韓國컴퓨터情報學會 論文誌
第15卷 第12號, 2010. 12.

2010-15-12-3-7

효율적인 스마트카드 사용자 인증 프로토콜

용 승 림*, 조 태 남**

An Efficient User Authentication Protocol using Smartcard

Seunglim Yong *, Taenam Cho **

요 약

개인 프라이버시 보호에 대한 관심이 증가하면서, 원격 시스템에서 사용자 익명성을 제공하는 스마트카드 기반의 인증 프로토콜에 대한 연구가 활발하게 진행되고 있다. 최근에는 사용자 익명성을 제공하는 기법과 더불어 악의적인 사용자를 추적할 수 있는 연구들이 제안되고 있다. 2008년 Kim은 사용자의 익명성을 보장하면서 악의적인 사용자를 감지하여 추적 가능한 인증 프로토콜을 제안하였고, 2009년 Choi는 Kim의 프로토콜의 익명성 문제를 제기하고 이를 개선한 새로운 프로토콜을 제안하였다. 그러나 두 프로토콜은 익명성을 제공하지 못하거나, 심각한 계산상의 오류를 가지고 있다. 본 논문에서는 두 프로토콜의 문제점을 분석하고 문제점을 해결한 개선된 프로토콜을 제안한다.

Abstract

Due to the increasing interest and demands of user privacy, remote user authentication schemes using smart card has been researched in active. Recently, a lot of suggestion have been made in order to provide user's anonymity and trace a malicious user. In 2008, Kim et al. proposed a traceable anonymity authentication scheme. In 2009, Choi et al. pointed out that Kim's protocol was insecure against outsider attacker and proposed an improved scheme. But Kim's and Choi's schemes fail to provide the user's anonymity or compute some values in the protocol. In this paper, we analyse those problems and suggest two improved schemes to resolve those problems.

▸ Keyword : 원격 사용자 인증(Remote User Authentication), 스마트카드(SmartCard), 익명성(Anonymity), 추적성(Traceability)

• 제1저자 : 용승림 교신저자 : 조태남
• 투고일 : 2010. 08. 18, 심사일 : 2010. 09. 27, 게재확정일 : 2010. 10. 13.
* 인하공업전문대학 컴퓨터시스템과 조교수 ** 우석대학교 정보보안학과 조교수
※ 이 논문은 2009학년도 인하공업전문대학 교내연구비지원에 의하여 연구되었음.

Ⅰ. 서 론

최근, 컴퓨터 및 네트워크의 발달로 사용자는 언제 어디서나 다양한 인터넷 서비스를 제공받고자 한다. 또한 분산컴퓨팅 환경에서 원격으로 작업을 수행하는 일이 빈번해지면서 인증에 대한 많은 연구가 진행되고 있다. 그 중 스마트카드를 이용한 원격 사용자 인증은 스마트카드가 지닌 이동성과 기능적 안전성으로 인하여 특히 주목받고 있다.

사용자 인증 프로토콜이란 서비스를 제공하는 서버와 서비스를 이용하려는 사용자간에 서로 상대의 신원을 확인하고 정당한 사용자와 서버라는 것을 상호 검증할 수 있는 프로토콜이다. 초기의 인증 프로토콜들에서 서버는 사용자 인증 요청에 대한 검증을 위해 검증 테이블을 저장하고 있어야 했다[4]. 하지만 서버에 대한 안전성과 신뢰가 요구되고 사용자의 아이디와 패스워드 관리 비용 부담 등의 문제가 제기되면서 검증테이블을 이용하지 않는 인증 기법들이 연구되고 있다. 특히 유비쿼터스 환경 하에서는 개인 정보보호와 프라이버시에 많은 관심이 증가되면서 스마트카드를 이용한 익명성을 제공하는 원격 인증 시스템에 대한 연구들이 수행되고 있다.

2004년 Das 등은 동적 아이디를 사용하여 사용자와 원격서버를 제외한 제 3자에 대해 사용자 익명성을 제공하려는 기법을 최초로 제안하였다[3]. 그러나 2005년 Chien과 Chen은 Das 등의 프로토콜이 로그인 단계에서 서버에게 보내는 데이터를 통해 사용자를 구분할 수 있고 그로 인해 사용자 익명성을 제공하지 못함을 밝혀내었다[1]. Chien의 프로토콜은 문제 해결을 위한 새로운 익명성 제공 프로토콜을 제안하였지만 제 3자에게만 안전한 사용자 익명성을 제공한다는 제약점을 가지고 있다. 2006년 Chai는 제 3자 뿐 아니라 원격서버에 대해서도 익명성을 보장하는 프로토콜을 처음 제안하였다[2].

