http://dx.doi.org/10.14400/JDPM.2013.11.12.423

모바일 환경에서 안전한 일회용 패스워드 인증

김동률
동명대학교 메카트로닉스공학과

Secure One-Time Password Authentication in Mobile Environments

Dong-Ryool Kim
Dept. of Mechatronics Engineering, Tongmyong University

요 약 인터넷을 이용한 전자상거래 및 금융 분야가 활성화되어 사용자와 서비스 제공자들 간의 상호 인증이 매우 중요해졌다. ID와 패스워드 기반의 인증은 보안성이 낮기 때문에 일회용 패스워드 인증방식이 많이 사용되고 있다. 기존의 일회용 패스워드 인증방식인 S/Key 인증방식은 평문 전송 외에 여러 문제점이 있고, 김홍기 등의 방식은 세션 키의 생성 및 분배 방법에 관한 제시가 없다는 문제점이 있다. 본 논문에서는 이러한 문제점을 해결하기 위한 프로토콜을 제안하였다.

주제어 : 일회용패스워드, 인증, 보안, 스마트폰, 해시함수

Abstract With the active Internet e-commerce and the financial sector, mutual authentication between users and service providers has become very important. Because ID- and password-based authentication is of low security, one-time password authentication methods are widely used. The existing one-time password authentication scheme of S/Key authentication method is fraught with a number of issues in addition to plain text transmission, and the method of Kim Gong-ki et al. does not offer suggestions for session key generation and distribution method. Proposed in this paper is a protocol that solves these problems.

Key Words : One Time Password, Authentication, Security, Smart Phone, Hash Function

1. 서론

인터넷은 불완전한 개방형 네트워크이기 때문에 물리적인 보안뿐 아니라 데이터, 통신 및 전자상거래를 보호하기 위한 추가적인 소프트웨어적인 보안이 필요하다. 특히 인터넷을 이용한 전자상거래와 전자금융 분야가 활성화됨에 따라 사용자와 서비스 제공자 사이의 상호 식별과 인증이 매우 중요하게 되었다[1].

기존의 인증방식은 ID와 패스워드를 사용한 방식이고 간편하게 사용할 수 있기 때문에 많은 시스템에서 사용된다. 사용자는 한번 사용한 패스워드를 주기적인 변경 없이 영구적으로 사용하는 경우가 많다. 또한 편리성 때문에 길이가 짧고 간단한 기억하기 쉬운 정보들을 사용하는 경우가 많아 보안성이 낮다. 그래서 아주 긴 패스워드를 사용하거나 매번 패스워드를 변경하는 것과 같은 방법들을 사용하여 보안성을 높인다. 아주 긴 패스워드는 사용자가 기억하기에 불편하므로 실용성이 낮기 때문에 매번 패스워드를 바꿔서 인증을 하는 일회용 패스워드가 요구된다. 일회용 패스워드의 특징 중 하나는 패스워드를 생성하기 위해 토큰이 필요하다. 토큰으로부터

Received 1 December 2013, Revised 20 December 2013
Accepted 20 December 2013
Corresponding Author: Dong-Ryool Kim(Tongmyong University)
Email: drkim@tu.ac.kr

ISSN: 1738-1916

일회용 패스워드를 생성하는 매체로 OTP 전용기기를 사용하면 기기 구매비용이 들고 휴대해야하는 불편함이 있다. 일회용 패스워드를 생성하는 토큰 중에서 카드형 OTP를 사용할 경우 휴대성 측면에는 OTP 전용기기에 비해 유리하지만 비용적인 면을 고려하면 부적합하다[2,3].

최근 스마트폰을 이용한 인터넷 서비스 및 애플리케이션이 급증하고 있다. 따라서 기존의 일회용 패스워드를 생성하는 토큰 대신 스마트폰의 애플리케이션으로 일회용 패스워드를 생성하는 것이 편리성, 휴대성, 비용적인 면에서 효율적이다. 스마트폰은 전화 및 다양한 서비스를 제공하기 때문에 많은 사람들이 이용한다. 또한 사용자는 스마트폰에 애플리케이션을 설치하면 OTP를 생성하여 사용할 수 있기 때문에 OTP 전용기기를 구입하지 않아도 되는 장점이 있다. 그러나 OTP 인증을 위한 전용 토큰에 비해 스마트폰과 같은 모바일 기기는 OTP 전용 토큰이 아니기 때문에 해킹의 위험이 높고, 키를 관리하는데 취약점이 있다. 이러한 문제점을 보완하기 위해 안전하고 효율적인 OTP 인증 방식에 관한 연구가 필요하다.

