Journal of Korea Multimedia Society Vol. 20, No. 4, April 2017(pp. 614-628)
https://doi.org/10.9717/kmms.2017.20.4.614

Privileged-Insider 공격에 안전한 원격 사용자 인증 프로토콜

이성엽[†], 박요한[††], 박영호[†††]

Secure Remote User Authentication Protocol against Privileged-Insider Attack

SungYup Lee[†], YoHan Park[††], YoungHo Park[†††]

ABSTRACT

Recently, Due to the rapid development of the internet and IT technology, users can conveniently use various services provided by the server anytime and anywhere. However, these technologies are exposed to various security threat such as tampering, eavesdropping, and exposing of user's identity and location information. In 2016, Nikooghadam et al. proposed a lightweight authentication and key agreement protocol preserving user anonymity. This paper overcomes the vulnerability of Nikooghadam's authentication protocol proposed recently. This paper suggests an enhanced remote user authentication protocol that protects user's password and provides perfect forward secrecy.

Key words: Remote User Authentication Protocol, Biometric, Privileged-Insider Attack, Symmetric Key

1. 서 론

최근 인터넷과 IT 기술의 급속한 발전으로 인하여 사용자들은 네트워크상의 다양한 자원들을 실시간으로 이용할 수 있게 되었다. 특히, 이러한 기술들은 사람들에게 보다 편리한 삶을 제공하고 삶의 질은 향상 시키고 있다. 하지만, 공개된 채널로 정보를 교환하므로 사용자의 중요 정보 노출, 메시지의 도청 및 변조 등 다양한 위험 요소에 노출되어 있으며 서비스 환경에 따른 제약점 역시 존재하고 있다. 이를 해결하기 위해서 제한된 환경을 고려한 보안 프로토콜 연구가 활발히 이루어지고 있다[1-4].

인증 프로토콜은 대부분 패스워드 기반의 one-factor 인증 방법을 사용하고 있다. 하지만 패스워드를 사용하는 one-factor 인증 방법은 공격자가 패스워드 추측 공격을 통하여 패스워드가 노출되면서 two-factor 인증 방법이 제안 되었다[5-6]. Two-factor 인증 방법은 패스워드와 스마트카드를 함께 사용하여 보안적인 측면을 향상 시켰지만 훔친 스마트카드 공격(stolen smart card attack)에 취약하게 되었다. 최근에는 two-factor 인증 방법을 보완하여 three-factor 인증 방법이 제안되고 있다. Three-factor 인증 방법은 패스워드, 스마트카드 및 생체정보를 이용한 방법이다[7-8]. 생체정보를 암호화하기 위해서 퍼지 추출(fuzzy extraction)을 사용하고 있으며 기존에 사용하던 bio-hashing의 문제점을 보완

※ Corresponding Author : YoungHo Park, Address: (702-701) 80 Daehakro, Bukgu, Daegu, Korea, TEL : +82-53-950-7842, FAX : +82-53-950-5505, E-mail : parkyh@knu.ac.kr
Receipt date : Dec. 9, 2016, Revision date : Jan. 26, 2017
Approval date : Feb. 23, 2017

[†] School of Electronics Engineering, Kyungpook National University (E-mail : lsy8837@naver.com)
[††] Division of Information Technology, Korea Nazarene University (E-mail : hanny12@gmail.com)
[†††] School of Electronics Engineering, Kyungpook National University

한 방법이다. 프로토콜 안에서 노이즈를 제어하기 위하여 헬퍼스트링 변수를 만들어 랜덤 스트링 값이 일정하게 생성되는 장점이 있으며 생체정보의 프라이버시도 보호 할 수 있다[9-11]. 따라서 최근 많은 프로토콜에서 생체정보와 퍼지추출을 이용한 three-factor 인증 방법을 사용하고 있다.

최근 연구된 사용자 인증의 대표적 방식들은 다음과 같다. 2012년 Wang[12] 등은 스마트카드와 패스워드를 사용한 two-factor 원격사용자 인증 방식을 제안하였다. 하지만 2013년 Chang[4]등은 Wang 등의 프로토콜은 DoS(denial of service) 공격 및 익명성을 제공하지 못하는 취약점을 보이며 Wang 등의 문제점을 보완한 프로토콜을 제안하였다. 2014년 Kumari[13] 등은 Chang 등의 프로토콜의 취약점을 분석하여 오프라인 패스워드 추측, 위장 및 DoS 공격의 취약점을 보이며 향상된 원격 사용자 인증 프로토콜을 제안하였다. Kumari 등은 기존의 자신의 프로토콜에 타임스탬프를 추가하여 보안이 향상된 패스워드 인증 프로토콜을 제안하였다. 하지만 Chaudhry [14] 등은 Kumari 등의 인증 프로토콜은 훔친 스마트카드 공격 및 익명성을 제공하지 못하는 문제점을 보이며 프라이버시를 보장하는 원격 사용자 인증 프로토콜을 제안하였다. 최근 Nikooghadam[15] 등은 Kumari 등과 Chaudhry 등의 프로토콜의 취약점을 분석하여 익명성 및 패스워드 추측 공격이 가능함을 보이며 사용자의 익명성을 제공하는 경량화 방식의 인증 프로토콜을 제안하였다.

본 논문에서는 Nikooghadam 등의 프로토콜을 분석하여 권위 있는 내부자 공격(privileged-insider attack), 순방향 안전성(perfect forward secrecy), DoS 공격에 취약점 및 패스워드 단계의 문제점이 있음을 분석하였으며 이러한 문제점을 개선하여 생체정보 기반의 인증 및 키 합의 프로토콜을 제안한다. 생체정보를 이용하면 나누거나 위조할 수 없고 잃어버리거나 잊어버릴 수 없으며 패스워드보다 추측하기 어려운 장점이 있다.

논문의 구성은 다음과 같다. 2장에서 퍼지 추출, 기호 및 공격자의 능력을 설명하고 3장은 Nikooghadam[15] 등의 인증 프로토콜을 등록, 로그인 & 인증, 패스워드 변경 단계로 설명한다. 4장은 Nikooghadam 등의 프로토콜이 권위 있는 내부자 공격, 순방향 안전성, DoS 공격 및 패스워드 변경 단계의 취약점을 보인다. 5장에서는 제안한 인증 프로토콜을 등록, 로그인 & 인증, 패스워드 변경 단계로 설명한다. 6장은 제안한 프로토콜을 보안 분석하고 결론은 7장에서 기술한다.

2. 관련 연구

2.1 퍼지추출(fuzzy extraction)

퍼지추출은 사용자의 생체 정보를 하나의 키로 생성해내는 방법이다. 생성된 키가 암호학적으로 사용되기 위해서 충분한 엔트로피가 보장되어야 한다. 퍼지추출을 이용한 인증 기술은 생체정보의 노이즈를 처리할 수 있고 생체 정보에 대한 프라이버시를 보장할 수 있다. 퍼지추출 과정은 생체정보에 헬퍼스트링을 사용하여 노이즈를 제어하며 랜덤 스트링을 생성하여 인증 과정에서 사용되고 있다. 퍼지 추출은 Gen(generate)와 Rep(reproduce)로 구성되어 있다.

