논문 17-42-02-09 The Journal of Korean Institute of Communications and Information Sciences '17-02 Vol.42 No.02
https://doi.org/10.7840/kics.2017.42.2.349

SMS 기반 인증의 보안 취약점을 개선한 스마트폰 소유 및 위치 확인 기법

권 성 재◆, 박 준 철°

Smartphone Ownership and Location Checking Scheme for Fixing the Vulnerabilities of SMS-Based Authentication

Seong-Jae Kwon◆, Jun-Cheol Park°

요 약

많은 웹 사이트들이 사용자가 패스워드를 분실하거나 온라인 결제를 진행하는 등의 상황에서 SMS(Short Message Service) 기반의 사용자 인증을 채택하고 있다. SMS 기반 인증에서 인증 서버는 텍스트를 평문으로 전송하기 때문에 공격자가 그 텍스트를 도청하거나 가로채면 다른 사람(피해자)인 것처럼 인증을 받을 수 있다. 본 논문에서는 사용자가 스마트폰을 지금, 어느 위치에서 소유하고 있는지를 인증하는 챌린지-응답(challenge-response) 형태의 인증 방식을 제안한다. 제안 방식은 서버가 보낸 챌린지, 사용자의 현재 위치정보, 스마트폰에 저장된 비밀 값을 모두 사용하여 응답을 생성한다. 그 결과로, 단순히 사용자가 받은 SMS 메시지를 어떤 가공도 없이 그대로 서버로 되돌리는 SMS 기반 인증에 비해, 제안 방식은 훨씬 더 안전하다. 제안 방식은 기존 SMS 기반 인증의 텍스트에 해당하는 응답의 입력과 더불어, 인증 과정의 시작을 위해 추가로 패스프레이즈(passphrase)의 입력을 요구하나, 추가 입력의 부담은 향상되는 보안성을 고려할 때 대부분의 사용자들이 감내할 수 있는 수준이라 판단한다.

Key Words : SMS-based authentication, Challenge-Response, Ownership and Location Checking, Authentication App, Passphrase

ABSTRACT

Many Web sites adopt SMS(Short Message Service)-based user authentication when a user loses her password or approves an online payment. In SMS-based authentication, the authentication server sends a text in plaintext to a user's phone, and it allows an attacker who eavesdrops or intercepts the text to impersonate a valid user(victim). We propose a challenge-response scheme to prove to the authentication server that a user is in a certain place at the moment with her smartphone beside her. The proposed scheme generates a response using a challenge by the server, user's current location, and a secret on the user's smartphone all together. Consequently, the scheme is much more secure than SMS-based authentication that simply asks a user to send the same text arrived on her phone back to the server. In addition to entering the response, which substitutes the SMS text, the scheme also requests a user to input a passphrase to get the authentication process started. We believe, however, the additional typing should be tolerable to most users considering the enhanced security level of the scheme.

※ 이 논문은 2016학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음.
◆ First Author : Department of Infrastructure Security, NHN Entertainment, rjstjdwo14@naver.com, 정회원
° Corresponding Author : Department of Computer Engineering, Hongik University, jcpark@hongik.ac.kr, 종신회원
논문번호 : KICS2016-12-406, Received December 27, 2016; Revised February 8, 2017; Accepted February 8, 2017

Ⅰ. 서 론

온라인에서 사용자를 인증하기 위한 여러 기술 중에서 유지비용과 사용편의성 측면에서 강점을 가진 비밀번호를 기반으로 하는 인증 기술이 가장 널리 쓰이고 있다. 하지만 사용자들은 같은 비밀번호를 여러 사이트에 사용하는 경향이 있기 때문에 한 사이트의 패스워드가 유출되면 다른 사이트들의 안전성도 위협받는다. 단일인증 방식의 보안성을 강화하기 위하여 이중인증(two-factor authentication)[1]이 제시되었는데, 휴대폰을 활용하는 SMS(Short Message Service) 기반 인증이 두 번째 인증, 또는 주 인증 수단의 사용이 불가능한 경우에 대체 인증 수단으로 많이 사용되고 있다.

SMS 기반 인증은 널리 사용되고 있음에도 불구하고, 여러 공격에 그 취약함이 드러났다. 2012년 12월에 인증 메시지를 탈취하는 목적의 악성코드 '체스트(chest)'[2]가 발견되고, 2016년 8월에는 SMS 인증의 취약점을 악용하여 모바일 메신저인 텔레그램(Telegram) 사용자의 개인정보를 탈취하는 사건[3]이 발생하는 등, SMS 기반 인증은 결코 안전한 인증 수단이라 볼 수 없다. 이에 국내에서는 2014년부터 은행권에서 SMS 기반 인증을 사용 중단하기 시작했고[4], 2016년 8월 미국 NIST(National Institute of Standards and Technology)는 정부 서비스 제공자들에게 SMS 기반 인증을 사용하지 말 것을 권고했다[5].

본 논문에서는 SMS 기반 인증 기법의 보안상 취약점을 분석하고, 보안성이 대폭 향상된 대체 인증 방식을 제안한다. 제안 방식은 인증 요청된 시각에, 서버에 등록된 주요 활동지역에서 인증 요청자가 스마트폰을 소유하고 있다는 사실을 인증 서버가 확인하도록 한다. 이를 위해 인증 요청자는 스마트폰을 통하여 인증 서버와 챌린지-응답 기반의 메시지 교환을 하며, 이 때 응답에 사용자의 현재 위치정보가 자동으로 포함된다. 스마트폰에서 구동되는 인증 프로그램은 사용자의 올바른 패스프레이즈(passphrase) 입력 시에만 동작하기 때문에 스마트폰 분실 시의 악용 가능성도 낮다.

