개인 정보를 이용한 OTP 기반의 스마트 잠금장치구현

성기택*

동명대학교

Implementation of the OTP Based Smart Locks Using Personal Information

Ki-Taek Seong*

*TongMyong University

E-mail : ktseong@tu.ac.kr

요 약

기존의 잠금장치는 복잡한 패스워드 적용으로 인한 사용회피, 자주 사용 되는 키패드의 흔적으로 인한 보안성 취약 등의 단점이 있다. 본 연구에서는 기존 장치의 단점을 극복하고, 네트워크가 되지 않는 환경에서 사용자의 정보를 이용하여 보안성을 고려하는 OTP 생성을 기반으로 하는 잠금장치를 제안하고 시뮬레이션을 통하여 알고리즘을 구현하였다.

ABSTRACT

Conventional locking device has a disadvantage such as a security vulnerability caused by using avoid traces of the keypad that are frequently used due to the complex password application. In this study, we proposed a OTP based locks by using user identical informations that overcomes the disadvantages of the conventional apparatus. We implemented and validated the proposed algorithm through simulations.

키워드

일회용 패스워드, OTP, One-Time Password, 사용자 인증

Ⅰ. 서 론

최근 정부기관 시설물의 출입통제를 위한 보안 시스템에 대한 부실한 이용으로 인하여 중요한 정보가 훼손되거나 복사가 되어 외부로 유출되는 사건이 발생되어 공공기관 출입통제에 대한 우려가 증폭되고 있다. 기존의 도어락과 같은 잠금장치는 미리 암호를 설치 시 저장해두고 사용자가 암기하여 사용토록 되어있으나 보안성을 높이기 위하여 길고 복잡한 숫자를 나열하여 사용토록 권장하고 있다. 그러나 보안성을 강조하여 길고 복장한 숫자 또는 기호를 조합함으로서 사용자는 이를 외우거나 기록하여 둠으로서 사용에 불편함이 제기되었고 이번 사건과 같이 암호코드를 잊어버릴 경우를 대비하여 잠금장치 근처에 적어두는 바람에 타인이 이를 이용하여 정보출입통제 시스템이 무너지는 사건이 되었다.

본 논문에서는 OTP(One-Time Password) 기반의 스마트기기를 이용한 잠금장치 구현에 관하여 기술하였다. 여러가지의 일회용 패스워드 생성방식이 있으나 본 연구에서는 기본적으로는 시간동기화방식을 이용하고 여기에 사용자의 고유의 정보를 이용함으로서 제한된 접근을 가능하게 하였다. 또한 잠금장치가 사용되는 환경을 유무선통신이 되지 않는 곳으로 하여 지하시설물, 도서지역 등 열악한 사용 환경을 고려하였다.

Ⅱ. 본 론

2.1 잠금장치의 인증방식

IT기술의 발달로 인하여 잠금장치는 기존의 물리적인 키를 이용한 방식에서 디지털값을 이용하여 잠금장치를 개폐하는 디지털 로어락 시스템으로 발전되었다. 디지털 도어락시스템의 핵심은 사

용자와 시스템사이의 패스워드를 이용한 인증방식이다. 이러한 패스워드 인증방식은 사용에 간단하다는 장점이있으나 일반적인 사용자들은 자신이 기억하기 쉬운 기호를 사용하여 패스워드를 구성하므로 사회공학적 공격에 취약하다. 또한 동일한 암호를 자주 사용하면 해당 키패드에 흔적이 남아 있어서 물리적인 취약점을 갖고 있다. 이러한 기존의 패스워드 인증방식의 문제를 해결하기 위하여 일회용 패스워드방식이 제시되었다. 다양한 기술의 OTP 기반의 잠금장치가 선보였으며 관련 특허도 출원되었다[1]. [1]에서는 서버를 기반으로 인증 관련된 정보를 저장해두고 필요시 OTP를 생성하여 이용하는 경우인데 이러한 경우에는 네트워크 환경이 필수적이다. 상품으로 제안된 도어락[2]은 네트워크 환경과 무관하게 사용할 수 있으나 사용자 등록 등 기존의 도어락 관리 방식과 유사하게 초기 등록 번호 암기, 비밀번호 암기 등과 같은 사용상의 단점과 일정한 패스워드에 해당하는 자리 수가 일정하여 장기간 관찰 시 숫자 위치정보가 노출될 가능성 있다.

2.2 제안하는 방식

일회용 패스워드방식은 인증서버와의 동기여부에 따라 동기/비동기 방식으로 분류될 수 있는데, 비동기 방식은 사용자가 직접 OTP값 생성에 기인하기 때문에 상호 인증과정을 거친다는 장점이 있으나 매번 사용자 입력으로 인한 불편함이 있으며, 동기화 방식은 상호 동기시간이 일치하지 않을 경우 인증받지 못하고 또한 동기화 시간동안은 패스워드 재사용 문제가 제기된다.

일반적인 잠금장치가 유무선통신이 되지 않는 공간에서 적용될 경우 기존의 인증서버를 이용한 일회용 패스워드방식을 적용할 수 없다.