① $Gen(B_i) = (R_i, P_i)$. 생체 인식 템플릿 B_i를 입력하여 비밀 키 R_i를 생성하는 확률적인 알고리즘이다. 랜덤 스트링인 R_i를 얻기 위해서 헬퍼스트링인 P_i의 도움이 필요하다.

② $\mathrm{Rep}(B_i^{'}, P_i) = (R_i)$. 생체 정보와 헬퍼스트링 P_i를 사용하여 랜덤 스트링 R_i를 다시 만들어내는 결정론적 알고리즘이다. 사용자의 생체 정보를 입력하게 되면 헬퍼 스트링인 P_i가 노이즈를 제어하여 오차범위 안에서 동일한 R_i를 생성한다.

2.2 기호

본 절에서는 Nikooghadam[15] 등의 프로토콜 및 제안한 프로토콜에서 사용되는 기호들의 정의를 다음 Table 1과 같이 나타낸다.

2.3 공격자의 능력

공격자 $\mathcal{A}$의 능력을 다음과 같이 가정한다.

- $\mathcal{A}$는 로그인 & 인증 단계에서 유저와 서버 사이에 전송되는 모든 통신채널(공개 채널)의 정보를 수정, 삭제, 삽입, 얻을 수 있다.
- $\mathcal{A}$는 유저의 스마트카드를 얻거나 훔칠 수 있고, 스마트카드 안에 저장되어 있는 정보를 얻을 수

Table 1. Notations

Notation	Meaning
U_i	User i
S_j	Server j
SC	Smartcard of user
ID_i	Identity of U_i
PW_i	Password of U_i
B_i	Biometric template of U_i
y_i	A random number to the U_i selected by S_j
x	The server private key
SK	The session key between the U_i and S_j
T_i	Timestamp
$\|$	Concatenate operation
$\oplus$	The exclusive-OR operation(XOR)
$E_k(.)/D_k(.)$	The symmetric encryption/decryption with the key k
$h(.)$	A secure one-way hash function

가 있다.

• $\mathcal{A}$가 insider 공격이나 outsider 공격을 통하여 서버에 인증을 받을 수가 있다.

3. Nikooghadam 등의 인증 방식

Nikooghadam[15] 등은 2016년 Kumari[13] 등과 Chaudhry[14] 등의 문제점을 개선한 사용자 익명성을 제공하는 인증 프로토콜을 제안하였다. 기존의 프로토콜은 해시 연산과 XOR 연산을 사용하였지만 Nikooghadam의 프로토콜에서는 대칭키 연산을 사용하여 공개채널의 정보의 기밀성 및 무결성을 보장하고 있다. 본 장에서는 Nikooghadam 등의 프로토콜을 등록단계, 로그인 & 인증 단계 및 패스워드 단계로 설명하고 있다.

3.1 등록 단계

본 단계에서 사용자 U_i는 서버 S_j에게 등록 요청 메시지를 보내고 서버는 합법적인 사용자로 스마트카드를 다음과 같이 발급할 수 있다

1) 사용자 U_i는 자신의 신원 ID_i, 패스워드 PW_i 및 랜덤넘버 r을 선택하고 $RPW_i = h(ID_i \| r \| PW_i)$를 계산을 한다. 사용자 U_i는 등록 요청 메시지=$\{RPW_i, ID_i\}$를 안전한 채널로 서버 S_j에게 전송한다.

2) 서버 S_j는 사용자 U_i에게 받은 등록 요청 메시지 정보$\{ID_i, RPW_i\}$를 받고 랜덤 넘버 N을 선택하여 J_i, L_i 및 RID_i를 다음과 같이 계산한다.

$$J_i = h(ID_i \| x)$$
$$L_i = J_i \oplus RPW_i \quad (1)$$
Chooses a random number N
$$RID_i = E_x(ID_i \| N)$$

3) 서버 S_j는 $\{L_i, RID_i, E_{key}(.), D_{key}(.), h(.)\}$를 스마트카드 ($SC$)안에 저장하여 사용자 U_i에게 안전한 채널로 전송한다.

4) 서버 S_j에게 스마트카드를 받은 사용자 U_i는

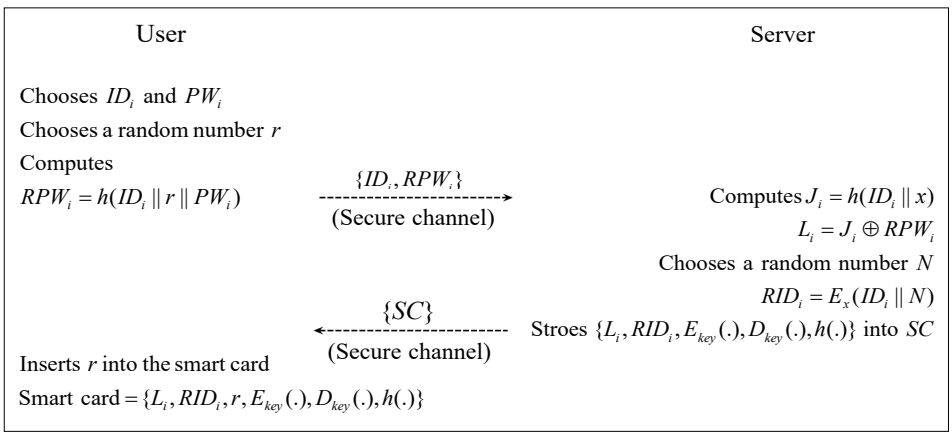

Fig. 1. Registration phase of Nikooghadam et al.'s protocol.

사용자의 랜덤 값 r을 스마트카드 안에 저장한다. 최종적으로 스마트카드 안에 저장된 정보는 $SC=\{L_i, RID_i, r, E_{key}(.), D_{key}(.), h(.)\}$이다.

3.2 로그인 & 인증 단계

본 단계에서 사용자 U_i는 로그인 요청 메시지를 서버에게 전송을 하고 서버 S_j는 올바른 사용자의 메시지인지 확인을 하고 응답 메시지를 사용자 U_i에게 전송을 한다. 응답메시지를 받은 사용자는 받은 메시지가 올바른 서버의 메시지인지 확인을 하고 세션키를 계산하여 서버에게 인증을 받는 단계이다.

1) 사용자 U_i는 스마트카드를 리더기에 넣고 ID_i, PW_i를 입력한다. 입력한 ID_i, PW_i를 사용하여 $J_i \leftarrow B_i \oplus h(ID_i \| r \| PW_i)$를 계산하고 랜덤 넘버 RN_i와 현재 타임스탬프 T_i를 선택한다.