김홍기[4] 등은 기존의 S/Key 인증방식[5,6]이 전송과정에서 정보가 노출되고 일회용 패스워드의 순차적인 사용으로 인해 제3자의 공격에 취약점이 있다. 이를 개선하기 위해 시간 값을 통해 임의성을 강화한 인증방식을 제안하였다. 김홍기 등이 제안한 방식에는 전송 정보에 필요한 키를 관리하는 방법과 구체적인 알고리즘을 제시하지 않았다. 본 논문에서는 키의 관리와 구체적인 알고리즘을 사용하여 모바일 환경에 알맞은 안전한 일회용 패스워드 인증방식을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 분석하고 3장에서는 모바일 환경에서 안전한 일회용 패스워드 인증방식에 대한 프로토콜을 제안한다. 그리고 4장에서는 제안한 방법에 대하여 분석하고, 마지막으로 5장에서 결론을 맺는다.

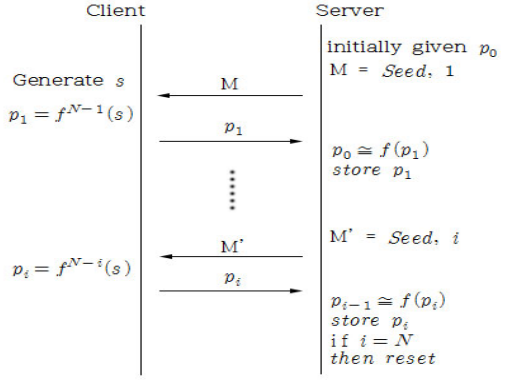
2. 관련연구

ID, 패스워드 기반의 인증방식은 사용자 편리성을 위해 짧은 길이의 패스워드를 사용하거나 기억하기 쉬운 간단한 단어를 오랫동안 변경하지 않고 사용하는 등의 문제점이 있다. 그래서 인증할 때마다 매번 바뀌는 일회용 패스워드를 이용하여 개체를 인증하기 위한 방식이 제안되었다. 본 장에서는 일회용 패스워드 인증방식에 대해서 분석한다.

일회용 패스워드 인증방식에는 RFC 표준인 일회용 패스워드 인증방식인 S/Key 인증방식[5,6]과 김홍기[4] 등이 제안한 인증방식에 대해 분석한다.

2.1 S/Key 인증방식

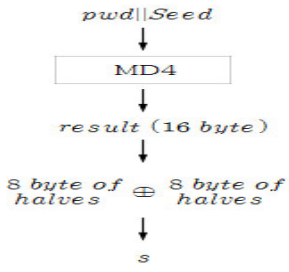
RFC 1760 표준인 S/Key 인증방식[5][6]은 해시함수 MD4를 이용하여 일회용 패스워드를 생성한다. S/Key 인증방식은 준비 과정, 일회용 패스워드 생성 과정 그리고 일회용 패스워드 검증 과정으로 구성된다. [그림 1]은 S/Key 인증방식이다.



[Fig. 1] S/Key Authentication

Step1. 준비 과정

S/Key 인증방식은 일회용 패스워드를 생성하는데 비밀 패스워드(Secret password) s의 크기가 8 바이트 정도 되어야한다. 비밀 패스워드 s를 생성하기 위해서 준비 과정이 필요하다. 이 과정에서 클라이언트는 패스워드와 $Seed$을 연접한 값을 MD4 해시함수의 입력 값으로 사용한다. 출력값 16바이트를 8바이트씩 나누어 배타적 논리합 연산한 결과인 8바이트를 비밀 패스워드 s로 한다.



