스마트 카드를 이용한 안전한 S/KEY 일회용 패스워드 인증 스킴

윤은준[O] 류은경 유기영
경북대학교 컴퓨터공학과 정보보호연구실
{ejyoon[O], ekryu}@infosec.knu.ac.kr yook@knu.ac.kr

A Secure S/KEY One-Time Password Authentication Scheme Using Smart Cards

Eunjun Yoon[0], Eunkyung Ryu, Keeyoung Yoo
Dept. of Computer Engineering Graduate School, Kyungpook National University Daegu, Korea

요 약

본 논문에서는 RFC 1760 표준 S/KEY 일회용 패스워드 인증 스킴이 서버 스푸핑 공격, 재시도 공격, 오프라인 패스워드 공격, 능동적 공격 등 여러 가지 공격에 취약함에 대해서 설명하고 이들 공격에 안전한 스마트 카드를 이용한 새로운 스킴을 제안한다. 본 논문에서 제안한 스킴은 스마트 카드를 이용한 안전한 S/KEY 일회용 패스워드 인증 스킴으로 기존의 S/KEY 일회용 패스워드 인증 스킴의 보안 문제점을 개선하였으며 상호 인증을 가능하게 하는 새로운 스킴으로 여러 가지 안전성 향상을 보인다.

1. 서 론

인터넷은 오늘날 없어서는 안되는 중요한 통신 수단이다. 하지만 현재 인터넷 환경은 사용자 위조, 인증 정보 엿보기 및 크랙, 통신 메시지 가로채기 등의 위협으로부터 안전하지 못하다.

패스워드 인증 스킴은 가장 일반적으로 사용되는 인증 방법이다. 하지만 전통적인 패스워드 인증 스킴은 사전 공격, 서버 스푸핑 공격, 재전송 공격 등에 안전하지 못하다. 이러한 취약점을 해결한 안전성이 향상된 여러 패스워드 기반의 인증 스킴들이 지금까지 제안되었다[1]-[6]. 그러나 이러한 스킴들의 대부분은 많은 연산량을 요구한다.

Lamport [1]가 재전송 공격에 안전한 일회용 패스워드 개념을 제안한 이후 Haller는 Lamport의 스킴을 응용한 S/KEY 일회용 패스워드 스킴 [7][8]을 제안하여 RFC 1760 [9]에 표준화되었다. 하지만 S/KEY 스킴은 재전송 공격의 특별한 경우인 재시도 공격과 오프라인 사전 공격과 같은 몇몇 강화된 공격에 안전하지 못하다고 지적하였다[10][11][12].

본 논문에서는 RFC 1760 표준 S/KEY 일회용 패스워드 스킴의 보안 취약점을 강화한 스마트 카드를 이용한 새로운 S/KEY 일회용 패스워드 스킴을 제안하며 제안한 스킴이 서버 스푸핑 공격, 재시도 공격, 오프라인 패스워드 공격 등 여러 가지 공격에 안전하고 상호인증과 기밀성이 제공되는 스킴임을 분석한다.

본 논문의 구성은 다음과 같다. 2장에서 FRC 1760 표준 S/KEY 일회용 패스워드 인증 스킴에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 스마트 카드를 이용한 S/KEY 일회용 패스워드 인증 스킴의 내용을 기술하고 안전성을 분석한다. 마지막으로 4장에서 결론을 맺는다.

2. 관련 연구

이 장에서는 본 논문에서 사용할 용어들을 정의하고, RFC 1760 표준 S/KEY 일회용 패스워드 인증 스킴을 소개하고 스킴이 가지는 보안문제를 분석한다.

2.1 용어 정의

- N : 전체 로그인 횟수
- t : t번째 로그인
- C : N-t로 계산되며, 남은 로그인 횟수
- SEED : 사용자 및 서버에서 패스워드 생성시 사용하는 아주 큰 랜덤수인 유일한 seed 값
- H(·) : 일방향 해쉬 함수(one-way hash function)
- ⊕ : 배타적 논리합 연산(Exclusive OR operation)
- x : 서버의 비밀키
- K : 사용자 비밀 패스워드
- p : 일회용 패스워드(one-time password)

2.2 S/KEY 일회용 패스워드 스킴

S/KEY 스킴에서는 사용자가 자신의 비밀키로 일회용 패스워드를 생성하기 위해서 일방향 해쉬 함수를 이용한다. S/KEY 일회용 패스워드 스킴은 등록 단계, 로그인 단계, 인증 단계로 구성되며 스킴의 수행 절차는 다음과 같다.

등록 단계:

User←Server: N, SEED
User→Server: p_0

(1) 서버는 일회용 패스워드의 일련번호 N을 선택하고, 사용자를 위한 서버 특유의 seed 값인 SEED를 생성하여 사용자에게 N과 SEED를 전송한다.