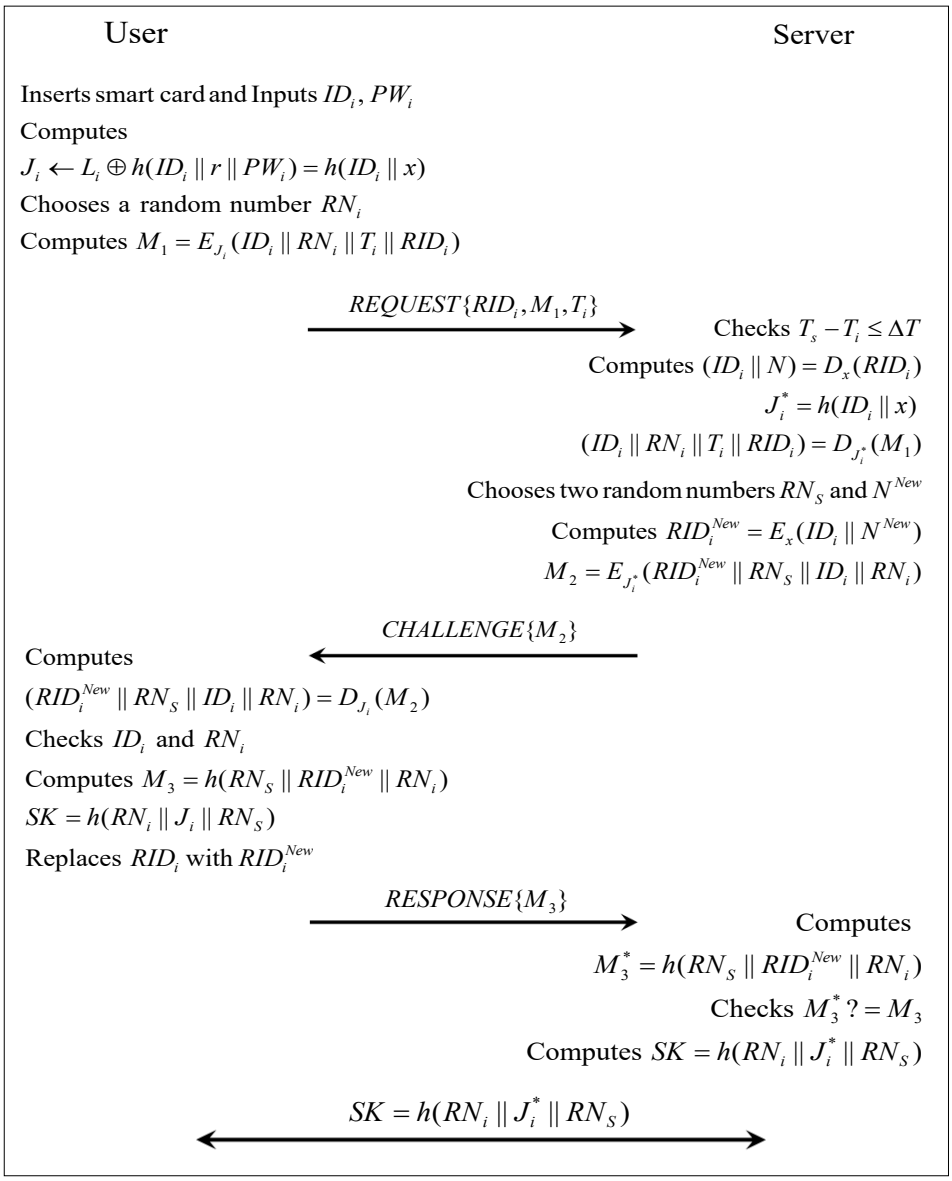

Fig. 2. Login & authentication phase of Nikooghadam et al.'s protocol.

RN_i와 T_i를 이용하여 $M_1 = E_{J_i}(ID_i \| RN_i \| T_i \| RID_i)$를 계산하여 서버 S_j에게 로그인 요청 메시지 = $\{RID_i, M_1, T_i\}$를 통신채널로 전송한다.

2) 로그인 요청 메시지를 받은 서버 S_j는 타임스탬프를 체크하며 다음 단계로 RID_i, J_i^*, M_1을 계산한다. 서버 S_j는 랜덤 넘버 RN_S와 N^{New}를 생성하여 RID_i^{New}, M_2를 계산하고 M_2를 사용자 U_i에게 전송한다.

$$(ID_i \| N) = D_x(RID_i)$$

$$J_i^* = h(ID_i \| x)$$

$$(ID_i \| RN_i \| T_i \| RID_i) = D_{J_i^*}(M_1) \quad (2)$$

$$RID_i^{New} = E_x(ID_i \| N^{New})$$

$$M_2 = E_{J_i^*}(RID_i^{New} \| RN_S \| ID_i \| RN_i)$$

3) 서버 S_j에게 M_2를 받은 사용자 U_i는 다음과 같이 M_2, M_3 및 SK를 계산하고 M_3를 서버 S_j에게 전송한다.

