비밀키를 이용한 토큰 업데이트 보안 인증 기법

A secure token-updated authentication scheme using security key

준량(JunLiang)*, 장인주(Jang In Joo)**, 유형선(Hyeong Seon Yoo)***

초 록

최근 들어 OTP를 고려한 스마트카드 환경에서의 많은 인증 기법이 제안되고 있다. 그러나 이러한 인증 기법들이 개인 컴퓨터 환경에서 구현될 경우 개인 컴퓨터상에서는 데이터를 쉽게 읽어 내고, 유출 시킬 수 있기에 위장 공격이나 Stolen-Verifier 공격 등에 대하여 취약점이 노출된다. 본 논문에서는 이러한 취약점을 개선한 인증 기법을 제시함으로 스마트카드 상에서 사용되는 인증 기법이 컴퓨터 환경에서도 효율적으로 사용될 수 있도록 하였다. 본 논문이 제안하는 기법은 Stolen-Verifier 공격과 위장 공격에 안전하며, 상호 인증 기능을 부여하고 비밀 키의 생성에 있어 쉽다는 장점을 지니고 있다.

ABSTRACT

Recently, a large number of authentication schemes based on smart cards have been proposed, using the thinking of OTP (one-time password) to withstand replay attack. Unfortunately, if these schemes implement on PCs instead of smart cards, most of themcannot withstand impersonation attack and Stolen-Verifier attack since the data on PCs is easy to read and steal. In this paper, a secure authentication scheme based on a security key and a renewable token is proposed to implement on PCs. A comparison with other schemes demonstrates the proposed scheme has following merits: (1) Withstanding Stolen-Verifier attack (2) Withstanding Impersonation attack (3) Providing mutual authentication; (4) Easy to construct secure session keys.

키워드 : 인증, 보안, 토큰, OPT, 공격
Authentication, Security, Token, OPT, Attack

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원사업의 연구 결과로 수행되었음.

* Chengdu university of technology in China
** 인하대학교 컴퓨터정보공학
*** 인하대학교 컴퓨터 공학부 교수

1. 서 론

인증 프로토콜은 현대의 정보 보호 시스템에서 무선 접근을 위한 중요한 역할을 담당하고 있다. 이에 최근 많은 인증 기법들이 제안되고, 발전되어 왔다. 보안성과 효율성은 이러한 프로토콜에서 항상 중점적으로 고려되는 사항이다.

1981년 람폴트(Lamport)는 처음으로 일회성 패스워드(OTP)를 제안하였다[1]. 그의 논문에서 설명 되어진 것처럼 OTP는 접속 정보를 반복적으로 사용 하는 대신 일회만 사용하는 방식이다. 이 후로 많은 수의 효율적인 OTP기법들이 제안되었다. 2000년 에는 Sandirigama 등에 의해 단순하고 강력한 패스워드 인증 기법인 SAS가 제안 되었고[2], 일년 후 Lin등에 의해 최적화된 스트롱 패스워드인증기법(OSPA)이 제안되었다[3]. 그러나 이 시점 까지 제안된 기법들은 권한 도용 공격[4]이나 위장 공격[5]에 대하여 취약점을 보이고 있다. 이를 보안하고 극복하기 위한 다양한 기법들이 린(Lin)[6], 첸(Chen)[7], 윤(Yoon)[8]등에 의해 제안되었으며 그 기법들은 실제 스마트카드에서 적용하여 사용 되었다. 2005년 Lee 등은 토큰에 기반을 둔 컴퓨터 환경에서 사용될 인증 프로토콜을 제안하였다[9]. 그 뒤 랜 카드의 아이디 나 맥 어드레스를 이용한 인증 기법들이 제안되는 등 개선 발전된 기법들이 많이 연구되고 발표되어 왔다.