[Fig. 2] Preparatory

Step2. 생성 과정

비밀 패스워드 s을 사용한 일회용 패스워드 생성 과정은 다음과 같다. 준비과정에서 클라이언트가 생성한 비밀 패스워드 s와 서버가 지정한 횟수 N-회 만큼 해시하여 해시값을 생성한다.

$$p_0 = f^N(s) \tag{1}$$

클라이언트는 [그림 3]과 같이 일회용 패스워드를 생성하기 위해서 지정된 횟수보다 1회 적게 해시한 값을 사용한다. 즉, i-번째 일회용 패스워드를 일반적인 수식으로 나타내면 (2)과 같다.

$$p_i = f^{N-i}(s) \tag{2}$$

$$p_0 = f^N(s)\ p_1 = f^{N-1}(s)\ p_2 = f^{N-2}(s)$$
$$p_3 = f^{N-3}(s)\ p_4 = f^{N-4}(s)\ p_5 = f^{N-5}(s)$$
$$p_6 = f^{N-6}(s)\ p_7 = f^{N-7}(s)\ p_8 = f^{N-8}(s)$$
$$\vdots$$
$$p_{i-2} = f^{N-i+2}(s)\ p_{i-1} = f^{N-i+1}(s)\ p_i = f^{N-i}(s)$$

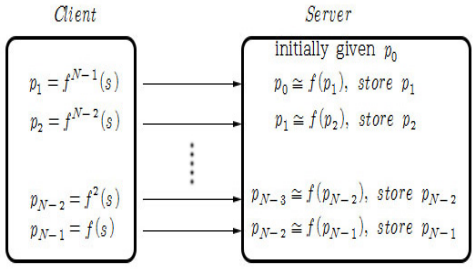
[Fig. 3] Generator

Step3. 검증 과정

일회용 패스워드의 검증 과정은 생성 과정에서 생성한 일회용 패스워드를 사용하여 검증한다. 클라이언트가 인증을 요청할 때, 서버는 해시할 횟수에 필요한 정보 i을 클라이언트에게 전송한다. 또한 서버는 클라이언트의 일회용 패스워드를 검증하기 위해서 일회용 패스워드(p_{i-1})를 저장하고 있다. 서버로부터 수신한 정보 i와 비밀 패스워드 s을 이용하여 $N-i$번 해시한 값을 클라이언트의 일회용 패스워드(p_i)라 하고 그 값을 서버에게 전송한다. 서버는 수신한 일회용 패스워드(p_i)를 한 번 더 해시한 값과 저장하고 있는 일회용 패스워드(p_{i-1})를 비교한다. 일회용 패스워드를 비교하는 일반적인 수식은 (3)과 같다.

$$p_{i-1} = f(f^{N-i-1}(s)) = f(p_i) \tag{3}$$

비교한 결과 일회용 패스워드가 일치하지 않으면 인증이 실패하고, 일치하면 서버는 일회용 패스워드 p_{i-1}을 p_i로 해시할 횟수에 필요한 정보 i을 $i+1$로 저장한다. [그림 4]는 일회용 패스워드의 수식 (3)에 의하여 인증을 거듭할수록 해시 함수의 반복 횟수가 줄어들고 i는 서버가 지정한 횟수 N과 일치하게 된다. 따라서 i와 N이 일치하게 되면 클라이언트와 서버는 N을 초기화한다.

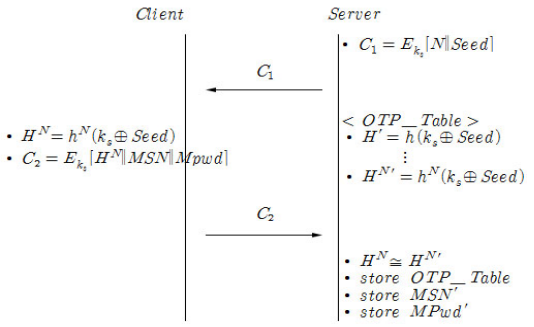


[Fig. 4] Verification

2.2 시간값을 통해 임의성을 강화한 인증방식

S/Key 인증방식은 초기화한 일회용 패스워드 테이블의 값이 순차적으로 기록되어 있어 $Seed$가 노출될 시 모든 일회용 패스워드 테이블이 노출당하는 문제점이 있다. 또한 초기 $Seed$ 전송 시 별도의 암호화 단계를 수행하지 않아 $Seed$가 노출될 위험이 매우 크다. 김홍기[4] 등은 이러한 문제점을 개선하기 위해서 시간값을 이용하여 생성된 일회용 패스워드 테이블을 임의적으로 사용하는 방식에 대하여 제안했다. 이 방식은 등록 단계와 인증 및 로그인 단계로 구성된다.