익명성을 보장하는 인증 프로토콜의 연구가 진행되면서 문제 발생 시 악의적인 사용자를 추적할 수 있는 기능의 필요성이 대두되었다. 2008년 Kim 등은 악의적인 사용자에 대해 추적할 수 있는 프로토콜을 제안하였다. 2009년 Choi[6]는 Kim[5]의 논문이 제 3자에 대한 사용자 익명성을 보장하지 못한다는 문제점을 찾아내고 새로운 추적 가능하고 익명성을 제공하는 프로토콜을 제안하였다. Choi의 제안 기법은 Kim의 사용자의 익명성을 개선하였지만, 로그인 단계에서 스마트카드의 계산 과정에 오류가 있다.

본 논문에서는 2009년 제안된 Choi 등의 논문에서의 계산 오류 문제를 밝혀내고 이를 해결하는 프로토콜을 제안한다. 제안하는 프로토콜은 Choi의 논문과 Kim의 논문을 개선함으로써 문제를 해결하였으며, 제 3자와 서버에 대해서 사용자 익명성을 만족하면서 악의적인 사용자 발견 시 추적이 가능하도록 설계되었다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구로 Kim과 Choi의 프로토콜을 살펴보고, 문제점을 분석한 후 3장에서 Kim의 프로토콜을 개선하여 문제점을 해결하고 제안한 프로토콜을 분석한다. 4장에서는 Choi의 프로토콜을 개선하여 문제점을 해결하고 제안한 프로토콜의 안전성과 효율성을 분석하고 5장에서 결론 및 향후 연구에 대해 기술한다.

본 논문에서 제안하는 프로토콜과 관련된 기존 연구의 프로토콜 기술에 이용되는 용어를 정의하고, 익명성을 제공하면서 추적이 가능한 기존의 사용자 인증 프로토콜의 문제점에 대하여 분석한다.

1. 용어 정의

본 논문에서 제안한 프로토콜 및 관련 연구에 사용될 용어를 [표 1]과 같이 정의한다.

표 1. 용어 정의
Table 1. Notation

기호	설명
U_i	i번째 사용자
ID_i	U_i의 아이디
PW_i	U_i의 패스워드
UIN_i	U_i의 개인 정보
$h()$	일방향 해쉬함수
$\oplus$	XOR 비트 연산자
x, y	서버의 비밀값
PTR	신뢰 기관의 공개키
TR	악의적인 사용자 추적을 위한 사용자 추적값
$E_k(m)$	키 k로 메시지 m을 암호화하는 공개키 암호 알고리즘
CS	신뢰기관에 제출하는 악의적인 사용자에 대한 고발장 (Complain Sheet)

2. Kim의 프로토콜

Kim 등은 서버와 제 3자에 대하여 익명성을 제공하면서 악의적인 사용자에 대해 서버가 사용자를 추적할 수 있는 프로토콜을 제안하였다.

2.1 Kim의 프로토콜

<등록 단계>

Step1. 사용자 U_i는 자신의 ID_i와 $h(PW_i)$를 안전한 채널을 이용하여 서버에게 전달한다.

Step2. 서버는 다음을 계산하고 스마트카드에{I, I_c, R_i,$h()$, TR, p, y,ATR}을 저장하여 발급한다.

(1) $R_i \leftarrow h(x) \oplus h(PW_i) \oplus h(ID_i \oplus x)$

(2) $I \leftarrow h(ID_i \oplus x)$

(3) $I_c \leftarrow h(ID_i \oplus x) \oplus h(ID_i) \oplus h(PW_i)$

(4) $TR \leftarrow E_{PTR}(ID_i \| UIN_i)$

(5) $ATR \leftarrow h(TR \oplus x)$

<로그인 단계>

Step1. 사용자는 스마트카드를 리더기에 삽입하고 ID_i와 PW_i를 입력한다.

Step2. 스마트카드는 랜덤값 a, e, r을 생성하고 타임스탬프 T를 생성하여 다음과 같이 수행한다.

(1) $I \oplus h(ID_i) \oplus h(PW_i) \stackrel{?}{=} I_c$ ····· 1)

(2) $X \leftarrow g^a \bmod p$

(3) $N_1 \leftarrow R_i \oplus I \oplus h(PW_i) \oplus h(PW_i \oplus r)$
$(= h(x) \oplus h(PW_i \oplus r))$

(4) $UTR \leftarrow h(PW_i \oplus r) \oplus TR$

(5) $OAR \leftarrow h(PW_i \oplus r) \oplus ATR$

(6) $M \leftarrow h(e \oplus T)$

(7) $V \leftarrow h(M \oplus y) \oplus h(PW_i \oplus r)$

(8) $C_{ID_i} \leftarrow h(N_1 \oplus h(y \oplus T), X, OAR)$

Step3. 인증을 위해 메시지 $\{T, C_{ID_i}, M, V, X, UTR\}$를 서버에게 보낸다.

<인증 단계>

Step1. 서버는 메시지 $\{T, C_{ID_i}, M, V, X, UTR\}$를 T' 시간에 받는다.

Step2. 서버는 T시간과 T' 시간 사이의 시간 간격(time interval)이 수용할 만큼 짧은지 확인하고, 다음 단계를 통하여 C_{ID_i}를 검증한다.