(2) 사용자는 자신의 비밀키 K를 계산하여 pass-phrase로 입력하여 SEED와 K를 XOR 연산하여 결합하고 N번의 일방향 해쉬 함수를 적용하여 초기 패스워드 $p_0=H^N(K\oplus SEED)$를 얻는다. 예를 들면, 만약 N이 100이면 사용자는 서버에 100번의 로그인을 할 수 있으며 초기 패스워드 $p_0=H^{100}(K\oplus SEED)$가 된다. 100번의 로그인 후 사용자는 서버에 재등록을 해야 한다.

로그인 단계:

User←Server: C=(N-t), SEED
User→Server: $p_t=H^C(K\oplus SEED)$

(1) t번째 로그인을 위해 서버는 사용자에게 SEED와 C=(N-t)를 챌린저(challenge)로 사용자에게 보낸다.
(2) 사용자는 수신한 C와 SEED를 이용하여 t번째 일회용 패스워드 $p_t=H^C(K\oplus SEED)$를 계산한 후 서버에 응답(response)한다.

인증 단계:

서버는 수신한 일회용 패스워드 p_t(예를 들면, 만약 t=2이면 $p_t=p_2=H^{98}(K\oplus SEED)$)에 한번 해쉬 연산을 적용하여 해쉬한 값과 데이터베이스에 저장되어 있는 t-1번째 일회용 패스워드 p_{t-1}과 비교한다. 위의 예로, $H(p_t)=H(p_2)=H(H^{98}(K\oplus SEED))=H^{99}(K\oplus SEED)=p_{t-1}=p_1$이 된다. 만약 두 값이 같으면, 서버는 사용자를 인증하여 데이터베이스에 저장되어 있는 t-1번째 일회용 패스워드 p_{t-1}을 t번째 일회용 패스워드 p_t로 업데이트하고 저장되어 있는 일련 번호를 C로 업데이트한다.

2.3 S/KEY 일회용 패스워드 스킴의 보안문제

RFC 1760 표준 S/KEY 일회용 패스워드 스킴의 기술 문서에서는 C와 SEED의 기밀성 유지 필요성에 관한 설명을 하지 않았다. C는 매 로그인마다 한번씩 감소하며, SEED는 N번의 로그인 동안 동일값을 유지한다. 그리고 이들 두 값은 평문으로 전송되어 공격자에 의해 쉽게 예측되어 질 수 있다.

추가적으로 사용자는 서버를 인증할 수 없다. 이러한 보안 헛점으로 인해 공격자는 다음번 챌린저를 미리 예측할 수 있으며 사용자에게 미리 예측한 챌린저를 전송하여 서버로 위장할 수 있다. 이후 공격자는 사용자의 일회용 패스워드를 얻은 후에 서버의 인증과정을 통과하기 위해 사용자로 위장할 수도 있다. 명백히 S/KEY 스킴은 재시도 공격을 방어할 수 없다.

오프라인 사전공격 또한 S/KEY 스킴에서 중요한 보안문제이다. 온라인 사전공격은 로그인 시도횟수를 제한함으로써 쉽게 방지할 수 있다. 공격자는 인터넷상에서 평문으로 전송되는 C와 SEED 그리고 t번째 일회용 패스워드 $p_t=H^C(K\oplus SEED)$를 가로채기하여 오프라인 사전공격으로 사용자의 비밀키 K를 얻을 수 있다.

3. 제안한 S/KEY 일회용 패스워드 스킴

이 장에서는 2.3절에서 언급한 RFC 1760 표준 S/KEY 일회용 패스워드 인증 스킴의 보안문제를 해결할 수 있는 스마트 카드를 이용한 새로운 스킴을 제안한다. 또한 제안한 스킴의 여러 가지 공격에 대한 안전성을 분석한다.

3.1 제안한 인증 스킴

본 논문에서 제안한 스마트 카드를 이용한 일회용 패스워드 인증 스킴은 등록 단계, 로그인 단계, 인증 단계로 구성되며 스킴의 수행 절차는 다음과 같다.

등록 단계:

본 논문에서 제안한 스킴의 등록 단계는 그림 1과 같다.