이 논문에서는 스마트카드에서 사용 되었던 기법을 토대로 토큰 업데이트 기법[2, 3, 4, 5]을 수정하였다. 본 논문에서 제안하는 기법은 토큰의 생성 시에 난수를 사용함으로써 네트워크 침범자들의 접근이나 능동적인 공격에 대하여 취약함을 보인 일회성 패스워드 시스템의 보안성을 강화시켰다. 이는 보안 분석을 통해 향상된 보안성을 확인하였다.

2. 제안된 토큰 업데이트 기법

스마트카드를 사용 할 때, 사용자는 그들의 접속 정보를 선택하고, 키 인포메이션 센터(KIC)를 통해 스마트카드 아이디(CID)를 얻는다. 이때 CID는 스마트카드의 롬에 저장이 된다. 그러나 토큰 업데이트 프로토 타입[9]에서 제시한 방법은 개인용 컴퓨터 랜 카드의 맥 주소를 사용함으로써 KIC의 기능을 대신 할 수 있도록 하였다. 여기서 일회성 패스워드의 역할을 하는 재생된 토큰은 사용자의 로그온 정보와 맥 주소로 생성된다. 그러나 불행히도 이 일회성 패스워드 시스템은 네트워크 침범자들의 접근이나 능동적인 공격에 대하여는 취약한 점을 보였다. 우선적으로 일회성 패스워드 시스템에서 고려되어야 할 보안 요소와 인증 시스템이 지녀야 하는 효율성에 대하여 살펴본 뒤 제안 하고자 하는 인증 기법을 제시하도록 하겠다.

네트워크 환경에서 일회성 패스워드 시스템은 우선적으로 다음의 요소들을 고려하여 설계되어야 한다[1,3,4].

● 위장 공격(Impersonation attack):

통신 프로토콜에 있어서 공격자가 적법

한 사용자의 아이디 중 하나를 성공적으로 얻어 사용하는 공격을 말한다.

- Stolen-Verifier 공격 :
  공격자가 서버로부터 검증 자를 훔쳐 마치 합법적인 사용자인 것처럼 위장하는 공격을 말한다.
- 상호 인증 (Mutual authentication):
  사용자와 서버가 통신이나 접속의 승인 이전에 각각을 서로 인증해 주는 과정은 말한다.

Chen[7]은 인증 기법을 암호를 기반으로 한 것과 해쉬함수를 기반으로 한 것의 두 가지 방법으로 설명하였다. 해쉬함수 기반의 기술은 SHA-1과 같이 충돌 회피 해쉬함수를 바탕으로 하며, 암호화에 기반을 둔 기술보다 구현에 있어 효과적이고 간단하다.

본 논문에서는 위에 언급한 해쉬함수 기반 기술 위에 개인용 컴퓨터상에서의 토큰 업데이터기술을 사용하여 인증을 수행하는 개선된 기법을 제안한다. 본 기법에서, 모든 사용자(U)는 데이터 보호를 위해 사용자가 개인적으로 선택한 비밀키 Y를 가지고 있다. 서버(S)는 사용자와 서버 간의 정보 공유에 사용하기 위해 서버의 비밀키 X를 가진다. 각 과정에서 사용자는 임의의 난수 N을 선택하고, 이 난수를 정보 변환에 사용한다. 정보 전달은 변환된 정보를 전달하는 것으로 이루어진다. 이렇게 전달받은 정보로부터 서버는 공유 정보로 사용할 사용자의 난수 N을 얻게 되며, 이 난수 N은 다시 정보에 삽입되어 사용자에게 전달된다. 위와 같은 과정으로 이루어지는 인증 시스템을 위한 본 논문의 제안 기법은 다음과 같다.

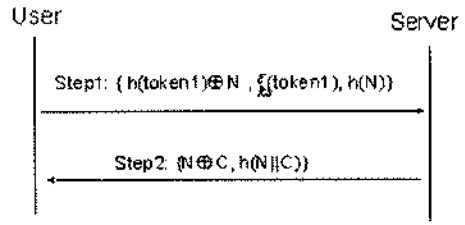
2.1 등록

다음의 〈그림 1〉은 본 논문에서 제안하는 기법의 등록 과정을 나타낸 것이다.