본 논문의 구성은 다음과 같다. 2장에서는 SMS 기반 인증 절차 및 취약점을 제시하고, 취약점을 악용하는 여러 실제 공격 사례들 및 이와 관련한 연구들을 살펴본다. 3장에서는 스마트폰에 설치되는 클라이언트 프로그램의 설계를 상세하게 서술한다. 4장에서는 제안한 인증 방식의 프로토타입 구현에 대해 개발환경, 구현방식, 실행 예를 위주로 설명한다. 5장에서는 제안 기법의 효용성을 보안성 및 편의성 측면에서 분석하고, 6장에서 결론을 제시한다.

Ⅱ. 배경 및 관련연구

2.1 SMS 기반 인증

SMS 기반 인증은 등록된 사용자 휴대폰 번호로 일회용 인증번호가 포함된 SMS를 전송한 뒤 홈페이지를 통해 입력받은 번호가 전송된 번호와 일치하는지 확인함으로써 상대방을 등록된 사용자로 인증하는 기법이다. SMS 기반 인증은 적용하는데 기술적, 비용적인 부담이 매우 적고, 이동통신사를 통하여 휴대폰 소유자의 인적사항을 확인할 수 있다는 장점이 있어, 사용자 주 인증 수단의 사용이 불가능한 상황(패스워드 분실 등)이나 중요한 서비스를 제공하기 전에 추가적인 인증수단이 필요한 경우(인터넷 뱅킹, 결제 등)에 널리 사용된다.

SMS 기반 인증은 사용자가 소유하고 있는 것(what you have)을 인증하기에 불충분하다는 한계를 가진다. 두 가지 문제점이 존재하는데, 첫째, 휴대폰의 도난이나 분실 시 취득자/도둑에 의해 해당 기기에 대한 물리적 접근이 가능해 진다는 것과 둘째, 인증 SMS를 훔쳐보거나, 우회시켜 공격자에게로 전송시키는 것이 가능하다는 것이다.

스팅레이(Stingray)같은 IMSI-캐처(International Mobile Subscriber Identity-Catcher)[6]를 사용하면 SMS의 내용을 훔쳐보거나 SMS를 가로챌 수 있다. IMSI-캐처는 VBTS(Virtual Base Transceiver Station)[7]으로, 휴대전화의 트래픽을 가로채고 휴대전화 사용자의 움직임을 추적하는데 사용되는 전화 도청장치이다. 이 장치는 기지국처럼 작동하며 주변 사용자의 모든 전화 통화와 텍스트를 도청하고, 심지어 트래픽을 다른 네트워크로 우회시킬 수도 있다. 그밖에 공격자가 문자메시지를 통한 스미싱 공격[8]으로 스마트폰에 악성코드[9]를 설치하면 인증 메시지를 공격자의 휴대폰으로 우회시키는 것도 가능하다. 2012년 12월 안랩은 국내 스마트폰 사용자의 금전 탈취를 노린 안드로이드 악성코드 '체스트'[2]를 발견했다고 발표했다. 악성코드 '체스트'는 SMS로 유포되며 사용자를 속이는 내용과 함께 URL을 같이 첨부하여 사용자가 악성코드를 설치하도록 유도한다. 이후 공격자는 사전에 가지고 있던 공격 대상자의 개인정보로 스마트폰 소액결제를 시도하고, 결제 시스템에서 피해자의 스마트폰으로 인증 SMS를 보내면 '체스트'가 인증 SMS를 가로채서 결제를 완료시킨다. 피해자는 과금

정보를 확인하기 전까지 이러한 공격을 당한 사실을 알기 어렵기 때문에 공격의 피해규모가 커질 수 있다. 2016년 8월에는, 종단 간 암호화(end-to-end encryption)를 적용하여 보안성이 매우 높다고 평가받는 메신저인 텔레그램(Telegram)이, 이란의 해커들에게 공격당하여 이란의 텔레그램 사용자 1500만 명의 전화번호와 아이디가 유출되는 사건이 있었다[3]. 그 결과로 공격자가 십여 개의 유출된 계정에 로그인한 후 일부 대화 내용을 복원해 내었다. 이 공격은 해커들이 사전에 텔레그램 사용자의 스마트폰에 악성코드를 침투시켜 놓고, 본인인증을 할 때 발생하는 인증 SMS를 가로채서 다른 스마트폰에 사용자 계정 자체를 복사하는 방식으로 이루어졌다.