(1) $N_2 \leftarrow V \oplus h(M \oplus y) \; (= h(PW_i \oplus r))$

(2) $TR' \leftarrow N_2 \oplus UTR$

(3) $N_1' \leftarrow N_2 \oplus h(x)$

(4) $OAR \stackrel{?}{=} h(TR' \oplus x) \oplus N_2$ ····· 2)

(5) $C_{ID_i} \stackrel{?}{=} h(N_1' \oplus h(y \oplus T), X, OAR)$ ····· 3)

Step3. Step2에서 C_{ID_i}의 검증이 성공하면$SK \leftarrow X^b \bmod p$, $Y \leftarrow g^b \bmod p$, $M_s \leftarrow h(OAR \oplus M \oplus y, SK, Y)$를 계산하여 M_s과 Y를 사용자에게 보낸다.

Step4. 사용자는 다음과 같이 M_s를 확인한다.

(1) $SK \leftarrow Y^a \bmod p$

(2) $M_s \stackrel{?}{=} h(OAR \oplus M \oplus y, SK, Y)$ ····· 4)

Step5. Step 4에서 M_s 검증이 성공하면, 사용자와 서버는 SK를 이용하여 $SK_{us} \leftarrow h(SK)$을 계산하고 SK_{us}를 세션키로서 공유한다.

<추적 단계>

Step1. 서버는 TR값과 CS를 함께 신뢰기관에 제출한다.

Step2. 신뢰기관은 CS를 확인하고, TR값을 자신의 개인키로 복호화하여 서버에게 사용자 정보를 알려준다.

<패스워드 변경 단계>

U_i가 자신의 PW_i를 새로운 PW_i^*로 바꾸고자 할 때, U_i는 서버와 상관없이 스마트카드만을 이용하여 새로운 PW_i^*로 교체한다. 스마트카드는 다음과 같이 수행한다.

Step1. U_i가 자신의 스마트카드를 리더기에 삽입한 후, 자신의 ID_i와 PW_i를 입력하면, 스마트카드는 다음을 확인한다.

$I \oplus h(ID_i) \oplus h(PW_i) \stackrel{?}{=} I_c$

Step2. 만약 값이 일치하면 스마트카드는 다음과 같이 수행하여 PW_i 검증 정보를 교체한다.

$I_c^* \leftarrow I_c \oplus h(PW_i) \oplus h(PW_i^*)$

$R_i^* \leftarrow R_i \oplus h(PW_i) \oplus h(PW_i^*)$

2.2. Kim의 프로토콜 분석

Kim 등은 사용자 익명성과 추적성을 제공하는 프로토콜을 제안하였다. 그러나 Choi는 공격자가 사용자와 동일한 서버에 등록된 정당한 사용자일 때 사용자의 익명성이 파기될 수 있음을 보였다. 공격자는 인증 단계에서 사용자가 서버에게 보내는 $\{T, C_{ID_i}, M, V, X, UTR\}$을 가로채고, 자신의 스마트카드에 저장되어 있는 y값을 이용하여 다음과 같이 사용자 추적 정보 TR값을 알아낼 수 있다.

(1) $N_3 \leftarrow V \oplus h(M \oplus y) \; (= h(PW_i \oplus r))$

(2) $N_3 \oplus UTR \; (= TR)$

제 3자는 인증 단계에서 서버에게 보내는 데이터를 가로채어 TR값을 계산해낼 수 있게 되고, 이 값을 신뢰기관에 제공하여 사용자를 알 수 있게 된다. 따라서 Kim의 프로토콜은 제 3자에 대한 사용자 익명성을 제공하지 못한다.

3. Choi의 프로토콜

Choi 등이 제안한 프로토콜은 Kim의 프로토콜의 문제점을 해결하기 위하여 $h(PW_i \oplus r)$ 값을 $h(h(ID_i) \oplus x)$로 변경하여 프로토콜을 제안하였다. Choi의 프로토콜은 등록, 로그인, 인증, 추적, 패스워드 변경단계로 구성되어 있으며, 추적단계와 패스워드 변경단계는 Kim의 프로토콜과 동일하다.

3.1 Choi의 프로토콜

<등록 단계>

Step1. 사용자 U_i는 자신의 ID_i와 $h(PW_i)$를 안전한 채널을 이용하여 서버에게 전달한다.

Step2. 서버는 사용자의 ID_i와 $h(PW_i)$를 이용하여 다음을 계산하고 $\{I, I_c, R_i, h(), TR, p, y, ATR\}$을 스마트카드에 저장한다.