$$(RID_i^{New} \| RN_S \| ID_i \| RN_i) = D_{J_i}(M_2)$$

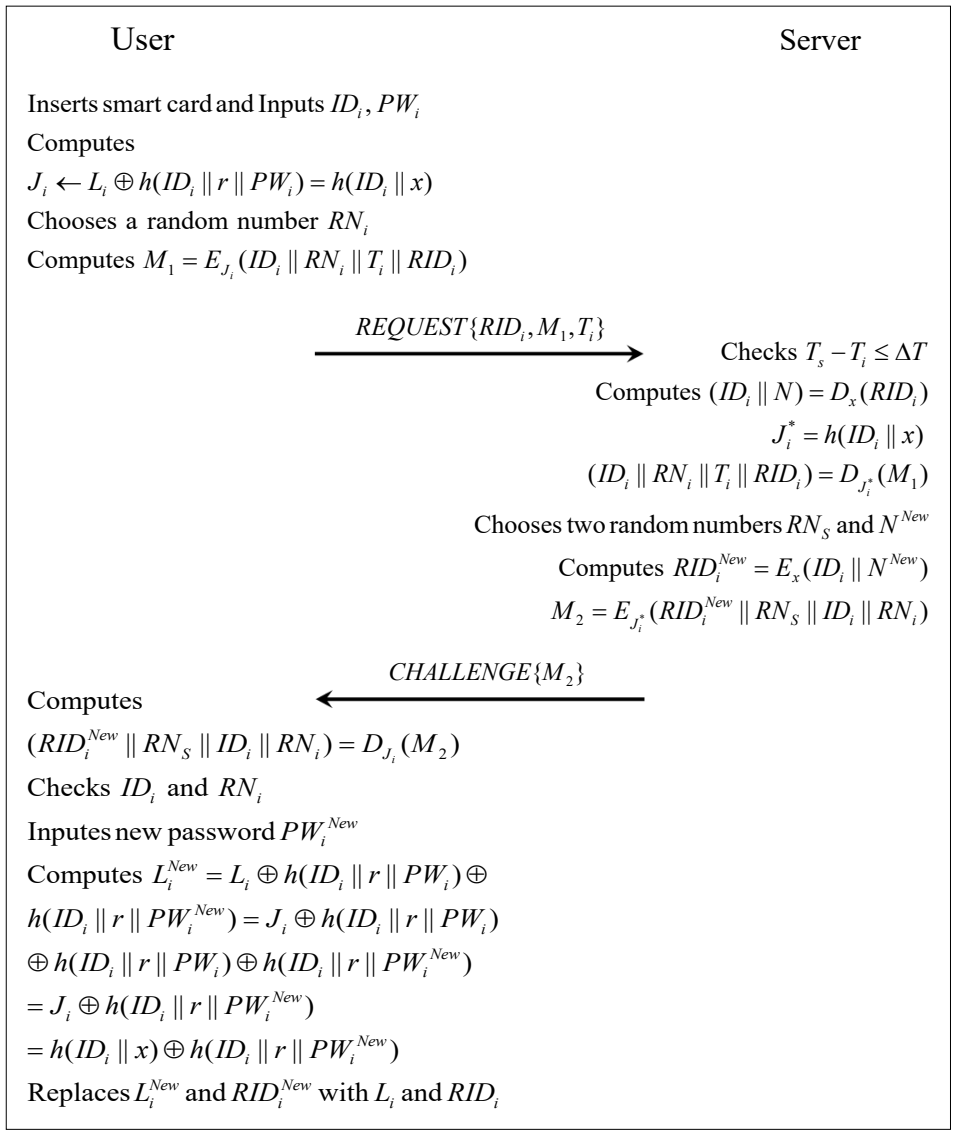

Fig. 3. Password change phase of Nikooghadam et al.'s protocol.

Checks ID_i and RN_i

$$M_3 = h(RN_S \| RID_i^{New} \| RN_i) \quad (3)$$

$$SK = h(RN_i \| J_i \| RN_S)$$

Replaces RID_i with RID_i^{New}

4) 사용자 U_i에게 M_3를 받은 서버 S_j는 $M_3^* = h(RN_S \| RID_i^{New} \| RN_i)$를 계산하고 사용자 U_i에게 받은 M_3와 M_3^*를 체크한다. 만약 값이 같으면 $SK = h(RN_i \| J_i^* \| RN_S)$를 계산하고 같지 않으면 세션을 종료한다.

5) 사용자 U_i와 서버 S_j가 성공적으로 상호인증을 하면 최종적으로 세션 키(SK)를 확립하고 인증 단계를 마치게 된다.

3.3 패스워드 변경 단계

본 단계에서 사용자 U_i가 패스워드를 변경을 하기 위한 단계로 사용자는 스마트카드를 리더기에 넣고 자신의 아이디와 패스워드를 입력하고 로그인 요청 메시지를 서버 S_j에게 전송을 하게 된다. 메시지를 받은 서버는 올바른 사용자가 접속하였는지 확인하고 응답메시지를 보내게 되면 사용자는 패스워드를 변경할 수 있다. 패스워드 변경 단계는 다음과 같다.

1) 사용자 U_i는 스마트카드를 리더기에 넣고 ID_i, PW_i를 입력한다. 입력한 ID_i, PW_i를 사용하여 $J_i \leftarrow B_i \oplus h(ID_i \| r \| PW_i)$를 계산하고 랜덤 넘버 RN_i와 현재 타임스탬프 T_i를 선택한다. RN_i와 T_i를 이용하여 $M_1 = E_{J_i}(ID_i \| RN_i \| T_i \| RID_i)$를 계산하여 서버 S_j에게 로그인 요청 메시지 = $\{RID_i, M_1, T_i\}$를 공개 채널로 전송한다.

2) 로그인 요청 메시지를 받은 서버 S_j는 타임스탬프를 체크하며 다음 단계로 RID_i, J_i^*, M_1을 계산한다. 서버 S_j는 랜덤 넘버 RN_S와 N^{New}를 생성하여 RID_i^{New}, M_2를 계산하고 M_2를 사용자 U_i에게 전송한다.

$$(ID_i \| N) = D_x(RID_i)$$

$$J_i^* = h(ID_i \| x)$$

$$(ID_i \| RN_i \| T_i \| RID_i) = D_{J_i^*}(M_1) \quad (4)$$

$$RID_i^{New} = E_x(ID_i \| N^{New})$$

$$M_2 = E_{J_i^*}(RID_i^{New} \| RN_S \| ID_i \| RN_i)$$

3) 서버 S_j에게 M_2를 받은 사용자 U_i는 다음과 같이 M_2를 계산하고 ID_i, RN_i를 체크하고 같으면 새로운 패스워드 PW_i^{new}를 입력한다. 새로운 패스워드 PW_i^{new}를 이용하여 L_i^{New}를 계산하고 스마트카드에 L_i^{New}와 RID_i^{New}를 업데이트한다.

$$(RID_i^{New} \| RN_S \| ID_i \| RN_i) = D_{J_i}(M_2)$$

Checks ID_i and RN_i (5)

$$L_i^{New} = L_i \oplus h(ID_i \| r \| PW_i) \oplus h(ID_i \| r \| PW_i^{New})$$

Replaces L_i^{New} and RID_i^{New} with L_i and RID_i

4. Nikooghadam 등의 방식의 취약점 분석

Nikooghadam[15] 등의 인증 프로토콜은 사용자의 프라이버시를 보호하는 경량화 인증 프로토콜이다. 하지만 권위 있는 내부자 공격, 순방향 안전성, DoS 공격에 취약하고 패스워드 변경단계가 효율적이지 못하다. 본 장에서는 Nikooghadam 등의 취약점을 분석한다.

4.1 권위 있는 내부자 공격

권위 있는 내부자 공격이란 공격자 $\mathcal{A}$가 등록단계의 정보나 데이터베이스에 저장되어 있는 정보를 이용하여 사용자의 중요정보(패스워드, 세션 키)를 찾을 수 있는 공격이다. Nikooghadam 등의 프로토콜 역시 공격자 $\mathcal{A}$가 등록단계에 전송되는 메시지를 획득하면 다음과 같이 쉽게 사용자의 패스워드 PW_i를 계산할 수 있다.

- $\mathcal{A}$는 등록단계에 전송되는 메시지 $\{ID_i, RPW_i\}$를 알고 있다.
- $\mathcal{A}$는 사용자의 스마트카드 안에서 r을 되찾아온다.
- $\mathcal{A}$는 사용자의 패스워드 PW_i^g를 추측한다.
- $\mathcal{A}$는 $RPW_i^* = h(ID_i \| r \| PW_i^g)$를 계산한다.
- $\mathcal{A}$는 등록단계에서 얻은 RPW_i와 계산한 RPW_i^*가 같은지 체크를 한다. 만약 값이 같으면, $\mathcal{A}$는 사용자의 패스워드 PW_i를 성공적으로 계산한 것이다. 만약 값이 다르면, $\mathcal{A}$는 3단계부터 반복 실행하면 된다.

4.2 순방향 안전성

순방향 안전성이란 long-term key가 공격자 $\mathcal{A}$에

게 노출이 되더라도 이전에 세션 키에 지장을 주지 않아야한다. 하지만 Nikooghadam등의 프로토콜은 공격자 $\mathcal{A}$가 long- term key인 x를 알게 되면 서버 S_j와 사용자 U_i 사이의 세션 키를 다음과 같이 쉽게 계산을 할 수 있다.