2.2 SMS 기반 인증 개선 연구

Varghese 등[10]은 휴대폰을 이용한 안전한 이중인증을 위해 SMS 기반의 일회용 암호(One-Time Password, 이하 OTP)를 제안하였다. 사용자가 SMS로 자신의 개인정보를 암호화하여 전송하면, 서버는 이를 확인한 뒤 유효한 사용자인 경우 Diffie-Hellman 알고리즘으로 OTP를 암호화하여 SMS를 통해 사용자에게 전송하는 방식이다. 지선수[11]는 사용자 아이디의 일부 정보, 입력받은 패스워드, 사용자가 SMS를 수신한 뒤 SMS의 값을 입력하는데 걸리는 시간을 모두 이용하여 일회용 패스워드를 생성하는 방식을 제안하였다. Ahmed 등[12]은 중첩(convolution)과 뒤섞기(shuffling)를 사용하고, 정적 암호화 키를 사용한 RC4 알고리즘을 사용하여 문자를 암호화하였다. 또한 추가 연구[13]로, 새로 제안한 해시 함수와 공유키를 사용하여 동적 암호화 키를 생성하는 방식을 제안하였다. 이 방식에서는 문자 송신자가 중첩과 뒤섞기를 적용하고 해시 함수와 타임스탬프, 공유키를 이용하여 동적으로 암호화 키를 생성한 뒤, 문자 내용을 암호화하고 타임스탬프를 더하여 문자를 전송한다. 수신자는 해당 과정을 역으로 수행하여 문자를 복원한다. 이와 같이 공유키와 타임스탬프를 이용하여 동적으로 암호화 키를 생성하는 것은, 본 논문에서 공유키와 챌린지를 응답 생성에 사용하는 것과 유사하다. 하지만 해당 방식[13]은 SMS의 기밀성을 보장하고 교환 메시지를 인증하기 위한 것임에 비해, 본 논문의 방식은 사용자가 특정 장소에서 스마트폰을 소유하고 있는지를 확인하기 위함으로 사용 목적이 서로 다르다. AiZomai 등[14]은 온라인 뱅킹 인증 솔루션의 대역 외 채널을 이용한 인증 방식이 공격에 취약하다고 주장하며, SMS 기반 인증 방식의 유용성을 개선하는 기법을 제시하였다. 해당 기법은 트랜잭션의 유효성을 검사하기 위해 화이트 계정 목록(white account list)과 블랙 계정 목록(black account list), 알려진 계정 목록(known account list)을 사용한다. 즉, 트랜잭션의 계좌번호가 화이트 계정 목록에 포함되어있는지 검사하여 신뢰할 수 있는지 판단하고, 트랜잭션의 계좌번호가 블랙 계정 목록에 있다면 사용자에게 공격에 대해 경고하는 방식으로 유용성을 개선한다. 본 논문의 제안 방식은 SMS 기반 인증을 아예 새로운 방식으로 대체하려는 것으로, SMS 인증의 개선이 주된 목적인 이러한 연구들과는 차이점을 가진다.

한편 SMS 기반 인증을 새로운 방식으로 대체하려는 여러 시도들이 발표되었다. 박지예 등[15]은 QR 코드 기반 상호 인증 시스템을 제안하였다. 이 기법은 스마트 기기와 QR 코드를 활용한 두 가지 채널(two-channel), 두 가지 요소의 인증 시스템이다. 그러나 해당 기법은 인증을 위해 사용자가 화면의 QR 코드를 카메라로 찍어야 하며, 모바일 기기로 로그인할 때는 모바일 기기의 QR 코드를 찍는 다른 장치가 필요하다는 불편이 따른다. 이상혁 등[16]은 사용자 편의성을 강화한 모바일 메신저 기반의 이중인증 기법을 제안하였다. 이 기법에서는 사용자를 웹 사이트에서 1차 인증 후, 웹 서버가 사용자의 모바일 메신저로 인증 URL을 전송한다. 사용자는 수신된 메시지를 확인 후 인증 URL을 클릭하여 인증을 완료한다. 그러나 해당 기법은 스마트폰 분실/도난 상황에 대한 고려를 하지 않았기 때문에 공격자가 스마트폰을 탈취/습득하게 되면 비교적 쉽게 공격을 성공시킬 수 있다. Abdurrahman 등[17]은 사전 공유 숫자(pre-shared number), GPS 위치, 타임스탬프를 이용한 인증 기법을 제안했다. 해당 기법은 사용자의 모바일 폰에서 GPS 위치 검색을 이용하면 사용자의 위치와 타임스탬프 정보가 모바일 폰과 GPS 서버에 저장된다는 것을 이용한다. 사용자 폰은 우선 사전 공유 숫자, GPS 위치, 타임스탬프 정보로 토큰을 만들고 인증 서버에게 전송한다. 인증 서버는 GPS 서버에서 해당 사용자의 GPS 정보(GPS 위치, 타임스탬프)를 받아와서 토큰을 만들고, 이를 사용자가 보낸 토큰과 비교하여 검증한다. 이 방식에서는 사용자 위치정보의 현재성을 검증하기 위해 GPS 서버에 저장된 GPS 검색 기록의 타임스탬프를 이용하기 때문에, 인증 서버가 GPS 서버에 사용자의 기록을 질의하는 과정이 필요하여 인증에 소요되는 시간이 증가한다. 또한 사용자 폰의 사전 공유 숫자를 보호하는 장치나 기법이 존재하지 않으므로 휴대폰의 분실이나 도난 시에 공격자에 의한

위장 인증이 가능할 수 있다.

Ⅲ. 휴대폰 소유 및 위치 확인을 통한 인증

제안하는 방식의 구현이 가능한 휴대폰은 3G 이동통신 단말기부터 탑재되는 심(SIM) 카드를 가진 Android 기반 스마트폰이다. 2016년 3월 한국인터넷진흥원 모바일인터넷이용실태조사 보고서[18]에 따르면 국내 스마트폰 사용자의 89.8%가 이에 해당하는 폰을 소유하고 있다.