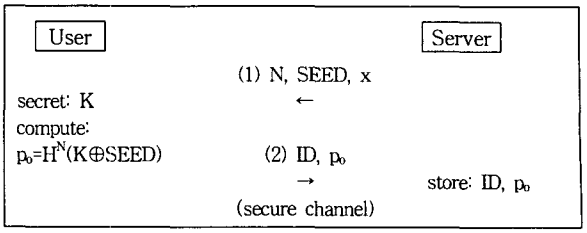

그림 1 제안한 스킴의 등록 단계

(1) 서버는 N, SEED, x를 챌린저(challenge)로 생성하여 안전한 채널을 통하여 사용자의 스마트 카드에 입력한다.
(2) 사용자는 스마트 카드에 저장된 N, SEED와 함께 자신의 임의의 길이의 비밀 패스워드 K를 이용하여 초기 패스워드 $p_0=H^N(K\oplus SEED)$를 계산한 후 ID와 함께 안전한 채널을 통하여 서버에 응답(response)한다. 서버는 사용자의 ID와 초기 패스워드 p_0를 데이터베이스에 저장한다.

로그인 단계:

본 논문에서 제안한 스킴의 로그인 단계는 그림 2와 같다.

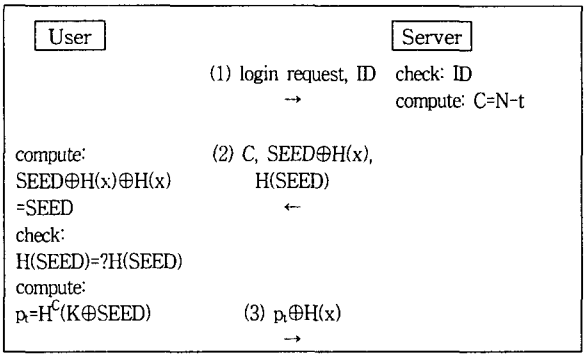

그림 2 제안한 스킴의 로그인 단계

(1) t번째 로그인을 위해 사용자는 서버에게 로그인 요청을 한다.
(2) 서버는 ID의 유효성을 검사한 후, C=(N-t)를 계산한다. SEED를 생성하여 서버의 비밀키 x를 한번 해쉬한 값 H(x)와 함께 XOR 연산한 SEED⊕H(x)와 H(SEED)을 계산하여 C와 함께 사용자에게 챌린저로 보낸다.
(3) 사용자는 자신의 스마트 카드에 저장된 서버의 비밀키 x를 한번 해쉬한 값 H(x)로 수신한 SEED⊕H(x)를 XOR 연산하여 SEED를 추출한다. 추출한 SEED를 한번 해쉬하여 수신한 H(SEED)와 같은지를 비교한다. 만약 두 값이 같으면, 사용자는 서버를 인증하여 t번째 일회용 패스워드 $p_t=H(p_{t-1})$를 계산한 후 H(x)와 XOR하여 서버에 응답한다.

인증 단계:

본 논문에서 제안한 스킴의 인증 단계는 그림 3과 같다.

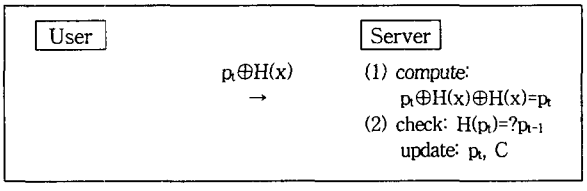

그림 3 제안한 스킴의 인증 단계

(1) 서버는 수신한 $p_t \oplus H(x)$를 자신의 비밀키 x를 한번 해쉬한 값 H(x)로 XOR 연산하여 t번째 일회용 패스워드 p_t을 추출한다.
(2) 서버는 추출한 t번째 일회용 패스워드 p_t을 한번 해쉬한 값 $H(p_t)$와 서버에 저장되어 있는 t-1번째 일회용 패스워드 p_{t-1}을 비교한다. 만약 두 값이 같으면, 서버는 사용자를 인증하여 데이터베이스에 저장되어 있는 t-1번째 일회용 패스워드 p_{t-1}을 p_t로 업데이트하고 저장되어 있는 일련 번호를 C로 업데이트한다.

3.2 안전성 분석

이 절에서는 제안한 스킴의 안전성을 분석한다.
(1) 서버 스푸핑 공격:
로그인 단계에서 사용자는 SEED, H(x)를 통하여 서버를 인증하며, 서버는 p_t에 의해서 사용자를 인증한다. 이러한 상호 인증은 공격자의 서버 스푸핑 공격을 방지할 수 있다.

(2) 재시도 공격:
서버의 챌린저가 서버의 비밀키 H(x)에 의해 보호되어 미리 예측될 수 없기 때문에 공격자는 서버로 위장할 수 없으며 일회용 패스워드를 사용함으로 사용자를 속일 수 있는 다음 챌린저를 미리 예측할 수 없다. 따라서 공격자가 이전의 챌린저 메시지를 가지고 있어도 다음 챌린저에서 그 메시지를 사용할 수 없으므로 재시도 공격을 수행할 수 없다.