• $\mathcal{A}$가 long-term key x를 안다고 가정을 한다.

• $\mathcal{A}$는 공개채널로 전송되는 메시지의 정보 $\{RID_i, M_1, T_i, M_2, M_3\}$를 획득한다.

• $\mathcal{A}$는 RID_i, J_i^*, M_1 및 M_2를 다음과 같이 계산한다.

$$\begin{aligned}
&(ID_i \| N) = D_x(RID_i)\\
&J_i^* = h(ID_i \| x)\\
&(ID_i \| RN_i \| T_i \| RID_i) = D_{J_i^*}(M_1)\\
&(RID_i^{New} \| RN_S \| ID_i \| RN_i) = D_{J_i^*}(M_2)
\end{aligned} \tag{6}$$

• $\mathcal{A}$는 계산한 값을 이용하여 세션 키를 쉽게 계산할 수 있다.

$$SK = h(RN_i \| J_i^* \| RN_S) \tag{7}$$

4.3 DoS 공격

DoS 공격은 서버를 악의적으로 공격해 해당 서버의 자원을 부족하게 하여 원래의 의도된 용도로 사용하지 못하게 하는 공격이다. Nikooghadam 등의 프로토콜은 로그인 단계에서 올바른 사용자가 접속하였는지 검증하는 단계가 생략이 되어있다. 따라서 공격자 $\mathcal{A}$가 자신의 로그인 요청 메시지를 만들어 다음과 같이 서버에게 DoS 공격을 할 수 있다.

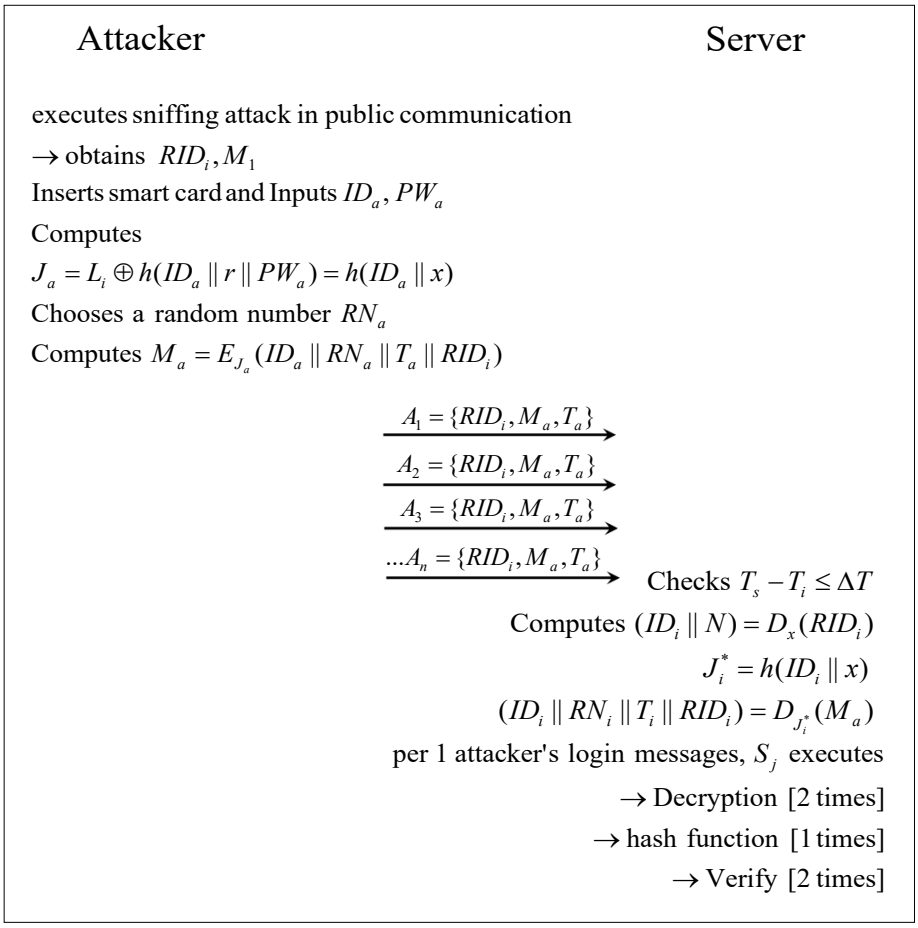

Fig. 4. DoS attack.

4.4 패스워드 변경 단계의 문제점

Nikooghadam 등의 패스워드 변경단계에는 사용자가 서버와 함께 패스워드를 변경하고 있다. 최근 제안된 패스워드 변경 단계는[13] 사용자가 패스워드 변경을 원하면 스마트카드를 이용하여 독자적으로 패스워드를 변경할 수 있다. 하지만 Nikooghadam 등의 프로토콜은 사용자와 서버가 통신하여 패스워드를 변경하므로 불필요한 메시지 전송, 계산, 랜덤 넘버 생성 및 스마트카드 변수 업데이트를 하고 있다. 따라서 Nikooghadam 등의 패스워드 변경 단계는 효율적으로 변경 되어야 한다.

5. Privileged−insider 공격에 안전한 원격 사용자 인증 프로토콜

본 절에는 Nikooghadam[15] 등의 인증프로토콜의 취약점을 보완한 원격 사용자 인증 프로토콜을 제안한다. 제안한 프로토콜은 생체정보 기반의 원격 사용자 프로토콜이며 생체정보를 랜덤 변수로 사용하기 위해 퍼지추출을 사용하였다. 프로토콜의 설명은 등록 단계, 로그인 & 인증 단계, 패스워드 변경 단계로 설명하고 있다.

5.1 등록 단계

본 단계에서 사용자 U_i는 서버 S_j에게 등록 요청 메시지를 보내고 서버는 사용자의 다이나믹한 MID_i를 데이터베이스에 저장하고 스마트카드를 다음과 같이 발급받을 수 있다.

1) 사용자 U_i는 자신의 신원 ID_i 및 패스워드 PW_i를 선택하고 생체정보 B_i를 입력하여 퍼지 추출 연산을 다음과 같이 $<R_i, P_i>$=Gen(B_i) 계산한다. U_i는 랜덤 스트링 R_i를 이용하여 $RPW_i = h(PW_i \| R_i)$, $AID_i = h(ID_i \| R_i)$를 계산하여 등록 요청 메시지=$\{AID_i, RPW_i\}$를 안전한 채널로 서버 S_j에게 전송한다.