3.1 인증 어플리케이션 다운로드 및 설치과정

사용자는 인증 서버에 회원가입을 한 뒤, 온라인 앱 스토어로부터 인증 애플리케이션(application, 이하 인증 앱(app)이라 함)을 다운로드받은 후 설치 과정을 진행한다. 설치 과정에서 사용자는 자신이 주로 활동하는 지역을 최대 3개까지 등록할 수 있다. 만약 사용자가 위치등록을 하지 않겠다고 선택(보안상 권장되지 않음)하면, 인증 서버와 인증 앱은 사전에 약속된 와일드카드 값(0으로만 이루어짐)을 위치정보로 사용한다. 이 경우 어느 위치에서든 인증을 받을 수 있지만, 제안 방식의 보안성 이점을 충분히 누리지 못함을 감수해야 한다. 등록 가능한 지역 개수는 추후 사용편의성을 고려하여 조정될 수 있다. 설치 과정은 그림 1과 같다.

사용자 스마트폰에서 인증 앱이 최초로 실행되면 앱은 그 사실을 서버에게 알린다. 서버는 자신의 인증서를 사용자 스마트폰으로 전송하고, 인증 앱은 서버의 인증서를 이용하여 SSL 통신으로 자신의 스마트폰 번호와 등록하고 싶은 위치정보를 전송한다. 인증 서버는 스마트폰 번호를 기반으로 비밀 값(S)을 생성하여 서버에 저장하고 사용자 스마트폰에 전송한다. 인증 앱은 사용자로부터 패스프레이즈를 입력받고, 패스

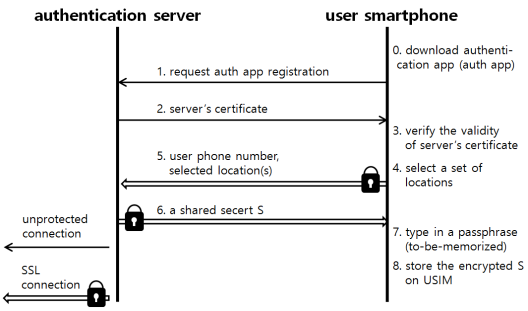


그림 1. 인증 앱 설치 과정 흐름도
Fig. 1. Flowchart of the Authentication Application Installation

프레이즈(P)의 해시함수 적용 결과인 $h(P)$를 암호화 키(K)로 사용하여, 비밀 값(S)을 대칭키 암호화한 결과인 $E(S, K)$를 유심(USIM)에 저장한다. 대칭키 암호화는 AES-256 알고리즘, 해시 함수는 SHA-256 알고리즘을 사용하여 구현한다.

3.2 인증 과정 개요

인증 과정에 참여하는 주체는 인증서버, 사용자 PC(또는 사용자 스마트폰) 및 서버의 인증 챌린지를 수신, 처리할 사용자 스마트폰이다. 사용자는 자신의 PC(또는 스마트폰)로 인증 서버가 제공하는 홈페이지에 로그인 후 이중인증의 첫 번째 인증을 완료한 뒤, SMS 기반 인증을 대체하는 두 번째 인증을 진행한다. 이 과정은 그림 2와 같다.

사용자는 스마트폰의 인증 앱을 실행하여 인증 요청을 보낸다. 인증 서버는 챌린지를 생성하여 사용자 스마트폰의 인증 앱에 전송한다. 이후 인증 서버와 인증 앱은 각각 응답을 계산한다. 사용자는 인증 앱이 계산한 응답을 PC(또는 스마트폰)의 입력창에 입력하고, 인증 서버는 사용자 PC(또는 스마트폰)가 전송한 응답과 서버에서 계산한 응답을 비교한 뒤 인증 결과를 인증 앱으로 전송한다.

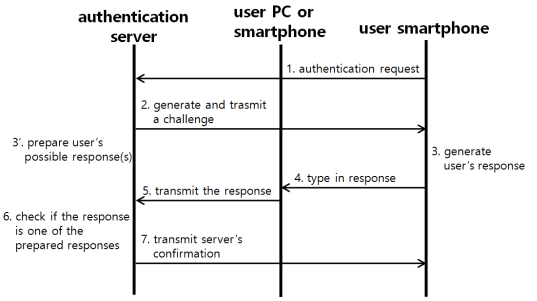


그림 2. 인증 과정 흐름도
Fig. 2. Flowchart of an Authentication Process

3.3 챌린지 생성 및 전송

서버가 어떤 사용자에게 요청 때마다 같은 챌린지를 반복 제시하면, 해당 트랜잭션의 내용을 탈취한 공격자는 이후 재생(replay) 공격을 시도할 수 있다. 따라서 서버의 챌린지 값은 적어도 동일 사용자에게 반복되어서는 안 된다. 이를 위해 서버는 HC-256 스트림 암호화 알고리즘[19]을 통해 생성된 랜덤 키스트림(keystream)을 128-bit 단위로 자른 것을 챌린지로 사용하며, 매번 새로운 위치에서 챌린지를 취한다. HC-256은 유럽의 eSTREAM이 선택한 스트림 암호화 알고리즘들 중 하나로 256-bit의 비밀 키(key) 와

256-bit의 초기 벡터(IV)를 이용하여 뛰어난 유사랜덤 성질을 보이는 키스트림을 최대 2^{128}bits 만큼 생성해 낸다. HC-256에 대해 알려진 어떤 공격도 2^{128}bits를 훨씬 초과하는 길이의 스트림을 필요로 한다는 점에서 생성된 키의 안전성이 담보된다. HC-256은 특허에 걸려있지 않아 사용에 제한이 없다. 서버는 랜덤한 키 및 IV 값으로 HC-256를 실행하면서, 챌린지를 요구하는 모든 사용자들에게 키스트림을 계속 128-bit 크기로 생산하여 제공한다. 이 과정을 제공한 챌린지의 총 크기가 2^{128}bits에 이를 때까지 계속한 후, 다시 새로운 키 및 IV를 가지고 챌린지 생성을 다시 시작한다.