등록 단계에서는 서버가 전체 로그인 횟수(N)와 초기 $Seed$을 사전에 클라이언트와 공유한 세션 키(k_s)로 암호화하여 클라이언트에게 전송한다. 클라이언트는 수신한 메시지(C_1)로부터 N과 $Seed$을 구한다. 그리고 k_s와 $Seed$을 XOR 연산한 결과를 N번 해시한 값(H^N), 모바일 시리얼 넘버(MSN)와 모바일 패스워드($MPwd$)를 k_s로 암호화하여 서버에게 전송한다. 서버는 수신한 메시지(C_2)로부터 해시값(H^N)과 서버에서 생성한 해시값($H^{N'}$)을 비교한 후, 일치하면 일회용 패스워드 테이블, MSN과 $MPwd$을 저장하고, 일치하지 않으면 저장하지 않는다. 이와 같은 과정은 [그림 5]와 같다.

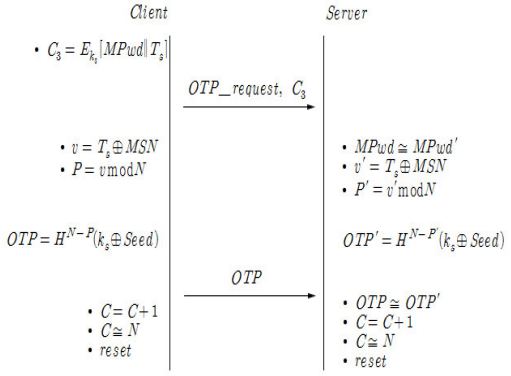


[Fig. 5] Registration

등록 단계가 끝나면 클라이언트는 서버에 인증 및 로그인을 시도하기 위해 다음과 같이 진행된다. 먼저, 클라이언트는 $MPwd$와 시간 동기화 값(T_s)를 k_s로 암호화한다. 그리고 암호문(C_3)과 일회용 패스워드 요청 메시지($OTP\_request$)를 서버에게 전송한다.

서버는 수신한 메시지(C_3)로부터 $MPwd$와 T_s을 구한다. 그리고 $MPwd$와 등록 단계에서 저장한 $MPwd'$을 비교한 후, 일치하면 T_s와 MSN을 XOR 연산한다. 연산한 결과 값(v')을 전체 로그인 횟수(N)로 나눈 나머지 P'을 계산한다. 클라이언트도 서버와 같은 방법으로 P을 계산한다.

그리고 클라이언트는 P을 이용하여 일회용 패스워드(OTP)를 구한 후, 서버에게 전송한다. 서버는 수신한 OTP와 P'을 이용하여 OTP'을 비교한다. 서버는 비교한 결과가 일치할 경우, 클라이언트를 인증 및 로그인하고 서버와 클라이언트는 동시에 카운트 값(C)을 1회 증가시킨 뒤 N과 비교하여 같으면, 일회용 패스워드를 초기화한다. 이와 같은 과정은 [그림 6]과 같다.



[Fig. 6] Authentication and Login

2.3 기존의 OTP 방식의 문제점

S/Key 인증방식은 모든 값이 평문으로 전송되어 제3자에게 쉽게 노출된다. 또한 서버의 난수인 $Seed$가 하나의 일회용 패스워드 테이블이 사용될 동안 동일하게 유지되고 있기 때문에 인증 차례를 나타내는 값 i가 노출되면 제3자는 다음번 일회용 패스워드 값을 유추할 수 있다. 그리고 MD4 해시 함수는 이미 취약점이 있다고 밝혀졌다. 또한 안전하지 않은 채널을 통해서 일회용 패스워드가 전송되므로 암호화가 필요하지만 평문으로 전송하고 있다.

김홍기[4] 등의 방식은 전체 로그인 횟수(N)와 $Seed$을 세션 키(k_s)로 암호화하여 전송하기 때문에 N과 $Seed$의 노출을 방지할 수 있다. 그리고 해시 함수와 세션 키 암호화를 함께 사용해 안전성이 강화되었다. 그러나 암호화에 사용하는 세션 키에 대한 생성 및 분배 방법을 명확하게 제시하지 않았으며, 인증 및 로그인 단계에서 $v = T_s \oplus MSN$와 같은 불필요한 연산을 수행한다.

본 논문에서는 위와 같은 방법의 취약점을 개선하기 위해서 세션 키에 대한 생성 및 분배 방법을 제시하고 XOR 연산량을 줄임으로서 기존 방식보다 효율적이다.