2) 서버 S_j는 사용자 U_i에게 받은 등록 요청 메시지 정보$\{AID_i, RPW_i\}$를 받고 사용자 U_i의 랜덤 넘버 y_i을 선택하여 J_i, L_i, V_i, MID_i 및 RID_i를 다음과 같이 계산한다.

$$\begin{aligned} J_i &= h(AID_i \| x) \oplus y_i \\ L_i &= J_i \oplus RPW_i \\ V_i &= h(AID_i \| RPW_i \| y_i) \\ MID_i &= AID_i \oplus y_i \\ RID_i &= E_x(MID_i \| N) \end{aligned} \tag{8}$$

3) 서버 S_j는 $\{MID_i, y_i\}$를 데이터베이스에 저장하고 스마트카드 안에 $\{L_i, V_i, RID_i, E_{key}(.), D_{key}(.), h(.)\}$ 저장하여 사용자 U_i에게 $\{SC \ \& \ y_i\}$를 안전한 채널로 전송한다.

4) 서버 S_j에게 메시지를 받은 사용자 U_i는 $Q_i = h(ID_i \| R_i \| PW_i) \oplus y_i$를 계산하여 스마트카드 안에

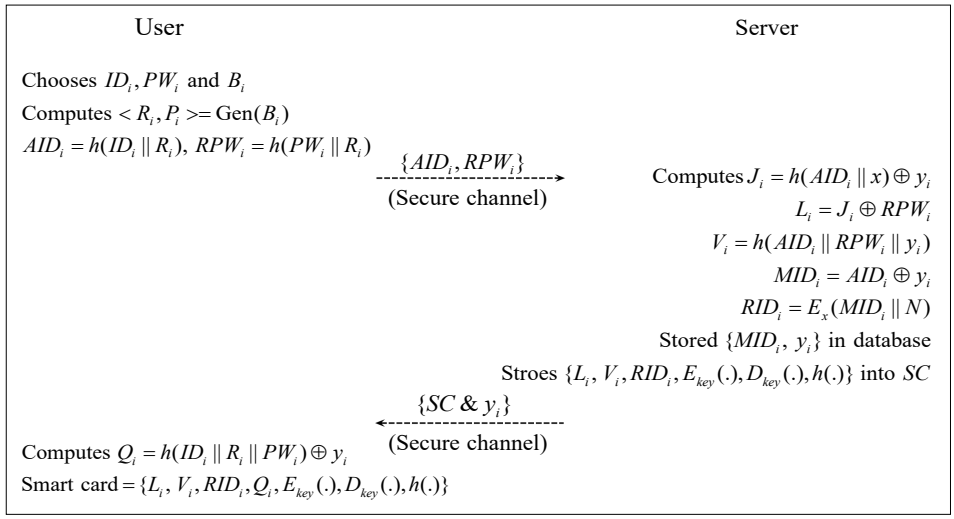

Fig. 5. Registration phase of proposed protocol.

저장한다. 최종적으로 스마트카드 안에 저장된 정보는 $SC=\{L_i, V_i, RID_i, Q_i, E_{key}(.), D_{key}(.), h(.)\}$이다.

5.2 로그인 & 인증 단계

본 단계에서 사용자 U_i와 서버 S_j 사이에 상호인증을 통하여 세션 키를 확립하는 단계이다. 서버는 올바른 사용자의 접속을 확인하기 위해 2번의 검증단계를 하게 되고 사용자는 올바른 서버의 메시지인지 확인하기 위해 1번의 검증 단계를 하게 된다. 세션 키는 다음과 같이 계산을 할 수 있다.

1) 사용자 U_i는 스마트카드를 리더기에 넣고 ID_i, PW_i 및 B_i를 입력한다. 입력한 B_i를 사용하여 $R_i^* = Rep(B_i^*, P_i)$를 퍼지추출하고 y_i^*, RPW_i^*, AID_i^* 및 V_i^*를 계산한다.

$$y_i^* \leftarrow h(ID_i \| R_i^* \| PW_i) \oplus Q_i$$
$$RPW_i^* = h(PW_i \| R_i^*) \quad (9)$$

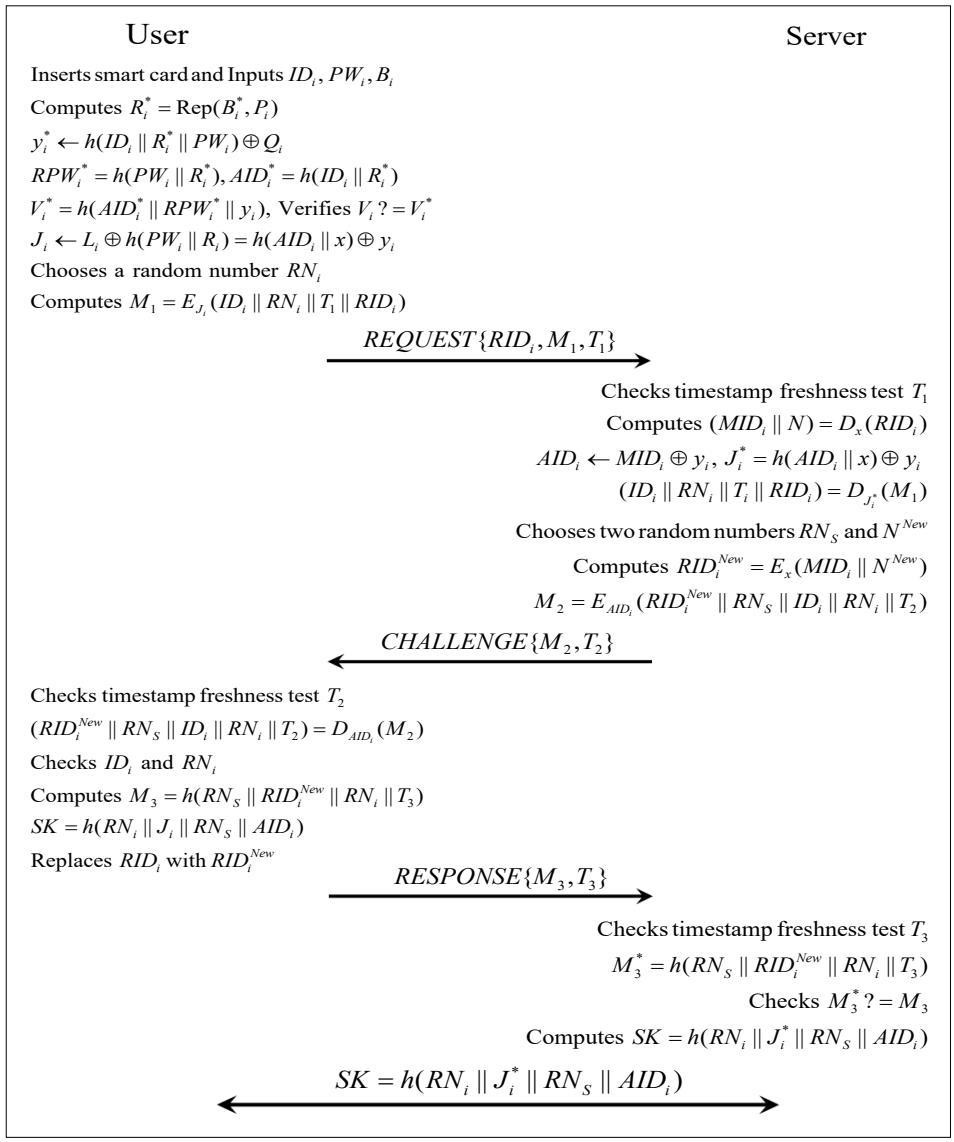

Fig. 6. Login & authentication phase of proposed protocol.