3.4 인증 앱에서 응답 생성

1) 인증 앱이 실행되면 GPS와 무선 네트워크를 이용하여 현재 스마트폰의 위치정보(위도, 경도)를 수집한 뒤 우리나라 행정구획상의 시/군/구 단위(예: □□□□시 ○○구, ◇◇◇도 △△시 등)까지의 위치정보(L)를 얻어 응답 생성에 사용한다. 만약 사용자가 인증 앱 설치과정에서 위치정보를 등록하지 않았다면, 와일드카드 값이 위치정보(L)로 사용된다.
2) 인증 앱은 서버와 공유하고 있는 비밀 값(S), 서버로부터 전송받은 챌린지(C), 현재 스마트폰의 위치정보(L)를 이용하여 응답을 생성하고 휴대폰 화면에 출력한다. 비밀 값(S)과 챌린지(C)는 128-bit 길이로 16진수 문자열로 표현된다. 위치정보(L)는 한글 문자열이며 utf-8 인코딩을 사용해 표현된다. 응답 계산에는 HMAC이 다음과 같이 사용된다.

$$Y = HMAC(S, C \| L) \quad (1)$$

$$= h((S^{+} \oplus opad) \| h((S^{+} \oplus ipad) \| (C \| L))) \quad (2)$$

R = last 36 bits of Y

(단, $h()$는 SHA-256, $\|$는 concatenation)

식(1)의 HMAC 계산의 해시 함수로 SHA- 256을 사용하였고, 식(2)의 S^{+}는 SHA-256의 입력 블록 크기인 512-bit (64- byte)만큼 0으로 S를 padding한 것, $ipad$와 $opad$는 각각 0x36, 0x5c를 64번 반복하여 SHA-256 블록 크기인 512-bit로 만든 것이다. 사용자가 입력한 패스프레이즈를 해시한 값, 즉 $h(P)$를 암호화 키(K)로 암호화하여 유심에 저장된 $E(S, K)$를 복호화한 결과인 S를 응답 생성에 사용한다. 만약 사용자가 잘못된 패스프레이즈를 입력한다면 복호화 과정에서 문제가 발생하거나, 서버와는 다른 S 값이 생성되어 응답을 만드는데 사용될 것이다. Y의 마지막 36-bit를 취하여 이를 R로 삼고, R을 6-bit 단위로 나누어 각각을 base64로 인코딩한 결과의 6개의 문자(숫자, 영대문자, 영소문자, +, @)를 캡챠(CAPTCHA) 형태로 화면에 출력한다. 사용자는 스마트폰 화면상의 응답을 PC (또는 스마트폰) 입력창에 입력하여 입력한 값이 서버로 전달되도록 한다.

3.5 서버의 응답 검증 및 키 갱신

서버는 사용자에게 보낸 챌린지(C), 서버에 저장된 해당 사용자의 비밀 값(S) 및 사용자가 사전에 등록한 위치정보(L)를 이용하여 위와 같은 방법으로 HMAC을 계산을 한다. 계산 결과인 R'이 사용자 스마트폰에서 전송된 R과 일치할 때만 서버는 성공적으로 인증을 완료한다. 사용자가 등록해놓은 지역이 2곳 이상일 경우 지역마다 각각 R'을 생성하여 수신된 R과 일치하는 R'이 하나라도 존재하면 서버는 인증 성공을 판정한다. 따라서 사용자 스마트폰이 현재 서버에 등록한 지역에 있지 않다면 인증은 실패하게 된다. 만약 사용자가 설치 과정에서 서버에 위치를 등록하지 않았다면, 서버에서는 위치정보(L)를 와일드카드 값으로 사용하여 R'을 계산하기 때문에 사용자의 현재위치가 어디든 상관없이 위치의 일치가 발생하게 된다.

인증과정이 정상적으로 완료되어 사용자가 인증에 성공하면 서버와 사용자 스마트폰 인증 앱 모두 비밀 값(S)을 갱신하여 서로 동기화한다. 비밀 값(S)은 해시함수 SHA-256을 이용하여 $S^{new} = h(S^{old})$ 와 같이 갱신된다. 사용자 스마트폰에서는 비밀 값(S)을 갱신한 뒤 앞서 입력받았던 패스프레이즈(P)를 기반으로 만든 암호화 키(K)를 이용하여 비밀 값(S)을 암호화한다. 즉 $E(S^{new}, K)$ ($K = h(P)$)를 유심에 저장하는 것으로 인증 과정이 완료된다.

Ⅳ. 프로토타입 구현

제안 인증 방식의 실현가능성을 보이기 위하여 프로토타입을 구현하였다. 인증 서버 프로그램은 파이썬(Python) 3.4.5을 이용하여 구현하였고, 인증 앱은 Android Studio 2.1.3을 이용하여 구현하였다.

