리 흐름을 나타낸다. 사용자는 안드로이드 폰에서 T-OTP로 생성된 6자리의 OTP를 이용하여 아두이노에 NFC 태깅을 한다. 아두이노는 NFC Shield를 통해 받은 OTP를 다시 한 번 WIFI Shield를 통해 서버로 전송을 한다. 서버에서는 서버에서 생성된 T-OTP와 비교하여 인증 여부를 판단하고 성공 또는 실패 값을 다시 아두이노로 전송하여 결과를 처리하게 되는 시스템이다.

3.3 T-OTP 로직

[그림 3.3] T-OTP 생성 로직
[Fig. 3.3] T-OTP Generation Logic

시간을 기반으로 한 T-OTP(Time Based One Time Password)로 OTP를 생성하였고 HMAC SHA-1 알고리즘을 이용하여 구현하였다. 안드로이드, 서버에서 각각 동일한 알고리즘을 이용하여 OTP를 생성하였기 때문에 같은 시간에 동일한 OTP가 생성된다.

[그림 3.4]는 SHA-1 압축 함수가 블록 하나를 처리하는 과정을 나타낸다. A, B, C, D, E는 각각 32비트 내부 상태이고, F는 계속 변하는 비선형 함수이며, Kt는 상수이다. 왼쪽 회전n은 n비트만큼 왼쪽으로 회전하는 연산이고, 덧셈은 232 모듈로 덧셈을 나타낸다.



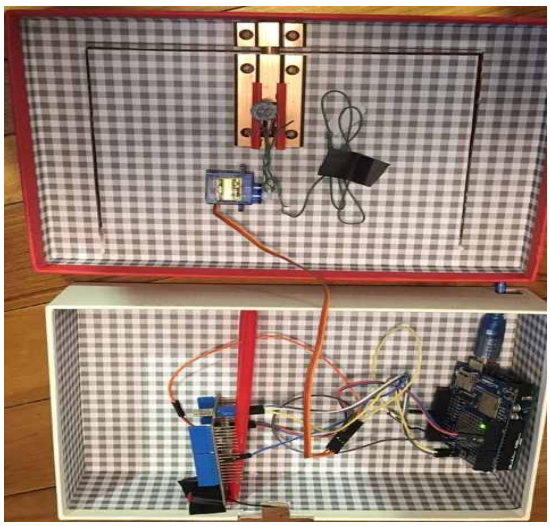
[그림 3.4] SHA-1 처리 과정
[Fig. 3.4] SHA-1 Processing

4. 구현

4.1 개발 환경

안드로이드 Application 개발 시에 Android Studio를 통해 Application을 개발하였고 아두이노는 아두이노 스케치를 사용하여 C언어 기반의 아두이노 코드 작성하였다. 또 웹 서버의 경우 윈도우 환경에서 Apache Tomcat 8.0과 JDK 1.8을 이용한 JSP를 사용하여 구현 하였고 데이터베이스를 연동시켜 Oracle DB를 사용하였다. 공통적인 AP를 사용하였다.

[그림 4.1]에서의 박스 안쪽에는 아두이노와 WIFI Shield, NFC Shield로 구성되어 있다. 박스의 문 쪽 부분에서는 문을 열고 닫기를 위한 문고리자물쇠와, 그것을 동작하기 위한 서보 모터로 구성하였다.



[그림 4.1] 아두이노 개발 모듈
[Fig. 4.1] Arduino Implementation Module

다음은 NFC 태깅 모듈과 T-OTP 인증 모듈 일부분을 가상 코드로 기술하였다.

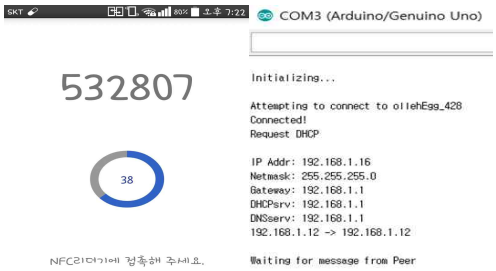
```
>>NFC Tagging<<
int msgSize = nfc.read(ndefBuf, sizeof(ndefBuf));
  if (msgSize > 0) {
       NdefMessage msg  = NdefMessage(ndefBuf, msgSize);
       int recordCount = msg.getRecordCount();
       for (int i=0;i<recordCount;i++)
       {
        NdefRecord record = msg.getRecord(i);
        int payloadLength = record.getPayloadLength();
         byte payload[payloadLength];
         record.getPayload(payload);
                String payloadAsString = "";
      }
  }

>>OTP<<
public static long create(long time){
        byte[] data = new byte[8];
        long value = time;
        for (int i = 8; i-- > 0; value >>>= 8) {
            data[i] = (byte) value;
        }
        Mac mac = null;
        try {
            mac = Mac.getInstance(ALGORITHM);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        try {
            mac.init(new SecretKeySpec(SECRET_KEY, ALGORITHM));
        } catch (InvalidKeyException e) {
            e.printStackTrace();
        }

        byte[] hash = mac.doFinal(data);
        int offset = hash[20 - 1] & 0xF;

        long truncatedHash = 0;
        for (int i = 0; i < 4; ++i) {
            truncatedHash <<= 8;
            truncatedHash |= hash[offset + i] & 0xFF;
        }
        truncatedHash &= 0x7FFFFFFF;
        truncatedHash %= 1000000;

        return truncatedHash;
    }
    public static String create() throws Exception {
        return String.format("%06d",
        create(new Date().getTime() / DISTANCE));
    }
    public boolean vertify(String code) throws Exception {
        return create().equals(code);
    }
```