$AID_i^* = h(ID_i \| R_i^*)$

$V_i^* = h(AID_i^* \| RPW_i^* \| y_i^*)$

2) 만약 V_i^*와 V_i가 같지 않으면 세션을 종료하고, 같으면 올바른 사용자 U_i가 접속한 것으로 생각하여 J_i와 M_1을 다음과 같이 계산을 하고 서버 S_j에게 로그인 요청 메시지 = $\{RID_i,\ M_1, T_1\}$를 전송한다.

$$J_i \leftarrow L_i \oplus h(PW_i \| R_i) = h(AID_i \| x) \oplus y_i \quad (1)$$
$$M_1 = E_{J_i}(ID_i \| RN_i \| T_1 \| RID_i)$$

3) 사용자 U_i에게 메시지를 받은 서버 S_j는 타임스탬프 체크를 하고 다음과 같이 RID_i, AID_i, J_i^* 및 M_1을 계산하고 랜덤 넘버 RN_S, N^{New}를 선택하여 RID_i^{New}와 M_2를 계산해서 $\{M_2,\ T_2\}$를 서버 S_j에게 전송한다.

Checks timestamp freshness test T_1

$(MID_i \| N) = D_x(RID_i)$

$AID_i \leftarrow MID_i \oplus y_i$

$J_i^* = h(AID_i \| x) \oplus y_i \quad (11)$

$(ID_i \| RN_i \| T_i \| RID_i) = D_{J_i^*}(M_1)$

$RID_i^{New} = E_x(MID_i \| N^{New})$

$M_2 = E_{AID_i}(RID_i^{New} \| RN_S \| ID_i \| RN_i \| T_2)$

4) 서버 S_j에게 메시지를 받은 사용자 U_i는 다음과 같이 M_2, M_3 및 SK를 계산하고 $\{M_3, T_3\}$를 서버 S_j에게 전송한다.

Checks timestamp freshness test T_2

$(RID_i^{New} \| RN_S \| ID_i \| RN_i \| T_2) = D_{AID_i}(M_2)$

Checks ID_i and RN_i

$M_3 = h(RN_S \| RID_i^{New} \| RN_i \| T_3) \quad (12)$

$SK = h(RN_i \| J_i \| RN_S \| AID_i)$

Replaces RID_i with RID_i^{New}

5) 사용자 U_i에게 메시지를 받은 서버 S_j는 타임스탬프를 체크하고 $M_3^* = h(RN_S \| RID_i^{New} \| RN_i \| T_3)$를 계산하고 사용자 U_i에게 받은 M_3와 M_3^*를 체크한다. 만약 값이 같으면 세션 키 $SK = h(RN_i \| J_i^* \| RN_S \| AID_i)$를 계산하고 같지 않으면 세션을 종료한다.

6) 사용자 U_i와 서버 S_j가 성공적으로 상호인증을 하면 최종적으로 세션 키가 확립된다.

5.3 패스워드 변경 단계

본 단계에서 사용자 U_i가 자신의 패스워드 PW_i를 변경하는 단계이다. Nikooghadam의 패스워드 변경 단계와 다르게 U_i는 서버 S_j의 인증을 받지 않고 패스워드를 다음과 같이 변경할 수 있다.

1) 사용자 U_i는 스마트카드를 리더기에 넣고 ID_i, PW_i, B_i 및 새로운 패스워드 PW_i^{New}를 입력한다. 입력한 생체정보 B_i를 퍼지추출을 이용하여 다음과

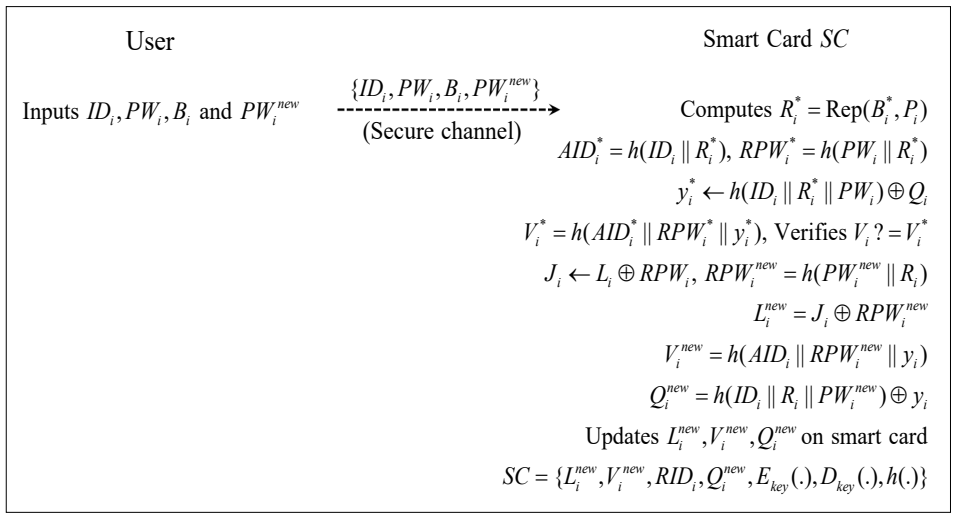

Fig. 7. Password change phase of proposed protocol.

같이 $R_i^* = Rep(B_i^*, P_i)$를 연산하고 AID_i^*, RPW_i^*, y_i^* 및 V_i^*를 계산한다.

$$\begin{aligned} &AID_i^* = h(ID_i \| R_i^*) \\ &RPW_i^* = h(PW_i \| R_i^*) \\ &y_i^* \leftarrow h(ID_i \| R_i^* \| PW_i) \oplus Q_i \\ &V_i^* = h(AID_i^* \| RPW_i^* \| y_i^*) \\ &\text{Verifies } \ V_i ? = V_i^* \end{aligned} \tag{13}$$

2) 만약 V_i^*와 V_i가 같지 않으면 세션을 종료하고, 같으면 올바른 사용자 U_i가 접속한 것으로 생각하여 다음과 같이 패스워드 변경 단계를 진행한다.

$$\begin{aligned} &J_i \leftarrow L_i \oplus RPW_i \\ &RPW_i^{New} = h(PW_i^{New} \| R_i) \\ &L_i^{New} = J_i \oplus RPW_i^{New} \\ &V_i^{New} = h(AID_i \| RPW_i^{New} \| y_i) \\ &Q_i^{New} = h(ID_i \| R_i \| PW_i^{New}) \oplus y_i \end{aligned} \tag{14}$$

3) 스마트카드는 계산한 L_i^{New}, V_i^{New}, Q_i^{New} 값을 기존의 L_i, V_i, Q_i에 저장하면 패스워드 변경 단계를 마친다.