4.1 서버 프로그램 구현

인증 서버 프로그램은 CentOS 6 운영체제에서 파이썬 3.4.5을 이용하여 구현하였다. 챌린지 생성을 위해, HC-256 스트림 암호화 알고리즘의 C언어로 구현된 소스코드를 동적 라이브러리 형식으로 변환 후, 파이썬 프로그램의 런 타임에 로드하여 사용하였다. HC-256 알고리즘에 입력으로 들어가는 키와 IV는 인증 서버만 알고 있는 비밀 값(M)과 현재 서버 시간(T_s)을 이용하여 SHA-256 해시함수로 $h(M \| T_s)$을 계산한 뒤 상위 128-bit를 키, 하위 128-bit를 IV로 사용하였다. 한번 계산된 키와 IV를 이용하여 키스트림을 2^{128} bits만큼 생성한 뒤에는, 달라진 현재 서버 시간 (T_s')을 이용하여 다시 $h(M \| T_s')$를 계산하여 키와 IV를 갱신한 후 HC-256 알고리즘을 실행하도록 한다.

4.2 인증 앱 구현

인증 앱은 윈도우 10(Windows 10) 환경에서 Android Studio 2.1.3을 이용하여 구현하였으며, gradle build 옵션은 compileSdkVersion 22, minSdkVersion 19, targetSdkVersion 22로 설정하였다. 주요 구현내용은 인증 앱 최초 설치과정, 패스프레이즈 등록과 검증, 화면 캡처 방지 및 보안 키보드이다.

1) 패스프레이즈 등록, 검증 구현
사용자가 잘못된 패스프레이즈를 입력하거나, 응답을 서버에 잘못 입력한 경우 오류 메시지를 띄우고, 다시 패스프레이즈를 입력 받는다. 패스프레이즈 입력 오류가 5번 발생하면 앱의 사용을 금지시킨다. 패스프레이즈를 분실하였거나 앱 이용 불가 상태가 되면 설치된 앱을 삭제하고 재설치 하도록 하여, 추측 공격에 의한 피해 가능성을 줄였다.
2) 계산된 응답을 화면에 보여주기
봇(bot)에 의한 자동화된 공격 시도를 막기 위하여 응답을 캡챠의 형태로 화면에 출력한다. 화면에 나타난 응답은 6개의 문자로 구성되는데, 각 문자는 0~9, a~z, A~Z, @, +의 64개 문자 중 하나이다. 숫자 1, 대문자 I, 소문자 l와의 혼동을 피하기 위해 '/'가 '@'으로 대체되었으며, 숫자와 +, @에는 밑줄을 넣었다. 응답 캡챠의 화면 출력은 그림 3과 같다.
3) 인증 앱의 화면 캡처 금지 기능
사용자가 인증 앱에 입력하는 위치정보나 패스프레이즈는 스마트폰 화면을 캡처하는 기능을 가진 악성코드에 의해 손쉽게 탈취당할 수 있으므로, 인증 앱이 실행되어 스마트폰 화면에 활성화 되어있는 동안에는 캡처 기능을 금지하도록 한다. 이를 위해 안드로이드의 WindowManager.LayoutParams에 있는 플래그들 중 FLAG_SECURE를 사용하여 안드로이드 운영체제 수준에서 캡처가 금지되도록 강제하였다. 그림 4는 캡처 금지 기능이 적용되어 있는 인증 앱에서 캡처를 시도했을 때의 화면이다.
4) 보안 키보드 구현
사용자가 스마트폰에 패스프레이즈를 입력할 때 정적인 가상 키보드를 이용한다면, 터치 좌표를 수집하는 악성코드에 의해 패스프레이즈가 탈취될 수도 있다. 이에 대비하여 패스프레이즈 입력 시 화면에 나타나는 키보드는 매 키 입력 직후 키 배치가 무작위로 바뀌도록 구현하였다. 그 결과 인증 앱은 안드로이드 기본 키보

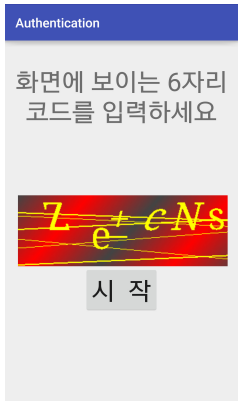


그림 3. 응답 화면 (캡챠 적용)
Fig. 3. Response Entry Screen (CAPTCHA applied)

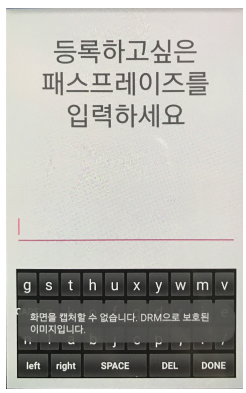


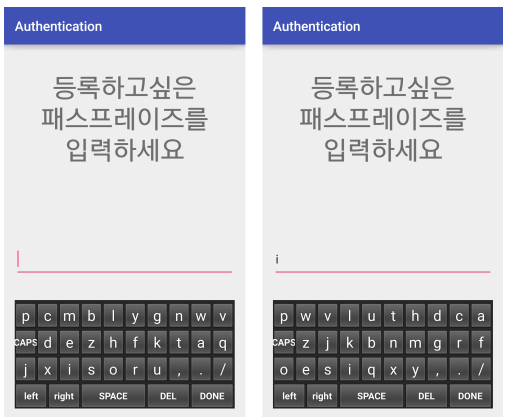
그림 4. 화면 캡처 금지
Fig. 4. Inhibit Screen Capture

드 대신에 키 배치가 매번 달라지는 보안 키보드를 통하여 사용자 입력을 받는다. 그림 5의 (a)와 (b)는 패스프레이즈 입력 시 한 문자를 입력하면 그 직후 다음 입력을 위하여 키보드 배치가 무작위로 바뀌는 것을 보여준다.