2.4 보안위협

일회용 패스워드를 사용하는데 있어 발생할 수 있는

보안위협은 다음과 같다[7][8].

첫 번째, 제3자가 인증 요청을 성공하기 위해 클라이언트와 서버간의 인증 정보를 악의적인 목적으로 도청할 수 있고 인증 정보를 탈취하여 불법 인증을 시도할 수 있다. 이러한 위협은 도청 및 탈취 공격(Eavesdropping and Extortion attack)에 의해 발생한다.

두 번째, 제3자가 인증 정보를 정당한 사용자의 것처럼 위장 등록하여 향후 전송되는 인증 정보를 통해서 불법 인증을 한다. 중간자 공격과 다른 점은 직접 제3자가 클라이언트인 것처럼 인증 정보를 등록한 것이다. 이러한 위협은 위장 공격(Disguise attack)에 의해 발생한다.

세 번째, 제3자가 서버에게 클라이언트인 것처럼 위장하거나 클라이언트에게 서버인 것처럼 위장하여 전송되는 인증 정보를 가로채 위조 및 변조하여 위장할 수 있다. 위장 공격과는 다르게 클라이언트와 서버 사이에서 인증 정보를 가로채는 과정을 통해 불법 인증 및 인증 정보를 탈취 할 수 있다. 이러한 위협은 중간자 공격(Man In Middle attack)에 의해 발생한다.

네 번째, 제3자가 도청 및 탈취, 위장, 중간자 공격 등을 통해 클라이언트의 인증 정보를 획득한 뒤 서버에게 재전송하여 클라이언트로 위장하여 원하는 정보를 얻거나 불법 인증을 시도할 수 있다. 이러한 위협은 재전송 공격(Reply attack)에 의해 발생한다.

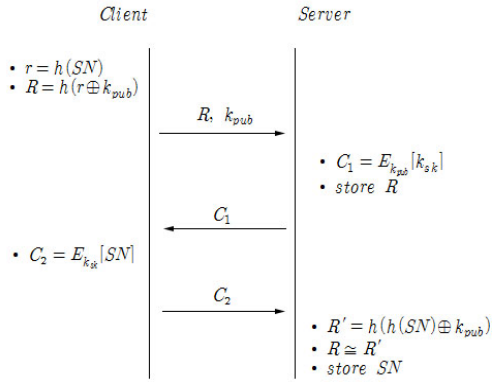
다섯 번째, 제3자가 인증 정보를 찾아내기 위해서 반복적인 검증시도를 통해 일회용 패스워드를 유도해낸다. 이러한 위협은 추측 공격(Guessing attack)에 의해 발생한다.

3. 모바일 환경에서 안전한 일회용 패스워드 인증 프로토콜

본 장에서는 2장에서 분석한 S/Key 인증방식[5,6]에서 암호화가 없고, 생성된 해시값들이 순차적으로 사용되는 문제점과 김홍기[4] 등이 제안한 인증방식의 키 생성 및 분배 방법에 대해 구체적인 제시가 없는 문제점 등을 해결하기 위해서 안전한 일회용 패스워드 인증 프로토콜을 제안한다. 제안한 프로토콜은 세션 키 생성 및 공유 단계, 일회용 패스워드 등록 단계, 인증 및 로그인 단계와 같이 3단계로 구성되어 있다.

3.1 세션 키 생성 및 공유 단계

세션 키를 생성 및 공유 단계에서는 모바일 기기에 대한 정보를 서버에 등록하는 것과 동시에 생성한 세션 키를 공유하므로 일종의 계정등록 역할을 겸한다. 모든 단계에서 사용하는 해시함수는 안전성이 뛰어난 SHA-512를 사용한다. [그림 7]은 세션 키 생성 및 공유 단계에 대한 프로토콜이다.



[Fig. 7] Session Key and Share

Step 1 : 클라이언트는 키 쌍(k_{pri}, k_{pub})을 생성하고, 모바일 기기의 고유번호(SN)를 추출한다. SN을 해시한 값 r과 공개키(k_{pub})를 XOR 연산하여 해시한 값 R과 k_{pub}을 서버에게 전송한다.