(3) 오프라인 사전공격:
본 논문에서는 서버에서 아주 큰 랜덤 수의 SEED를 생성함으로 인해서 사용자의 비밀키 K를 보호하는데 사용하였다. 그러므로 공격자가 오프라인 사전 공격을 시도하려면 K와 SEED를 동시에 결정하여야 함으로 공격이 어려우며 사용자 비밀키 K는 서버로 인해 안전하게 보호되어 오프라인 사전공격에 안전하다.

(4) 능동적 공격:
능동적 공격에 안전하고 메시지 내용이 누설되지 않는 비밀성을 가지기 위해서는 세션 암호를 사용하는 것이 필요하다. 제안한 스킴에서는 통신 당사자간에 주고받는 메시지를 암호화하기 위해 인증 과정에서 서버가 자신의 비밀키 x에 일방향 해쉬 함수를 적용하여 세션키로 사용하기 때문에 능동적 공격에 안전하며 비밀성을 유지 할 수 있다.

표1은 RFC 1760 표준 S/KEY 일회용 패스워드 스킴과 제안한 스킴의 안전성 등을 비교 분석한 결과이다.

표 1 제안한 스킴과의 비교 분석

	RFC 1760 표준 S/KEY	제안한 스킴
사용횟수	N회	N회
서버 스푸핑 공격 방지	O	O
재시도 공격 방지	X	O
오프라인 사전공격 방지	X	O
능동적 공격 방지	X	O
메시지 엿보기 방지	X	O
안전성 기반	일방향 해쉬 함수	일방향 해쉬 함수
상호인증 기능	X	O
세션키 생성	X	O

4. 결 론

본 논문에서 제안한 스킴은 S/KEY 일회용 패스워드 인증 스킴과 스마트 카드를 이용하여 안전하고 효율적인 상호 인증을 가능하게 하는 새로운 스킴이다. 스마트 카드와 일방향 해쉬 함수에 의존한 여러 가지 안전성은 기존의 S/KEY 방식의 문제점을 완전하게 개선하고 있으며 간편한 구조로 이루어져 있다.
또한 제안한 스킴은 상호인증과 기밀성이 제공되는 스킴으로서, 상호 인증을 위해 SEED와 서버의 비밀키 x를 이용하여 서버를 인증하는데 사용하였으며 기밀성을 제공하기 위해 인증 처리동안 세션키를 생성 및 유지하였다.
RFC 1760 표준 S/KEY와 달리 본 논문에서 제안한 스킴은 사용자가 스마트 카드를 사용하여 임의의 로컬 컴퓨터에서 서버로 쉽게 로그인 할 수 있도록 사용자 로그인 처리를 간편하게 하였기 때문에 이식성이 좋다.
따라서 본 논문에서 제안한 새로운 스킴이 가지는 이러한 장점들은 보안이 요구되는 온라인 뱅킹, 온라인 계약, 온라인 협약과 같은 민감한 통신 환경에 아주 이상적으로 사용 될 수 있다.

참고문헌

[1] L. Lamport, "Password authentication with insecure communication," Communications of ACM, Vol. 24, pp.770-772, 1981.
[2] T.Kwon, "Authentication and key agreement via memorable password," Proc. 2001 Internet Society Network and Distributed System Security Symposium, San Diego, CA, Feb. 2001.
[3] T.Wu, "Secure remote password protocol," Proc. 1998 Internet Society Network and Distributed System Security Symposium, San Diego, CA, March 1998.
[4] S.M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," Proc. IEEE Symposium on Research in Security and Privacy, pp.72-84, Oakland, May 1992.
[5] S.M. Bellovin and M. Merritt, "Augmented encrypted key exchange: A Password-based protocols secure against dictionary attacks and password file compromise," AT&T Bell Laboratories, 1994.
[6] L. Gong, M.Lomas, R. Needham, and J, Saltzer, "Protecting poorly chosen secrets from guessing attacks," IEEE J.Sel. Areas Commun., vol.11,no.5, pp.648-656, June 1993.
[7] N.M. Haller, "A one-time password system," RFC 1938, May 1996.
[8] N.M. Haller, "On internet authentication," RFC 1704, Oct. 1994.
[9] N.M. Haller, "The S/KEY one-time password system," RFC 1760, Feb. 1995.
[10] C.J. Mitchell and L. Chen, "Comments on the S/KEY user authentication scheme," ACM Operation Systems Review, vol.30, no.4, pp.12-16, Oct. 1996.
[11] S.M. Yen and K.H. Liao, "Shared authentication token secure against replay and weak key attacks," Information Processing Letters, vol.62, pp.77-80, 1997.
[12] T.C.YEH, H.Y.SHEN and J.J.HWANG. "A Secure One-Time Password Authentication Scheme Using Smart Cards," IEICE Trans. Commun., vol.e85-b, no.11 November 2002.