(1) $R_i \leftarrow h(x) \oplus h(PW_i) \oplus h(h(ID_i) \oplus x)$

(2) $I \leftarrow h(ID_i \oplus x)$

(3) $I_c \leftarrow h(ID_i \oplus x) \oplus h(ID_i) \oplus h(PW_i)$

(4) $TR \leftarrow E_{PTR}(ID_i \| UIN_i)$

(5) $ATR \leftarrow h(TR \oplus x)$

<로그인 단계>

Step1. 사용자는 스마트카드를 리더기에 삽입하고 ID_i와 PW_i를 입력한다.

Step2. 스마트카드는 랜덤값 a, e, r을 생성하고 타임스탬프 T를 생성하여 다음과 같은 수행을 한다.

(1) $I \oplus h(ID_i) \oplus h(PW_i) \stackrel{?}{=} I_c$ ..... 5)

(2) $X \leftarrow g^a \bmod p$

(3) $N_1 \leftarrow R_i \oplus h(PW_i)$
$(= h(x) \oplus h(h(ID_i) \oplus x))$

(4) $UTR \leftarrow h(h(ID_i) \oplus x) \oplus TR$

(5) $OAR \leftarrow h(h(ID_i) \oplus x) \oplus ATR$

(6) $M \leftarrow h(e \oplus T)$

(7) $V \leftarrow h(M \oplus y) \oplus h(ID_i)$

(8) $C_{ID_i} \leftarrow h(N_1 \oplus h(y \oplus T), X, OAR)$

Step3. 인증을 위해 메시지 $\{T, C_{ID_i}, M, V, X, UTR\}$를 서버에게 보낸다.

<인증 단계>

Step1. 서버는 메시지 $\{T, C_{ID_i}, M, V, X, UTR\}$를 T' 시간에 받는다.

Step2. 서버는 T 시간과 T' 시간 사이의 시간 간격이 수용할 만큼 짧은지 확인하고, 다음 단계를 통하여 C_{ID_i}를 검증한다.

(1) $N_2 \leftarrow V \oplus h(M \oplus y) (= h(ID_i))$

(2) $N_1' \leftarrow h(x) \oplus h(N_2 \oplus x)$

(3) $TR' \leftarrow h(N_2 \oplus x) \oplus UTR$

(4) $OAR \stackrel{?}{=} h(TR' \oplus x) \oplus h(N_2 \oplus x)$ ..... 6)

(5) $C_{ID_i} \stackrel{?}{=} h(N_1' \oplus h(y \oplus T), X, OAR)$ ..... 7)

Step3. Step2에서 C_{ID_i}의 검증이 성공하면 $SK \leftarrow X^b \bmod p$, $Y \leftarrow g^b \bmod p$, $M_s \leftarrow h(OAR \oplus M \oplus y, SK, Y)$를 계산하여 M_s과 Y를 사용자에게 보낸다.

Step4. 사용자는 다음과 같이 M_s를 확인한다.

(1) $SK \leftarrow Y^a \bmod p$

(2) $M_s \stackrel{?}{=} h(OAR \oplus M \oplus y, SK, Y)$ ..... 8)

Step5. Step4에서 M_s 검증이 성공하면, 사용자와 서버는 SK를 이용하여 $SK_{us} \leftarrow h(SK)$을 계산하고 SK_{us}를 세션키로서 공유한다.

3.2 기존 프로토콜의 문제점 분석

Choi의 기법에서 사용자의 스마트카드는 로그인 단계에서 사용자가 입력한 아이디와 패스워드, 그리고 스마트카드에 저장된 값을 이용하여 $\{C_{ID_i}, M, V, X, UTR\}$값을 계산하고, 이를 서버에게 전송한다. Choi 등은 Kim 등의 프로토콜의 문제점을 지적하면서 로그인 단계의 Step2에서 스마트카드가 UTR의 값을 $h(h(ID_i) \oplus x) \oplus TR$을 계산하여 구한다고 하였다. 그러나 프로토콜에서 스마트카드가 알 수 있는 값은 Step1에서 계산된 $h(h(ID_i) \oplus x)$와 $h(x)$를 XOR 연산한 N_1 값뿐이다. 만약 Choi의 제안처럼 사용자가 UTR값을 계산하려면 사용자는 서버의 비밀값 x를 알아서 $N_1 \oplus h(x)$를 계산할 수 있어야 한다. 그러나 사용자는 서버의 비밀값 x를 알지 못하기 때문에 N_1로부터 $h(h(ID_i) \oplus x)$ 또는 $h(x)$값

을 알아낼 수 없다. 따라서 UTR, OAR 값을 생성하기 위해 필요한 $h(h(ID_i)\oplus x)$ 값은 스마트카드가 계산할 수 없다.

III. 개선된 Kim의 프로토콜

본 논문에서는 Kim의 프로토콜과 Choi 등의 프로토콜을 수정하여, 익명성을 제공하면서 기존의 제안된 논문들에서 분석된 문제점을 해결하는 개선된 기법에 대하여 제안한다. 먼저 본 장에서는 Kim의 프로토콜을 수정하여 발견된 문제점을 해결하는 방법을 제안하고, 제안한 방법의 안전성과 효율성을 분석한다[5].