Step 2 : 서버는 세션 키(k_{sk})를 생성한다. 클라이언트로부터 수신한 k_{pub}을 사용해 k_{sk}을 암호화한 암호문(C_1)을 클라이언트에게 전송한다. 그리고 클라이언트의 k_{pub}을 검증하기 위해 R을 저장한다.

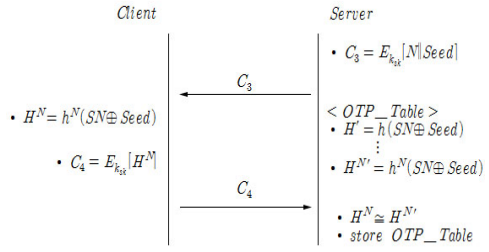
Step 3 : 클라이언트는 서버로부터 수신한 메시지(C_1)를 개인키(k_{pri})로 복호하여 k_{sk}을 저장한다. 그리고 고유번호를 k_{sk}을 사용하여 암호화한 암호문(C_2)을 서버로 전송한다.

Step 4 : 서버는 클라이언트로부터 수신한 메시지(C_2)를 복호하여 획득한 고유번호와 k_{pub}를 사용해 R'를 생

성하고 저장된 R와 비교한다. 두 값이 일치할 경우 고유번호를 저장한다.

3.2 일회용 패스워드 등록 단계

클라이언트는 서버에서 제공한 정보를 가지고 생성한 해시체인 값을 서버에게 전송하여 비교한다. 서버는 비교한 값이 일치하면 생성한 해시 값들을 저장하는 것으로 일회용 패스워드를 등록한다. 이러한 일회용 패스워드 등록 단계는 다음과 같다.



[Fig. 8] O TP Registration

Step 1 : 클라이언트는 서버에게 OTP 등록을 요청한다.

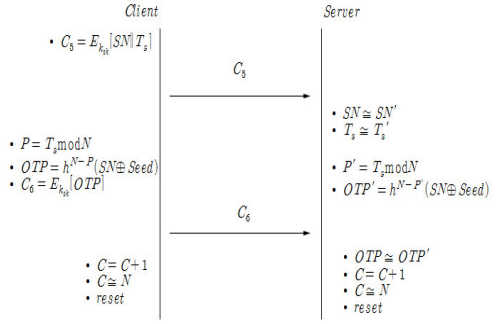
Step 2 : 서버는 전체 로그인 횟수를 결정하는 인증 횟수 N과 임의의 수 $Seed$을 생성한다. 그리고 N과 $Seed$을 연접하여 세션 키(k_{sk})를 사용해 암호화한 암호문(C_3)를 클라이언트에게 전송한다.

Step 3 : 서버는 고유번호(SN)와 $Seed$을 XOR 연산하여 각각 1~ N번 해시한 값을 일회용 패스워드 테이블로 만든다. 클라이언트는 수신한 메시지(C_3)로부터 N과 $Seed$을 복호한다. 그리고 SN과 $Seed$을 XOR 연산한 결과를 N번 해시한 값(H^N)을 k_{sk}을 사용해 암호화하고 서버로 전송한다.

Step 4 : 서버는 클라이언트로부터 수신한 메시지(C_4)로부터 해시 값(H^N)을 얻는다. 미리 생성해 놓은 일회용 패스워드 테이블의 해시 값($H^{N'}$)과 클라이언트로부터 수신한 해시 값(H^N)을 비교하여 일치할 경우, 서버는 일회용 패스워드 테이블을 저장한다.

3.3 인증 및 로그인 단계

일회용 패스워드 등록 단계가 완료된 후 클라이언트는 서버에게 인증 및 로그인을 시도한다. 이러한 과정은 다음과 같다.



[Fig. 9] Authentication and Login

Step 1 : 클라이언트는 시간 동기화 값(T_s)을 생성한다. 그리고 고유번호(SN)와 T_s을 연접한 뒤, 세션 키(k_{sk})로 암호화한 암호문(C_5)을 서버에게 전송한다.

Step 2 : 서버는 수신한 메시지(C_5)로부터 SN과 T_s을 복호한다. 그리고 서버가 저장해 놓은 고유번호(SN')와 클라이언트로부터 수신한 SN, 그리고 서버의 시간 동기화 값(T_s')와 수신한 T_s을 비교하고, 일치하는 경우 T_s을 N으로 나눈 나머지 값인 P'을 생성한다. 클라이언트도 동일한 방법으로 P을 생성한다. 클라이언트는 P을 통해 구한 OTP을 암호화하여 서버에게 전송한다.