Step1: $U \rightarrow S$:

$$\{h(token1) \oplus, \boldsymbol{\varepsilon}_{ks}(token1), h(N)\}$$

우선적으로, 사용자(U)는 그의 아이디 id와 패스워드pw, 컴퓨터의 맥 어드레스인 ma로 $token1(token1 = (id \mid pw \mid ma))$을 만든다.



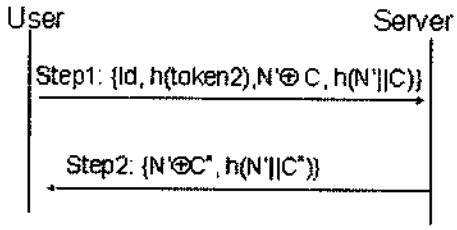
〈그림 1〉 제안 기법의 등록과정

선택한 난수 N은 데이터와 함께 사용되어 전달될 정보를 생성시킨다. 생성된 정보 $\{h(token1) \oplus, \boldsymbol{\varepsilon}_{ks}(token1), h(N)\}$는 서버에게 전달된다.

Step2: $S \rightarrow U$:$\{N \oplus C, h(N \mid C)\}$

데이터를 받은 뒤, S는 $\boldsymbol{\varepsilon}_{ks}(token1)$의 복호화 과정을 통해 $token1$을 얻게 된다. 그리고 수식 $\{h(token1) \oplus, h(token1) \oplus N\}$을 연산함으로써 난수 N을 찾아내고, 올바른 N인지를 확인한다. 서버S는 난수 rs를 선택하고 $C(:=h(rs \mid X))$를 계산한 뒤 사용자U에게 전송한다. 이때 서버는 난수 rs와 사용자의 $token1$을 저장해 둔다.

전송된 데이터를 받은 사용자는 계산식 $N \oplus N \oplus C$을 통하여 C를 얻어내고 $h(N \mid C)$를 계산하여 전송된 값과 비교함으로써 전송자의 신뢰성을 확인하게 된다.



〈그림 2〉 로그인과 인증과정

2.2 로그인과 인증

제안된 인증 기법의 로그인과 인증 과정은 〈그림 2〉와 같다.

Step1: $U \rightarrow S$:

$$\{id, h(token2), N \oplus C, h(N \mid C)\}$$

사용자U는 $token1$과 C값으로 $token2$를 생성하고, 새로운 난수 N를 선택한 후 값 $\{id, h(token2), N \oplus C, h(N \mid C)\}$을 서버에게 전송한다.

Step2: $S \rightarrow U$: $\{N \oplus C^*, h(N \mid C^*)\}$

전송된 데이터를 받은 후 서버S는 저장해 둔 rs값을 이용해 $C = h(rs \mid X)$값을 계산한다. 서버는 식 $N \oplus C \oplus C$을 통하여 N값을 얻어 내고 $h(token2)^*$와 $h(N \mid C)^*$을 생성한다. 서버는 $h(token2)^*$와 생성한값 $h(N \mid C)^*$을 전송 받은 값들과 비교하여 전송 자의 신뢰성을 확인한다. 확인 된 경우 서버는 새로운 난수 rs^*를 선택하고 서버의 비밀 키 X를 이용하여 새로운 공유 정보인

$C^* = h(rs^* \mid X)$를 생성한다.

이 후 서버는 값 $\{N \oplus C^*, h(N \mid C^*)\}$을 사용자에게 전송하고 다음 로그인 과정에서 사용하기 위해 rs를 rs'로 대체 저장한다. 이러한 과정의 반복으로 로그인에 사용될 공유 데이터는 계속 임의의 값으로 업데이트 되어 실행이 된다.

3. 토론과 분석

이제 본 논문에서 제안하는 기법의 보안성과 효율성을 살펴보고 다른 기법들과 비교하여 본다.