1. 개선된 Kim의 프로토콜

개선된 프로토콜은 Kim의 프로토콜과 마찬가지로 등록단계, 로그인단계, 인증단계, 추적단계, 패스워드 변경단계로 나누어진다. 수정된 프로토콜에서는 등록단계에서 사용자의 추적 정보 TR 값을 변경함으로써 문제점을 해결한다. 또한 효율성 향상을 위하여 랜덤값 e와 M을 없앤다. Kim의 프로토콜을 수정한 개선된 프로토콜은 다음과 같으며, 패스워드 변경 단계는 Kim의 프로토콜과 동일하다.

<등록 단계>

Step1. 등록단계에서 새로운 사용자 U_i는 자신의 ID_i와 $h(PW_i)$를 안전한 채널을 이용하여 원격 서버에게 전달한다.

Step2. 서버는 사용자의 ID_i와 $h(PW_i)$를 이용하여 다음을 계산한다.

(1) $R_i \leftarrow h(x)\oplus h(PW_i)\oplus h(ID_i\oplus x)$

(2) $I \leftarrow h(ID_i\oplus x)$

(3) $I_c \leftarrow h(ID_i\oplus x)\oplus h(ID_i)\oplus h(PW_i)$

(4) $TR \leftarrow E_{PTR}(ID_i \| UIN_i)\oplus x$

(5) $ATR \leftarrow h(TR\oplus x)$

Step3. 서버는 스마트카드에 $\{I, I_c, R_i, h(), p, y, TR, ATR\}$를 저장하고 스마트카드를 사용자에게 발급한다.

<로그인 단계>

Step1. 사용자는 스마트카드를 리더기에 삽입하고 ID_i와 PW_i를 입력한다.

Step2. 스마트카드는 랜덤값 a, r 과 타임스탬프 T를 생성하고 다음과 같이 수행한다.

(1) $I\oplus h(ID_i)\oplus h(PW_i) \stackrel{?}{=} I_c$ ····· 9)

(2) $X \leftarrow g^a \bmod p$

(3) $N_1 \leftarrow R_i\oplus I\oplus h(PW_i)\oplus h(PW_i\oplus r)$
$(= h(x)\oplus h(PW_i\oplus r))$

(4) $UTR \leftarrow h(PW_i\oplus r)\oplus TR$

(5) $OAR \leftarrow h(PW_i\oplus r)\oplus ATR$

(6) $V \leftarrow h(X\oplus y)\oplus h(PW_i\oplus r)$

(7) $C_{ID_i} \leftarrow h(N_1\oplus h(y\oplus T), X, OAR)$

Step3. 인증을 위해 메시지 $\{T, C_{ID_i}, V, X, UTR\}$를 서버에게 보낸다.

<인증 단계>

Step1. 서버는 메시지 $\{T, C_{ID_i}, V, X, UTR\}$를 T' 시간에 받고 T 시간과 T' 시간 사이의 시간 간격이 수용할 만큼 짧은지 확인하고, 다음 단계를 통하여 C_{ID_i}를 검증한다.

(1) $N_2 \leftarrow V\oplus h(X\oplus y)\ (= h(PW_i\oplus r))$

(2) $N_1' \leftarrow N_2\oplus h(x)$

(3) $TR' \leftarrow N_2\oplus UTR$

(4) $OAR \stackrel{?}{=} h(TR'\oplus x)\oplus N_2$ ····· 10)

(5) $C_{ID_i} \stackrel{?}{=} h(N_1'\oplus h(y\oplus T), X, OAR)$ ····· 11)

Step2. Step1에서 C_{ID_i}의 검증이 성공하면 $SK \leftarrow X^b \bmod p$, $Y \leftarrow g^b \bmod p$, $M_s \leftarrow h(OAR\oplus X\oplus y, SK, Y)$를 계산하여 M_s와 Y를 사용자에게 보낸다.

Step3. 사용자는 다음과 같이 M_s를 확인한다.

(1) $SK \leftarrow Y^a \bmod p$

(2) $M_s \stackrel{?}{=} h(OAR\oplus X\oplus y, SK, Y)$ ····· 12)

Step4. Step3에서 M_s 검증이 성공하면, 사용자와 서버는 SK를 이용하여 $SK_{us} \leftarrow h(SK)$을 계산하고 SK_{us}를 세션키로서 공유한다.

<추적 단계>

Step1. 악의적인 사용자 발견 시 서버는 신뢰기관에게 $TR\oplus x\ (= E_{PTR}(ID_i \| UIN_i))$을 계산하여 CS와 함께 전달한다.

Step2. 신뢰기관은 CS를 확인하고, 사용자 정보를 확인하기 위하여 서버가 보낸 $TR \oplus x (= E_{PTR}(ID_i \| UIN_i))$을 자신의 개인키로 복호화하고 사용자 정보를 서버에게 알려준다.