Step 3 : 서버는 클라이언트로부터 수신한 메시지(C_6)로부터 복호한 OTP와 계산한 P'을 통해 구한 OTP'를 비교하여 일치할 경우 인증 및 로그인을 승인한다. 그리고 서버와 클라이언트는 각각 카운트 값(C)을 증가시키고 N과 비교하여 로그인 횟수를 달성할 경우 일회용 패스워드 테이블을 초기화한다.

4. 안전성 및 효율성 분석

본 장에서는 제안한 프로토콜의 분석으로 2.4절의 보안위협에 대해 안전성과 성능적인 측면에 대한 효율성을

비교 분석한다.

4.1 안전성 분석

본 논문에서 제안한 프로토콜을 보안위협에 대해 안전성을 분석한다.

4.1.1 도청 및 탈취 공격

제3자는 안전하지 않은 채널을 도청할 수 있다. 그러나 전송 정보는 항상 암호화되어 전송되기 때문에 평문 정보는 노출되지 않는다. 또한 탈취를 하여도 암호화된 정보를 열어서 확인하기 위해서는 클라이언트와 서버 간의 세션 키(k_{sk})가 필요하다. k_{sk}가 노출되지 않는 이상은 암호화된 정보는 안전하다고 할 수 있다.

4.1.2 위장 공격

서버는 클라이언트와 세션 키 교환 단계에서 이미 클라이언트의 모바일 기기 고유번호(SN)를 저장하고 있다. 위장 공격을 시도하기 위해서는 위장하고자 하는 클라이언트의 SN을 우선적으로 알아야 한다. 그러나 SN은 암호화되어 전송되고 있으며, 해당 모바일 기기에 접근하지 않으면 얻기가 어렵다. 그래서 위장 공격은 시도하기 어렵다.

4.1.3 중간자 공격

중간자 공격을 위해서는 전송 정보를 도청 혹은 탈취, 위장 등의 방법으로 가로채야하는데 세션 키 공유단계에서 교환한 세션 키(k_{sk})로 정보를 암호화하기 때문에 k_{sk}를 모르면 평문 정보를 가로챌 수 없다. 제3자는 k_{sk}를 모르기 때문에 위조 또는 변조된 가짜 정보를 서버가 복호할 수 없으므로 위조 또는 변조는 어렵다.

4.1.4 재전송 공격

OTP를 전송 할 때 암호화하여 보내기 때문에 세션 키(k_{sk})를 모르면 일회용 패스워드를 입수할 수 없다. 또한 서버와 클라이언트는 로그인 및 인증을 시도할 때마다 시간 동기화 값(T_s)을 사용해 만든 $P(P')$에 의해 일회용 패스워드는 항상 변경되기 때문에 이미 사용한 패스워드를 재전송하여 인증하기란 어렵다.

4.1.5 추측 공격

매번 클라이언트와 서버는 암호 통신을 하기 때문에 일회용 패스워드 테이블의 추측을 위한 정보를 얻기가 힘들다. 일회용 패스워드 테이블이 노출되어도 시간 동기화 값(T_s)를 사용해 만든 $P(P')$에 의해 일회용 패스워드는 임의로 사용하기 때문에 규칙성이 없기 때문에 반복적인 검증시도로 일회용 패스워드를 추측하기 어렵다.

4.2 효율성 분석

연산량과 통신량의 비교는 <표 1>과 같다. 연산량은 클라이언트와 서버의 각 단계에서 연산별로 횟수를 합하

〈Table 1〉 Comparative Evaluation

	Operation	S/Key Authentication	Kim Gong-ki et al[4]	Proposed Scheme
Session Key and Share	Hash	0	0	4
	Modular	0	0	0
	Encryption	0	0	2
	XOR	0	0	2
OTP Registration	Hash	2N	2N	2N
	Modular	0	0	0
	Encryption	0	2	2
	XOR	2N	2N	2N
Authentication and Login	Hash	P	(N-P)	N-P
	Modular	0	2	2
	Encryption	0	1	2
	XOR	P	(N-P)+2	N-P
Traffic	Session Key and Share	0	0	3
	OTP Registration	2	2	2
	Authentication and Login	2	2	2