V. 보안성 및 사용 편의성 분석

5.1 보안성 분석

공격자가 사용자의 스마트폰으로 전송되는 챌린지를 탈취할 수 있는 환경을 구축한 뒤, 미리 탈취한 사용자의 개인정보로 결제를 시도하는 상황을 가정하자. 기존 SMS 기반 인증은 공격자가 결제 시도를 하면서 악성코드로 스마트폰으로 오는 문자를 빼돌림으로써 공격자가 결제 과정을 완료할 수 있다. 제안 방식은 공격자가 인증 챌린지를 빼돌리는데 성공하여도, 응답을 만들기 위해서는 (1) 스마트폰에 저장된 비밀 값(S), (2) 사용자가 서버에 등록해놓은 위치정보(L)를 모두 알아야 한다. 비밀 값(S)은 스마트폰 내부 유심에 암호화 되어 저장되어 있고, 외부로 평문 S 자체가 전송되는 경우는 없다. 따라서 비밀 값(S)을 복호화하려면, 공격자는 피해자의 스마트폰을 입수해야 하며 또한 피해자의 패스프레이즈(P)를 알아야 한다. 더불어 피해자가 서버에 등록해놓은 위치정보(L)를 알아내서 그 위치로 직접 가거나, GPS 데이터를 조작해야 한다. 물론 피해자가 위치정보를 등록하지 않았다면, 이 과정은 필요치 않다. 언급한 이러한 모든 가정이 동시에 만족되는 상황에서만 공격자의 공격이 성공할 수 있다. 추가로, 스마트폰을 분실하거나 도난당했음을 확인한 피해자는 스마트폰 분실신고를 하여 해당 폰의 유심 사용을 정지시킬 수 있으며, 이런 방법으로 유심에 저장되어 있는 비밀 값(S)을 스마트폰 습득자가 획득할 수 없도록 막을 수 있다.



(a) 입력 전 (b) 입력 후

그림 5. 키 하나 입력 전 후의 키보드 배치
Fig. 5. Keyboard Layout Before and After a Key Entry

제안 방식은 응답을 받은 그대로 서버에 전달하는 대신에 비밀 값(S)과 현재 위치 정보(L)를 포함시켜 암호화 가공을 거친 응답을 서버에 전달한다. 따라서 제안 방식은 기존 SMS 기반 인증 방식과 달리, 피해자를 대신하여 몰래 인증을 시도하는 공격 시나리오에 대하여 매우 강력한 보안성을 제공한다.

또한 제안 방식은 스마트폰이 악성코드에 감염되는 경우에도 매우 높은 수준의 보안성을 제공한다. 제안 방식에서 비밀 값(S)은 사용자가 기억하는 패스프레이즈(P)로부터 유도된 암호화 키로 대칭키 암호화된 채 유심에 저장된다. 그런데 패스프레이즈(P)가 악성코드의 일종인 키로거 (keylogger)[20]에 의해 공격자에게 노출되었다면, 공격자가 피해자의 스마트폰을 손에 넣는 경우에 훔친 패스프레이즈(P)를 이용하여 비밀 값(S)을 얻을 수 있다. 이를 방지하여 악성코드가 패스프레이즈(P)를 획득하기 매우 어렵게 만들기 위해, 제안 방식 구현 시, 화면 캡쳐 방지 기능과 키 입력 시 키보드 심볼들의 배치가 변경되는 보안 키보드를 포함시켰다. 물론 악성코드가 안드로이드 운영체제의 모든 제어 권한을 획득하고 메모리까지 조작 가능하다면, 제안 방식의 보호 장치들을 모두 해제하고 공격을 가할 수 있다. 하지만 그런 공격은 피해자가 스마트폰 보안 설정을 해제하여 악성 코드의 침투를 가능하게 하였으며, 스마트폰이 감염되었다는 것을 모르는 상태에서 제안하는 인증 앱을 구동시키는 경우에만 가해질 수 있기 때문에, 실제 공격 성공 가능성은 매우 낮다고 판단된다.

5.2 사용 편의성 분석

제안하는 인증 앱은 보안 키보드를 이용한 패스프레이즈 및 응답 6글자를 사용자가 입력하도록 요구한다. 즉 기존 SMS 기반 인증 방식에서 사용자가 SMS 인증번호 최대 6글자를 입력하는 것 보다 산술적으로 부담이 늘어난 것이다. 하지만 추가적으로 입력해야 하는 패스프레이즈 및 캡챠를 통한 응답 입력이 대다수의 인터넷 사용자들에게 상당히 익숙하다는 점과 이러한 인증이 자주 이루어지지 않는다는 점을 고려한다면, 제안 방식은 획기적으로 향상된 보안성을 제공받기 위해 감내할 수 있는 수준의 추가부담을 지운다고 볼 수 있다.