2. 분석

2.1 안전성

1) 제 3자에 대한 익명성

공격자는 인증 단계에서 사용자가 서버에게 보내는 $\{T, C_{ID_i}, V, X, UTR\}$을 가로채고, 자신의 스마트카드에 저장되어 있는 y값을 이용하여 다음과 같이 사용자 추적 정보 TR값을 계산해낼 수 있다.

(1) $N_3 \leftarrow V \oplus h(X \oplus y)\ (= h(PW_i \oplus r))$

(2) $N_3 \oplus UTR\ (= TR)$

위의 식으로부터 공격자는 $h(PW_i \oplus r)$값을 계산해내어 TR 값을 알아낼 수 있다. 그러나 공격자는 사용자의 아이디를 알아내기 위하여 신뢰기관에 TR 값을 제공한다 하더라도 실제 사용자의 아이디는 알아낼 수 없다. 왜냐하면 계산된 식으로 얻을 수 있는 값은 $E_{PTR}(ID_i \| UIN) \oplus x$이다. 즉, 신뢰기관이 사용자를 추적하기 위해서는 서버의 비밀값 x를 알아야 하나, 공격자는 서버의 비밀값을 알지 못하기 때문에 신뢰기관에 그 값을 제공할 수 없다. 따라서 공격자가 비록 TR 값을 구할 수는 있다고 하더라도 공격자는 그 값으로부터 사용자의 신분을 추적해낼 수 없게 되기 때문에 제 3자로부터의 익명성이 제공된다고 할 수 있다.

제안한 기법은 Kim의 논문에서 언급한 바와 같이 사용자의 불추적 익명성(unlinkability)을 제공하지는 못한다. 서버는 사용자의 실제 아이디를 프로토콜을 통해서 알 수는 없지만 사용자가 로그인을 할 때마다 매번 같은 TR 값을 얻기 때문에 서버는 이전 세션과 같은 사용자임은 알 수 있다.

2) 내부자 공격

서버의 내부자 공격을 막기 위하여, 사용자는 등록단계에서 ID_i는 제공하지만 패스워드는 암호학적 해시함수를 적용하여 $h(PW_i)$만 제공하였기 때문에 PW_i는 노출되지 않는다.

3) 재전송 공격

타임스탬프를 이용하여 메시지의 유효성을 검사하고 있기 때문에 시간차를 두고 공격을 수행하는 재전송 공격으로부터 안전할 수 있다. 만약 공격자가 사용자의 메시지를 저장하고, 재전송할 경우 그 메시지는 인증단계에서 검증을 통과할 수 없다.

2.2 효율성

제안한 프로토콜에서는 Kim의 프로토콜에서 이용되는 랜덤 값의 수를 3개에서 2개로 줄임으로써 로그인 단계에서의 해쉬 연산의 수를 1회 감소시키고 통신량도 감소시켰다. Kim의 프로토콜에서는 사용자 측에서 랜덤 값 e를 생성하고 $M = h(e \oplus T)$를 생성하여 로그인 단계와 인증 단계에 이용하였다. 그러나 이 값은 사용자가 매 세션마다 생성하는 X 값으로 대체하여도 안전성에 문제가 없다. 따라서 제안한 프로토콜에서는 M값과 이를 생성하기 위한 랜덤값 e를 생성하지 않았고, 이로써 해쉬 연산의 수와 로그인 단계에서 스마트카드가 서버에게 보내야 하는 데이터의 양도 줄임으로써 통신량도 감소하게 되었다.

표 2 . 효율성 분석
Table 2. Analysis of efficiency

		제안프로토콜	Kim의 프로토콜
해시 연산	LP	5H,1P	6H,1P
	AP	4H	4H
	TP	5H	5H

H: 해시연산, P:지수연산, LP:로그인단계, AP:인증단계
TP:추적단계

IV. 개선된 Choi의 프로토콜

1. 개선된 Choi의 프로토콜

개선된 Choi의 프로토콜에서는 Choi 프로토콜의 로그인 단계에서 스마트카드가 생성하는 UTR, OAR 값을 변경함으로써 문제점을 해결한다. 또한 효율성 향상을 위하여 랜덤 값 e와 M값을 없앤다. 개선된 프로토콜은 Choi의 프로토콜과 마찬가지로 등록단계, 로그인단계, 인증단계, 추적단계, 패스워드 변경단계로 나누어진다. 추적단계와 패스워드 변경 단계는 Choi의 프로토콜과 동일하다.

<등록 단계>

Step1. 등록단계에서는 스마트카드를 사용하는 새로운 사용자 U_i가 자신의 ID_i와 $h(PW_i)$를 안전한 채널을 이용하여 서버에게 전달한다.

Step2. 서버는 사용자의 ID_i와 $h(PW_i)$를 이용하여 다음을 계산한다.