3.1 보안 분석

3.1.1 공격 분석

본 논문에서 제안하는 기법에 있어서 중요한 특성은 사용자와 서버가 모두 자신들만의 비밀 키 (Y,X)를 지닌다는 점이다. 각 비밀 키는 자유롭게 선택되어 보안이 유지되면서도 공유 정보의 생성과 복구에 사용된다. 서버 측에서 정보가 누출되었다 하여도 정보 획득자는 서버나 사용자의 비밀 키를 알 수 없어 C값를 얻을 수 없게 된다.

● 위장 공격 :

본 논문에서 제안한 기법에서 사용자는 전송받은 $h(N \mid C)$와 계산한 $h(N \mid C)^*$값을 비교하고, $N \oplus C$에서 C값을 추출해 냄으로써 난수 N의 유효성을 확인하게 된다. 따라서 공격자가 데이터 위조를 위해 필요한 식

$\{N \oplus C_f, h(N \mid C_f)\}$을 만들기 위해서는 서버와 사용자의 비밀 키가 필요하게 된다. 누출된 정보로부터 사용자와 서버 각각이 자유롭게 선택한 비밀 키의 정보를 밝힐 수 없기에 공격에 관하여 안전성을 보장한다.

● Stolen-Verifier 공격 :

제안된 기법에서, 공격자가 *token1*과 새로운 난수 rs, 그리고 서버의 비밀 키 X를 훔쳐냈다 하더라도 사용자의 비밀 키 Y를 알 수 없어 값 $h(token1 \mid h(rs \mid X))$의 계산을 불가능 하게 하여 안전성을 유지한다.

● Password-Guessing 공격 :

참고문헌[9]의 기법에서는 공격자가 정당한 사용자의 아이디 와 비밀 번호를 알고 있다 하더라도 맥 어드레스를 모르는 상태에서는 서버를 속일 수 없게 된다고 하였다. 그러나 맥 어드레스의 노출이 일어난 경우라 하더라도 임의로 정의된 난수 값을 얻지 못한 공격자는 사용자의 공유 정보를 복원시킬 수 없게 되므로 사용자의 비밀 번호와 맥 어드레스의 노출만으로는 로그인 과정을 성공시킬 수 없게 된다.

● Replay attack :

매 전송 때 마다 난수가 선택되고, 그 난수가 적용된 정보가 서버에게 전송된다. 이러한 전송과정은 공격자가 사용자의 전송 정보를 사용 할 수 없게 하며, 성공적인 로그인 후 서버는 새로운 *token2*생성하여 다음 로그인에 사용한다. 따라서 난수가 노출 된다하여도, 서버의 비밀키 X를 모르는 공격자는 *token2*생성할 수 없어 로그인을 성공시킬 수 없게 된다.

● 패킷 스니핑(Packet Sniffing) :

제안된 인증 기법 역시 공격자는 전송되는 패킷으로부터 정보를 얻을 수 없게 된다. 전송되는 정보는 모두 해쉬 함수 또는 암호 알고리즘에 의해 암호화 된 형태로 전송된다.

● 위조 공격(Forgery attack) :

매 로그인 때, 서버는 사용자와 *token2*의 유효성을 확인한다. 공유정보 C를 복원시키는데 사용되는 사용자와 서버의 비밀키가 없는 공격자는 사용자의 정보를 위조 할 수 없게 된다.

● 상호 인증 (Mutual authentication):

본 논문에서 제안하고자 하는 인증 기법은 상호 인증을 위해 매 통신 과정마다 사용자가 난수를 선택한다. 전송된 데이터 중 사용자는 난수를 포함하는 데이터를 검색하고, 사용자가 저장해 놓은 난수와의 일치성을 비교하여 정보의 유효성을 결정한다. 그리고 그 유효한 서버의 가부를 결정한다.