Ⅵ. 결 론

본 논문에서는 SMS 기반 인증을 대체하는 인증 방식을 제안하고 프로토타입 구현을 통해 실현가능성을 보였다. 제안 방식이 공격자에 의해 깨지려면 다음과 같은 극단적인 가정이 필요하다. 공격자는 (1) 스마트폰 분실 신고가 접수되기 전에 공격 대상자의 스마트폰을 소유하면서, (2) 획득한 스마트폰 인증 앱의 패스프레이즈(P)를 알아내거나, 또는 유심에 저장된 암호화된 비밀 값(S)을 스마트폰 해킹 등으로 알아내고, (3) 피해자가 사전에 서버에 등록한(하나 이상 정보를 등록했다 할 때) 위치정보(L)를 알아내어 실제 그 위치에서 공격을 시도하거나, GPS 값을 해당 위치정보로 조작해야만 한다. 제안 방식은 기존 SMS 기반 인증과 마찬가지로, 피해자가 주 인증 수단으로 이미 인증을 받은 상태이거나, 주 인증 수단의 사용이 불가능한 상황에서의 보조 인증 수단으로 주로 사용된다. 따라서 제안 방식에 대한 공격을 성공시키기 위해서는 공격자가 주 인증 수단에 사용되는 피해자의 인증 정보를 탈취했거나, 주 인증 수단을 통한 사용자의 인증을 무력화시켰다는 전제가 추가적으로 필요하다. 결과적으로 제안 방식은 사용자에게 패스프레이즈 입력이라는 충분히 감당할 만한 부담을 추가로 지우면서, 기존 SMS 기반 인증에 비교하여 그 보안성을 크게 높였다는 점에서 그 의미를 찾을 수 있다.

References

[1] B. Schneier, "Two-Factor Authentication: Too Little, Too Late," *Commun. ACM*, vol. 48, no. 4, p. 136, Apr. 2005.

[2] AhnLab, *Alert Smartphone malware to small sum settlement*(2013), Retrieved January 11, 2013, from http://blog.ahnlab.com/ahnlab/1680.

[3] BBC, *Telegram denies Iranian mass breach*(2016), Retrieved August 3, 2016, from http://www.bbc.com/news/36964075.

[4] D. J. Seo and T. S. Kim, "Influence of personal information security vulnerabilities and perceived usefulness on bank customers' willingness to stay," *J. KICS*, vol. 40, no. 8, pp. 1577-1587, Aug. 2015.

[5] NIST(National Institute of Standards and Technology), *DRAFT NIST Special Publication 800-63B Digital Authentication Guideline*(2016), Retrieved May 18, 2016, from https://pages.nist.gov/800-63- 3/sp800-63 b.html.

[6] D. Strobel, "IMSI Catcher," *Chair for Commun. Secur.*, Jul. 2007.

[7] R. Bott and J. Frick, *Method for identifying the user of a mobile telephone or for eavesdropping on outgoing calls*, Patent EP1051053 A3, 2001.

[8] D. W. Park and J. M. Seo, "A study of information leakage prevention through certified authentication in phishing, vishing, SMiShing attacks," *J. The Korea Soc. Comput. Inf.*, vol. 12, no. 2, pp. 171-180, Jun. 2007.

[9] H. H. Kim and M. J. Choi, "Android malware detection using auto-regressive moving-average model," *J. KICS*, vol. 40, no. 8, pp. 1551-1559, Aug. 2015.

[10] A. Varghese and D. Mathews, "Securing SMS-based approach for two factor authentication," *J. Comput. and Commun. Technol.*, vol. 3, no. 3, pp. 25-28, Mar. 2014.

[11] S. S. Ji, "The improved scheme of two factor authentication using SMS," *J. Korea Ind. Inf. Syst. Res.*, vol. 17, no. 6, pp. 25-30, Dec. 2012.

[12] S. T. Ahmed and L. E. George, "Secure messaging system over GSM based on third party support," *IJEIT*, vol. 4, no. 2, pp. 27-32, 2014.

[13] S. T. Ahmed and L. E. George, "Secure SMS based on internet service," *Int. J. Comput. Sci. Mob. Comput.*, vol. 3, no. 10, pp. 164-171, 2014.

[14] M. AiZomai, A. Jøsang, A. McCullagh, and E. Foo, "Strengthening SMS-Based authentication through usability," *Int. Symp. Parall. and Distrib. Process. with Appl.*, pp. 683-688, 2008.

[15] J. Y. Park, J. I. Kim, M. S. Shin, and N. H. Kang, "QR-code based mutual authentication system for web service," *J. KICS*, vol. 39B, no. 04, pp. 207-215, Apr. 2014.

[16] S. H. Lee, H. Kim, and D. H. Lee,

"Two-factor authentication scheme based on mobile messenger with improved usability," *J. Secur. Eng.*, vol. 10, no. 5, pp. 549-566, Oct. 2013.

[17] U. A. Abdurrahman, M. Kaiiali, and J. Muhammad, "A new mobile-based multi-factor authentication scheme using pre-shared number, GPS location and time stamp," *ICECCO*, pp. 293-296, 2013.

[18] KISA, *2015 Survey on the Mobile Internet Usage Executive Report*, p. 138, 2016.

[19] H. Wu, "A new stream cipher HC-256," *Int. Wksp Fast Softw. Encryption*, pp. 226-244, 2004.

[20] F. Mohsen and M. Shehab, "Android keylogging threat," *9th IEEE Int. Conf. Collaborative Computing: Netw., Appl. and Worksharing*, pp. 545-552, 2013.

권 성 재 (Seong-Jae Kwon)

2015년 2월 : 홍익대학교 컴퓨터공학과 졸업
2017년 2월 : 홍익대학교 컴퓨터공학과 석사 졸업
현재 : NHN 엔터테인먼트(인프라 보안) 사원
<관심분야> 시스템/인프라 보안

박 준 철 (Jun-Cheol Park)

1986년 : 서울대학교 계산통계학과
1988년 : KAIST 전산학과 석사
1998년 : U. of Maryland, College Park, 전산학 박사
현재 : 홍익대학교 컴퓨터공학과 교수
<관심분야> 네트워크/시스템 보안