(1) $R_i \leftarrow h(x) \oplus h(PW_i) \oplus h(h(ID_i) \oplus x)$

(2) $K \leftarrow h(ID_i \oplus x)$

(3) $I_c \leftarrow h(ID_i \oplus x) \oplus h(ID_i) \oplus h(PW_i)$

(4) $TR \leftarrow E_{PTR}(ID_i \| UIN_i)$

(5) $ATR \leftarrow h(TR \oplus x)$

Step3. 서버는 스마트카드에 $\{I, I_c, R_i, h(), TR, p, y, ATR\}$를 저장하고 스마트카드를 사용자에게 발급한다.

<로그인 단계>

Step1. 사용자는 스마트카드를 리더기에 삽입하고 ID_i와 PW_i를 입력한다.

Step2. 스마트카드가 다음과 같은 수행을 한 후 서버에게 메시지 $\{T, C_{ID_i}, V, X, UTR\}$을 보낸다.

(1) $I \oplus h(ID_i) \oplus h(PW_i) \stackrel{?}{=} I_c$ ····· 13)

(2) $X \leftarrow g^a \bmod p$

(3) $N_1 \leftarrow R_i \oplus h(PW_i)$
$(= h(x) \oplus h(h(ID_i) \oplus x))$

(4) $UTR \leftarrow N_1 \oplus TR$

(5) $OAR \leftarrow N_1 \oplus ATR$

(6) $V \leftarrow h(X \oplus y) \oplus h(ID_i)$

(7) $C_{ID_i} \leftarrow h(N_1 \oplus h(y \oplus T), X, OAR)$

<인증 단계>

Step1. 서버는 메시지 $\{T, C_{ID_i}, V, X, UTR\}$를 T' 시간에 받아 T 시간과 T' 시간 사이의 시간 간격이 수용할 만큼 짧은지 확인하고, 다음 단계를 통하여 C_{ID_i}를 검증한다.

(1) $N_2 \leftarrow V \oplus h(X \oplus y) (= h(ID_i))$

(2) $N_1' \leftarrow h(x) \oplus h(N_2 \oplus x)$

(3) $TR' \leftarrow N_1' \oplus UTR$

(4) $OAR \stackrel{?}{=} N_1' \oplus h(TR' \oplus x)$ ····· 14)

(5) $C_{ID_i} \stackrel{?}{=} h(N_1' \oplus h(y \oplus T), X, OAR)$ ····· 15)

Step2. Step1에서 C_{ID_i}의 검증이 성공하면 서버는 $SK \leftarrow X^b \bmod p$, $M_s \leftarrow h(OAR \oplus X \oplus y, SK, Y)$, $Y \leftarrow g^b \bmod p$를 계산하고, M_s와 Y를 사용자에게 보낸다.

Step3. 사용자는 다음과 같이 M_s를 확인한다.

(1) $SK \leftarrow Y^a \bmod p$

(2) $M_s \stackrel{?}{=} h(OAR \oplus X \oplus y, SK, Y)$ ····· 16)

Step4. Step 3에서 M_s 검증이 성공하면, 사용자와 서버는 SK를 이용하여 $SK_{us} \leftarrow h(SK)$을 계산하고 SK_{us}를 세션키로서 공유한다.

2. 분석

이 절에서는 제안한 프로토콜에 대한 효율성과 안전성에 대하여 분석한다.

2.1 안전성

1) 위장공격

공격자는 로그인 단계에서 사용자가 서버에게 전송하는 메시지 $\{T, C_{ID_i}, V, X, UTR\}$를 가로챌 수 있다. 만약 공격자가 사용자 서버에 등록된 정당한 사용자라면 자신의 스마트카드의 y값을 이용하여 $V \oplus h(X \oplus y) (= h(ID_i))$을 계산할 수는 있다. 하지만 서버의 비밀값 x를 모르기 때문에 $h(h(ID_i) \oplus x)$값과 $h(x)$값을 만들어 낼 수 없으므로 정당한 UTR과 C_{ID_i}를 조작할 수 없어 사용자로 위장할 수 없다.

2) 사용자 익명성

제안한 프로토콜은 제 3자에 대한 사용자 익명성을 제공한다. 공격자가 사용자를 추적하려면 UTR로부터 사용자 추적 정보 TR값을 알아내어야 한다. 공격자는 앞서 기술한 바와 같이 전송 데이터로부터 $V \oplus h(X \oplus y) (= h(ID_i))$을 계산해 낼 수는 있다. 그러나 $h(h(ID_i) \oplus x)$와 $h(x)$값을 계산해 낼 수 없기 때문에 UTR로부터 TR값도 알아낼 수 없다.