3.1.2 보안성의 비교

사용자와 서버간의 공유정보가 없는 Lee의 토큰 업데이트기법에서는 만일 공격자가 사용자의 공개키를 이용하여 공격자의 위조 데이터를 암호화하여 전송 한다면 사용자는 속임을 당하게 된다. 그리고 다음 번 로그인 과정에서 사용자가 서버에 로그인 하기 위해 이 위조된 데이터를 사용하게 되고 서버는 요청된 서비스 제공을 거절하게 된다. 이 기법은 Stolen-Verifier 공격에도 약하다. 만일 공격자가 서버 상의 $h(token2)$값을 훔친다면, 공격자는 사용자의 *id*로 쉽게 서버에 접속할

수 있다. 이 기법의 또 다른 약점은 사용자에 의하여 전송된 데이터가 서버에 의하여 인증되지만, 서버에 의하여 전송된 데이터는 사용자에 의하여 인증되지 않는다는 것이다. 이는 사용자와 서버 사이에서 공유되는 보안 정보를 지니고 있지 않기 때문이다. 이러한 기법과 유사한 기법[2,3,6,8]들이 시스템의 보안과 효율성을 위해 비밀키나 해쉬 함수, 난수 등을 이용하여 제안 되었다. 그러나 Sandrigama등이 제안한 단순 패스워드 인증 기법[2]이나, Lin등이 제안한 OSPA 프로토콜[3] 등은 위장 공격이나 Stolen-Verifier 공격에 대하여 취약하여 공격자의 공격에 대한 취약성을 지닐 뿐 아니라, 상호 인증의 기능도 제공하지 못한다. 이에 반해 본 논문에서 제안하는 기법은 보안성이 좋을 뿐 아니라 상호 인증의 기능까지 제공하는 효율적인 인증 기법이라 할 수 있다.

3.2 효율성 분석

3.2.1 효율성 비교

Chen[7]의 논문에서 해쉬 함수 기반 기술은 SHA-1 과 같이 충돌 회피 해쉬 함수에 기초하기에 암호화 기반 기술보다 구현에 있어 간단하며 효율적이라고 언급하였다. 그리고 대부분의 기법에 있어서, 로그인 과정과 인증 과정의 효율성은 그 높은 발생 빈도 때문에 중점적인 연구 문제가 된다. 본 논문에서 제안하는 기법에 대해서도 그 로그인 과정과 인증 과정의 효율성에 대해 이곳에서 논의하고자 한다. 본 논문이 제안하는 기법에서는 로그인과 인증 과정에 대해 사용자는 3회의 해싱 작업을 수행하며, 서버는 5회 해싱 작업을 수행한다. 이는 해싱 작업과 암,복호화 과정을 거치는 다른 기법[9]들과 비교 할 때 향상된 효율성을 지니게 된다[7,9].

〈표 1〉 기법들 간의 보안성과 효율성에 관한 비교

Scheme	a	b	c	d	e	f	g	h	i
SAS	5T(h)	Yes	No	No	Yes	Yes	Yes	Yes	Yes
OSPA	7T(h)	Yes	No	No	Yes	No	Yes	Yes	Yes
Lin et al's	8T(h)	Yes	No	No	Yes	Yes	No	No	Yes
E.J Yoon's	9T(h)	Yes	Yes	No	Yes	No	No	No	No
Lee et al's	3T(h)+1T(e)+1T(d)	Yes	No	No	Yes	No	No	Yes	Yes
our Scheme	8T(h)	Yes	Yes	No	Yes	No	No	No	No

T(h) : 해싱시간
T(e) : 암호화 시간
T(d) : 복호화 시간
a : 인증 과정의 계산 비용
b : 사용자의 패스워드 선택의 자유
c : 상호 인증
d : 클럭 동기화의 필요성
e : 저장 요구의 필요성
f : Guessing 공격
g : Replay 공격
h : Impersonation 공격
i : stolen verifier 공격

스마트카드와는 달리 PC 환경에서는 그 열린 환경 때문에 사용자 측의 저장 데이터가 임의의 공격자들에게 유출 될 가능성이 훨씬 높다. 그러나 등록 과정에서 사용자 측에서는 서버로부터 변환된 데이터를 얻게 되므로 암호화 기반 기술에 의하여 정보 보호의 강력한 수단이 제공된다. 즉 본 논문이 제안하는 기법은 공격자가 전송 정보를 불법적으로 획득했다 하더라도 공격자가 원하는 정보를 얻을 수 없게 되어 보안성을 잃지 않게 됨을 알 수 있다. 아래의 〈표 1〉은 본 논문의 제안 기법과 다른 기법과의 보안성과 효율성에 관한 비교 내용을 표로 간략화 시켜 정리한 것이다.