본 연구에서는 도어락과 사용자 스마폰을 이용한다는 환경을 전제로 한다. 그리고 도어락에 사용자 정보를 기반으로하는 OTP 생성 방식을 적용한 잠금장치를 제안한다. 잠금장치는 초기에 사용자 정보(예를 들면 사용사의 스마트폰의 MAC 주소)를 저장해 두고 있으며 초기에 두 장치간 인증을 위한 통신은 부루투스와 같은 무선 통신 장치를 갖고 있으며 브루투스의 경우 상호연결(paring) 정보는 미리 되어 있는 것으로 한다. 그림 1은 제안하는 방식을 기반으로 하는 잠금장치 동작 절차이다.

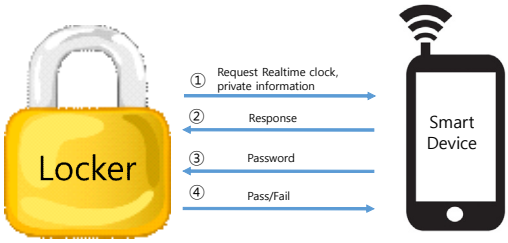

그림 1. 제안하는 방법의 동작 절차

절차 1 : 잠금장치는 사용자에게 실시간 정보 및 개인정보를 요청.

절차 2 : 사용자의 스마트폰은 현재시간정보와 개인정보(48비트의 스마트폰 MAC 주소) 전달. (이후 잠금장치와 사용자의 스마트폰은 상호 OTP 생성를 생성한다.)

절차 3 : 사용자는 스마트폰에서 생성한 OTP를 잠금장치의 키패드를 통하여 입력한다.

절차 4 : 잠금장치는 자신이 생성한 OTP와 입력된 OTP를 비교하여 일치하면 잠금기능을 해제한다.

2.3 OTP 알고리즘 구현

본 연구에서 사용한 OTP 알고리즘은 실시간 시간 정보를 이용한 방식이다. 사용자 스마트 폰의 경우 시간정보를 보유하고 있으나 잠금장치에서는 에너지 손실을 최소화하기 위하여 CPU 동작모드를 STOP/IDLE/SLEEP 등으로 운영되며 따라서 실시간 정보는 없을 수 있다. 또한 네트워크 환경이 아니므로 시간 정보를 가져올 수 없다. 이러한 환경에서 잠금장치가 현재시간에 대한 정보를 얻기 위해서는 자체에서는 소프트웨어를 이용한 실시간 발생 알고리즘 구현 및 현재시간 정보가 요구된다. 본 연구에서는 하드웨어 자체 내부 타이머를 이용한 실시간 발생 알고리즘(swRTC: software Real Time Clock)을 이용하였다[3]. [3]의알고리즘에서 입력 항목으로는 특정 키 값과 현재 시간 값 두가지를 요구한다. 특정 키 값은 상호 인증을 하기 위한 비밀 키로서 반드시 동일한 입력값을 사용해야 한다.

본 연구에서는 특정 키를 예비 사용자 인증수단으로 사용한다. 예를 들면 즉 잠금장치에서는 미리 등록한 다수의 사용자 스마트폰의 MAC주소를 저장해 두고 이를 이용하여 인증함과 동시에 OTP 생성에 이용한다. 사용자 인증 수단은 지문, 특정 패스워드 등 다용한 방법이 적용가능하다.

III. 구현 및 결론

제안한 방법의 구현을 위하여 아두이노 UNO를 이용하였으며 개발 프렛폼과 결과는 그림 2와 같다. 인증값으로는 개발 PC의 MAC주소를 사용하였다.

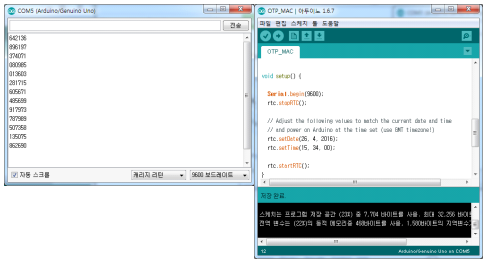

그림 2. 개발 프렛폼 결과

그림에 나타난바와 같이 6자리의 난수가 발생됨을 보이고 있다.

본 연구에서는 네트워크가 되지 않는 특수한 상황에서 사용자의 정보를 이용한 OTP 기반 스마트잠금장치를 제안하였다. 기존의 온라인 서버를 이용할 수 없는 환경에서 사용가능하며, 보편화된 스마트 기기를 사용하여 사용자 인증을 고려한 OTP 생성 방법을 기반으로 하였다. 사용자 인증은 사용자 기기정보뿐만이 아니라 사용자의 지문 정보도 활용가능하며 잠금장치와 스마트폰을 이용하여 잠금장치의 사용 이력, 잠금장치 사용자 이력 등의 다양한 응용이 기대된다.

참고문헌

[1] 디지탈 도어락과 무선통신 단말기를 이용한 출입 인증 방법 및 그 장치 , WO 2014157770 A1

[2] http://www.hidekey.co.kr/

[3] https://github.com/leomil72/swRTC