제안한 프로토콜은 서버에 대한 사용자 익명성도 제공한다. 서버는 자신이 가지고 있는 서버의 비밀값과 사용자로부터 받은 메시지 $\{T, C_{ID_i}, V, X, UTR\}$를 이용하여 아이디를 확인함으로써 정당한 사용자임을 인증할 수 있다. 이때, 서버는 정당한 사용자임은 확인할 수 있지만 사용자의 아이디는 알 수 없기 때문에 서버에 대한 사용자 익명성이 제공된다. 그러나 제안한 기법 역시 Kim의 논문에서 언급한 바와 같이 사용자의 불추적 익명성을 제공하지는 못한다. 서버는 사용자의 실제 아이디를 프로토콜을 통해서 알 수는 없지만 사용자가 로그인을 할 때마다 매번 같은 $h(ID_i)$값과 TR 값을 얻기 때문에 서버는 사용자가 누구인지는 알 수 없으나 이전 세션과 같은 사용자임은 알 수 있다.

3) 내부자 공격

서버의 내부자 공격을 막기 위하여 사용자의 ID_i는 제공하지만 패스워드는 암호학적 해시함수를 이용하여 제공하기 때문에 PW_i는 노출되지 않는다.

4) 재전송 공격

타임스탬프를 이용하여 메시지의 유효성을 검사하고 있기 때문에 시간차를 두고 공격을 수행하는 재전송 공격으로부터 안전할 수 있다. 만약 공격자가 사용자의 메시지를 저장하고, 재전송할 경우 그 메시지는 인증 단계에서 검증을 통과할 수 없다.

2.2 효율성

제안한 프로토콜에서는 Kim의 개선된 프로토콜에서와 마찬가지로 이용되는 랜덤 값의 수를 3개에서 2개로 줄임으로써 로그인 단계에서의 해쉬 연산의 수를 1회 감소시키고 통신량도 감소시켰다.

표 3 . 효율성 분석
Table 3. Analysis of efficiency

		제안프로토콜	Choi의 프로토콜
해시 연산	LP	5H,1P	6H,1P
	AP	5H	5H
	TP	5H	5H

H: 해시연산, P:지수연산, LP:로그인단계, AP:인증단계
TP:추적단계

V. 결론 및 향후 연구

최근 스마트카드 기반의 인증 기법들은 사용자 익명성을 제공하면서도 악의적인 사용자를 추적하는 연구가 활발하다.

본 논문에서는 Kim과 Choi 등이 제안한 프로토콜에서 로그인 단계의 계산 오류 문제점을 제시하고 정당한 사용자가 제 3자에 대한 익명성을 제공받지 못함을 보였다. 그리고 Kim의 프로토콜과 Choi의 프로토콜을 개선하여 발견된 문제점을 해결함과 동시에 사용자의 익명성과 추적성이 제공되는 개선된 프로토콜을 제안하였다. 제안한 프로토콜은 Kim의 기법과 Choi의 기법에서 발견된 계산 오류 문제점을 해결하였으며 정당한 사용자의 위장공격으로부터 안전하게 설계되었다. 또한 프로토콜에서 사용되는 사용자 생성 랜덤 값의 수를 조정함으로써 해쉬 연산의 수와 통신량을 감소시켰다.

제안한 프로토콜은 기존의 프로토콜과 마찬가지로, 불추적 익명성을 제공하지 못한다. 향후, 이를 제공하는 프로토콜에 대한 연구가 진행되어야 할 것이다.

참고문헌

[1] H. Y. Chien and C. H. Chen, "A Remote Authentication Scheme Preserving User Anonymity", IEEE AINA'05, Vol. 2, pp. 245-248, 2005.

[2] Z. Chai, Z. Cao and R. Lu, "Efficient Password-Based Authentication and Key Exchange Scheme Preserving User Privacy", WASA'06, LNCS 4138, pp. 467-477, 2006.

[3] M. L. Das, A. Saxena, and V. P. Gulati, "A Dynamic ID-based Remote User Authentication Scheme", IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, pp. 629-631, 2004.

[4] L. Lamport, "Password Authentication with Insecure Communication", Communications of the ACM, Vol. 24, No. 11, pp. 770-772, 1981.

[5] 김세일, 천지영, 이동훈, "추적이 가능한 스마트카드 사용자 인증 기법", 한국정보보호학회논문지, 제 18권, 제 5호, pp. 31-39, 2008.

[6] 최종석, 신승수, "사용자 익명성을 제공하는 추적 가능한 인증 프로토콜", 한국콘텐츠학회논문지, 제 9권, 제 4호, pp. 95-102, 2009.

저 자 소 개

용 승 림

2006 : 이화여자대학교 공학박사
2006-2007 : 이화여자대학교 컴퓨터정보공학부 전임강사
2008 - 현재 : 인하공업전문대학 컴퓨터시스템과 전임강사
관심분야 : 정보보호, 암호프로토콜, 디지털 저작권 보호

조 태 남

2004년 : 이화여자대학교 과학기술대학원 컴퓨터학과(공학박사)
1988년 - 1996년 :
한국전통신연구원 선임연구원
2004년 - 2005년 :
이화여자대학교 컴퓨터학과 전임강사
2005년 - 현 재 :
우석대학교 정보보안학과 조교수
관심분야 : 키관리, IPTV, TPM, 암호프로토콜 등