4.2 구현 결과

사용자가 가지고 있는 안드로이드 폰의 NFC 기능을 활용한 시스템으로, 시간 정보와 개인 핸드폰 번호를 키 값으로 사용하여 OTP 를 생성하게 된다. 동시에 인증 서버에서도 동일한 알고리즘을 이용한 OTP 를 생성하게 되고 안드로이드 폰과 서버의 OTP가 동일한지 인증 서버에서 판별하여 결과 값을 전송해주는 인증 시스템이다.

[그림 4.2]는 안드로이드 폰에서 생성되는 OTP를 나타내며 1분마다 OTP는 생성되고 원 모양의 Progress bar는 시간의 경과에 따라 변화된다. [그림 4.3]은 아두이

노 실행 화면으로서 AP 접속 후 안드로이드 폰으로부터의 데이터 전송을 기다린다. [그림 4.5]는 인증 서버 화면으로 서버에서도 안드로이드와 동일한 알고리즘을 통해 OTP를 생성한다. [그림 4.4]와 같이 NFC를 태깅하게 되면 Client의 OTP 값을 출력한다.



[그림 4.2] 안드로이드 [그림 4.3] 아두이노
[Fig. 4.2] Android [Fig. 4.3] Arduino

[그림 4.4] 아두이노로 NFC 태깅
[Fig. 4.4] NFC Tagging on Arduino Module

[그림 4.5] 인증 실패
[Fig. 4.5] Authentication Failure

[그림 4.6] 인증 성공
[Fig. 4.6] Authentication Sucess

[그림 4.7] 인증 성공 시 도어 작동
[Fig. 4.7] Unlocking the Door in success

5. 결론

기술의 발전에 따라 사람들은 점점 편의성을 찾게 되고 이에 따른 보안성도 더욱 요구되고 있다. 많은 사람들이 안드로이드 기반의 스마트 폰을 사용하고 있고 NFC 기능을 많은 분야에 활용할 수 있지만 보안상의 취약점으로 인해 사용하는 분야가 극히 제한적이었다. 본 논문은 NFC의 기능에 OTP 암호 기법을 다중 적용하여 보안성을 강화하였고 안드로이드 폰만 가지고 있다면 누구나 손쉽게 Application을 통해 이용할 수 있다. 구현된 시스템을 통해 '도어 락' 이라는 결과물을 제작하였지만 이 외에도 최근에 등장한 모바일 결제 서비스에도 적용할 수 있고 교통 카드 등 보안이 중요시 되는 인증 시스템에서 유용하게 사용 할 수 있을 것으로 기대한다.

향후 연구 과제로는 OTP의 키 값을 스마트 폰 번호만이 아닌 사용 목적에 따라 키 값을 추가하여 도어락, 결제 시스템 등 Application 내에서 세션 별로 구분하면 시스템을 개발하는 것이다. 이를 통해 사용자가 다양한 인증 시스템을 하나의 Application을 통해 더 편리하게 이용할 수 있고 OTP의 키 값이 목적에 따라 다르게 설정되기 때문에 보안성 또한 더욱 향상 될 것이다.

REFERENCES

[1] 그림으로 쉽게 설명하는 안드로이드 프로그래밍 , 천인국 지음, 생능출판
[2] 아두이노 상상을 스케치하다 , 허경용 지음, 제이펍
[3] Beginning NFC (Near-field Communication With Arduino, Android, and Phonegap), O'Reilly Media 지음, O'Reilly Media

[4] 알기 쉬운 정보보호개론 흥미로운 암호 기술의 세계, 히로시 유키 지음, 인피니티북스
[5] 스마트폰 블루투스 이더넷 WiFi 그리고 아두이노, 조도현 원영진 동성수 남상엽 지음, 복두 출판사
[6] 두산백과, doopedia

이 범 진(BeomJin Lee) [학생회원]

▪ 2017년 : 한신대학교 컴퓨터 공학부 졸업예정

<관심분야>
사물인터넷, 컴퓨터 정보통신

김 주 성(JuSeong Kim) [학생회원]

▪ 2017년 : 한신대학교 컴퓨터 공학부 졸업예정

<관심분야>
사물인터넷, 컴퓨터 정보통신

양 기 열(GiYeol Yang) [학생회원]

▪ 2017년 : 한신대학교 컴퓨터 공학부 졸업예정

<관심분야>
사물인터넷, 컴퓨터 정보통신

양 우 범(WooBeom Yang) [학생회원]

▪ 2017년 : 한신대학교 컴퓨터 공학부 졸업예정

<관심분야>
사물인터넷, 보안 시스템

최 창 원(Changwon Choi) [정회원]

▪ 1990년 : 고려대학교 전산과학 학사
▪ 1992년 : 고려대학교 전산과학 석사
▪ 1995년 : 고려대학교 전산과학 박사
▪ 1996년 ~ 한신대학교 컴퓨터공학부 교수

<관심분야>
사물인터넷, 컴퓨터 정보통신, 네트워크 보안 등