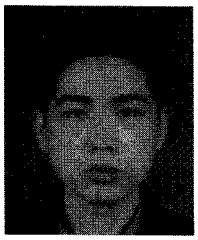
4. 결 론

본 논문에서는 PC상에서 이용되는 사이트 인증 기법으로 개선된 토큰 업데이트 기법을 제시하였다. 매 과정에서 서버는 새로운 공유 정보를 난수와 비밀키를 이용하여 생성하고, 이를 사용자에게 전송한다. 이때 사용자와 서버는 전달받은 데이터의 진, 위를 구분할 수 있다. 제안된 기법은 매 과정에서 난수를 이용하여 전달된 데이터를 새롭게 한다. 로그인과 인증 과정에서 데이터를 보호하기 위해 해쉬 함수를 사용한다. 그러므로 제안된 기법은 시스템의 연산 요구를 줄이면서도 상호 인증 기능을 제공하고 자주 발생하는 공격으로부터 시스템을 보호하는 향상된 인증 기법이다.

참 고 문 헌

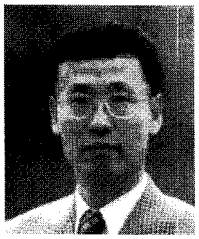
1. L. Lamport, "Password authentication with insecure communication," Communications of the ACM, 24, 770-772, 1981.
2. M. Sandirigama, A. Shimizu, M.T. Noda, "Simple and secure password authentication protocol (SAS)", IEICE Transactions on Communications, 83 (6), 1363-1365, 2000.
3. C.L. Lin, H.M. Sun, T. Hwang, "Attacks and solutions on strong- password authentication," IEICE Trans. on Communications, 84 (9), 2622-2627, 2001.
4. C.M. Chen, W.C. Ku, "Stolen-verifier attack on two new strong-password authentication protocols", IEICE Trans. on Communications, 85 (11), 2519-2521, 2002.
5. 5. T. Tsuji, A. Shimizu, "An impersonation attack on one-time password authentication protocol OSPA," IEICE Trans. on Communications, 86 (7), 2182-2185, 2003.
6. C.W. Lin, J.J. Shen, M.S. Hwang, "Security enhancement for optimal strong-password authentication protocol," ACM SIGOPS Operating Systems Review, 37 (3), 12-16, 2003.
7. T.H. Chen, W.B. Lee, G. Horng, "Secure SAS-like password authentication schemes", Computer Standards & Interfaces, 27,25-31, 2004.
8. E.J. Yoon, E.K. Ryu, K.Y. Yoo, "Fixing problems in Lin et al.'s OSPA protocol," Applied Mathematics and Computation, 166, 46-57, 2005.
9. J.H. Lee, I.J. Jang, H.S. Yoo, "Modified token-update scheme for site authentication", LNCS 3481, 111-116, 2005

저 자 소 개



준량 (E-mail: gsbeyond@gmail.com)
Chengdu university of technology in China, Department of Computer science and technology, MS
관심분야 Applied Cryptography

장인주 (E-mail: jangij@dreamwiz.com)
인하대학교 컴퓨터정보공학 (석사)
관심분야 Applied Cryptography



유형선 (E-mail: hsyoo@inha.ac.kr)
Ghent University, Belgium 기계공학 (박사)
현재 인하대학교 컴퓨터 공학부 교수
관심분야 Applied Cryptography, Scientific Computation