였다. 통신량은 클라이언트와 서버간의 각 단계의 통신 횟수를 합하였다. S/Key 인증방식과 김홍기 등의 방식은 키의 생성 및 분배 방법에 관한 구체적인 제시를 하지 않아서 비교할 수는 없지만 제안한 방식에서는 구체적인 방법을 제시했다. S/Key 인증 방식에서는 암호화를 하지 않았지만 김홍기 등의 방식과 제안방식은 암호화와 모듈러 연산을 하여 연산량은 증가하였지만 평문의 노출을 막고 임의성을 주어 상대적으로 안전성 측면에서 효율적이다. 김홍기 등의 방식과 제안방식은 암호화와 모듈러 연산에서 약간의 차이가 있지만 동등한 효율성을 보장할 수 있다.

5. 결론

일회용 패스워드 시스템의 특징 중 하나로는 패스워드를 생성하기 위한 토큰이 필요하다. 토큰으로는 OTP 전용기기, 카드형 OTP 등이 있지만 별도의 구매비용이나 휴대성 등의 측면에서 불편함이 따른다. 하지만 최근 스마트폰의 대중화로 스마트폰 자체의 인터넷 서비스 및 애플리케이션을 사용하여 기존의 일회용 패스워드 토큰 역할을 스마트폰의 애플리케이션으로도 수행할 수 있다. 스마트폰의 경우 휴대성이 좋으며 OTP를 위한 애플리케이션을 설치만 하면 사용할 수 있는 장점이 있다. 스마트폰은 OTP 전용 토큰이 아니기 때문에 해킹의 위험이 높고, 키 관리의 취약함이 있다. 이러한 문제점들이 있어 안전하고 효율적인 OTP 인증 방식에 관한 연구가 필요하여, 김홍기[4] 등은 기존의 S/Key 인증방식 [5,6]의 취약점을 개선하기 위해 시간 값을 통해 임의성을 강화한 인증방식을 제안하였다. 기존방식에는 키를 생성하고 공유하는 방법과 암호 알고리즘을 사용하는 것을 제시하지 않았다. 본 논문에서는 서버가 AES 암호 알고리즘을 사용하여 세션 키를 생성하고 RSA 암호 알고리즘을 사용하여 세션 키를 교환하는 방식을 제안하였다. 그리고 제안한 방식은 연산량을 효율적으로 줄이고 여러 가지 보안위협에 대하여 안전함을 보였다. 그러나 서버가 안전하다는 가정 하에 제시되었기 때문에 정당한 서버인지 검증하는 과정이 필요하다.

REFERENCES

[1] Yeon-Ho, Ryu, Cross Authentication Model for Client-Server by used OTP Concept, The Korean Institute of Information Scientists and Engineers, Vol.30, No.2 I , pp.652-654, 2003.

[2] S. D. Park, J. C. Na, Y. H. Kim, and D. K. Kim, Efficient OTP(One Time Password) Generation using AES-based MAC, Journal of Korea Multimedia Society, Vol.11, No.6, pp.845-851, 2008.

[3] Dong-hyun Choi, Seung-joo Kim, Dong-ho Won, One-Time Password Technique Analysis and Standardization Trends, Journal of Korea Institute of Information Security And Cryptology, Vol.17, No.3, pp.12-17, 2007.

[4] Hong Gi Kim, Im Yeong Lee, A Study on One-Time Password Authentication Scheme in Mobile Environment, Journal of Korea Multimedia Society, Vol.14, No.6, pp.785-793, 2011.

[5] Neil M. Haller, The S/KEY One-Time Password System, RFC 1760, 1995.

[6] N. M. Haller, C. Metz, P. Nesser, and M. Straw, A One-Time Password System, RFC 2289, 1998.

[7] J. Archer Harris, OPA : A One-Time Password System, 10.1109 / ICPPW, 2002, 1039708, 2002.

[8] Soo-Yong Kang, Im-Yeong Lee, A Study on Secure and Efficient OTP Authentication Scheme using Improved S/Key Scheme, Journal of Korea Multimedia Society, pp.109-112, 2007.

김 동 률(Kim, Dong Ryool)

- 2005년 3월 ~ 현재 : 동명대학교 메카트로닉스공학과 조교수
- 관심분야 : 암호이론, 정보보호, 모바일 보안
- E-Mail : drkim@tu.ac.kr