6. 보안 분석

Nikooghadam[15] 등의 프로토콜은 해시 연산과 XOR 연산 및 대칭키 연산을 사용하여 효율적으로 프로토콜을 제안하였지만 권위 있는 내부자 공격, 순방향 안전성 및 DoS 공격의 취약점이 있다. 제안한 프로토콜은 권위 있는 내부자 공격에 안전하기 위해서 생체정보를 사용하여 해시 연산을 통하여 랜덤 값을 찾을 수 있게 변경하였고 순방향 안전성을 제공하기 위하여 마스터 키 x와 AID_i를 연산하여 대칭키로 사용하고 있다. 본 논문에서 제안한 프로토콜은 다음과 같은 informal analysis를 사용하여 보안 분석한다.

6.1 Informal analysis

6.1.1 권위 있는 내부자 공격

권위 있는 내부자 공격은 공격자가 등록단계에서 보내지는 메시지를 가로채거나 얻어서 사용자의 중요정보(패스워드 세션 키)를 계산할 수 있는 공격이다. Nikooghadam 등의 프로토콜은 공격자가 등록단계에서 보내지는 메시지와 스마트카드 안에 저장되어있는 사용자의 랜덤 넘버 r을 알고 있으면 쉽게 사용자의 패스워드를 계산할 수 있지만 제안한 프로토콜은 등록단계에서 생체 정보를 이용하여 가상의 아이디와 가상의 패스워드를 만들고 있다. 따라서 공격자가 등록단계에서 보내지는 메시지를 알고 있더라도 사용자의 패스워드를 찾을 수가 없다.

6.1.2 순방향 안전성

순방향 안전성은 공격자가 long-term key를 알고 있어도 이전의 세션 키에 영향을 줄 수 없어야 한다. 제안된 프로토콜에서는 공격자가 long-term key로 x를 알고 있다고 가정을 하여도 AID_i값을 알지 못하기 때문에 안전하다. 공격자가 세션 키를 계산하려면 RN_i, J_i, RN_S, AID_i를 알고 있어야 하는데 공격자는 랜덤 스트링 R_i와 서버가 선택한 유저의 랜덤넘버 y_i값이 없으면 모든 변수 값을 알 수는 없다. 따라서 제안한 프로토콜은 완전 순방향 안전성을 제공한다.

6.1.3 DoS 공격

DoS 공격은 공격자가 서버를 악의적으로 공격해 해당 서버의 자원을 부족하게 하여 원래의 의도된 용도로 사용하지 못하게 하는 공격이다. Nikooghadam 등의 프로토콜은 랜덤 넌스와 타임스탬프를 사용하여 DoS 공격에 안전하지만, 로그인 단계에서 올바른 사용자가 접속하였는지 검증하지 않고 있다. 그러므로 Nikooghadam 등의 프로토콜은 DoS 공격에 취약하다. 제안한 프로토콜은 로그인 인증 단계 동안 메시지를 받게 되면 타임스탬프 확인과 올바른 서버의 메시지인지, 사용자의 메시지인지를 검증을 하고 있다. 따라서 제안한 메시지는 DoS 공격에 안전하다.

6.1.4 익명성(anonymity)

익명성을 제공하기 위해서는 공격자가 사용자의 아이디를 계산하거나 얻을 수 없어야한다. 제안된 프로토콜에서 사용자의 아이디를 얻기 위해서는 $AID_i = h(ID_i \| R_i), Q_i = h(ID_i \| R_i \| PW_i) \oplus y_i,$ $M_1 = E_{J_i}(ID_i \| RN_i \| T_1 \| RID_i)$를 계산할 수 있어야 한다. 하지만 제안한 프로토콜은 공격자가 생체정보를 퍼지추출한 값 R_i도 모르고 J_i, RN_i, y_i를 계산할 수 없으

므로 익명성을 제공한다.

6.1.5 내부자 공격(insider attack)

내부자 공격은 합법적인 제3자의 사용자가 서버의 고유한 값을 계산하여 다른 사용자의 아이디나 패스워드 같은 중요한 정보를 계산하는 공격이다. 제안한 프로토콜에서 서버의 고유한 값 x를 $J_i = h(AID_i \| x)$와 같이 사용자마다 다이나믹한 값으로 변경했다. 그래서 공격자는 서버의 고유한 값을 얻을 수가 없다. 따라서 제안된 프로토콜은 인사이더 공격에 안전하다.

6.1.6 추측 공격(guessing attack)

공격자는 사용자의 아이디와 패스워드를 추측할 수 있다. 아이디 추측 공격은 공격자가 임의로 아이디 값을 추측하여 기존의 값과 비교하여 동일한 값을 얻어 아이디를 계산하는 공격이다. 제안한 프로토콜에서 아이디를 추측하려면 $AID_i = h(ID_i \| R_i)$, $Q_i = h(ID_i \| R_i \| PW_i) \oplus y_i$, $M_1 = E_{J_i}(ID_i \| RN_i \| T_1 \| RID_i)$을 알아야한다. 하지만 공격자는 위의 값을 얻을 수 있는 R_i, y_i, RN_i를 얻을 수 없다. 패스워드 추측 공격 역시 아이디 추측 공격과 비슷하게 공격을 할 수 있다. 공격자가 임의의 패스워드를 추측하여도 $RPW_i = h(PW_i \| R_i)$, $Q_i = h(ID_i \| R_i \| PW_i) \oplus y_i$의 값을 알아야한다. 공격자는 사용자의 생체정보 B_i의 랜덤 스트링 R_i 및 사용자의 랜덤 넘버 y_i도 모른다. 따라서 제안된 프로토콜은 아이디 및 패스워드 추측 공격에 안전하다.

6.1.7 훔친 스마트카드 공격

훔친 스마트카드 공격은 공격자가 사용자의 스마트카드를 획득하여 스마트카드 안에 정보로 아이디, 패스워드 및 중요한 정보를 계산하는 공격이다. 제안한 프로토콜에서 공격자가 스마트카드를 획득하게 되면 스마트카드 정보= $\{L_i, V_i, RID_i, Q_i, E_{key}(.), D_{key}(.), h(.)\}$를 사용할 수 있다. 하지만 공격자는 위의 정보를 가지고 $J_i = h(AID_i \| x) \oplus y_i$, $AID_i = h(ID_i \| R_i)$와 같은 중요 변수를 계산할 수 없다. 따라서 공격자가 스마트카드의 정보를 훔쳐도 세션키나 아이디, 패스워드 같은 중요한 정보를 얻을 수가 없다.

6.1.8 위장 공격(impersonation attack)

위장 공격에는 사용자 위장공격과 서버 위장공격 두 가지가 있다. 사용자 위장 공격을 하기 위해서는 사용자가 서버로 보내는 로그인 요청 메시지를 만들어야하고 서버 위장 공격을 하기 위해서는 사용자에게 보내는 서버의 응답 메시지를 생성해야한다. 제안된 프로토콜에서 로그인 요청 메시지는 $\{RID_i, M_1, T_i\}$고 응답 메시지는 $\{M_2, T_2\}$이다. 공격자가 로그인 메시지나 응답 메시지를 생성하려면 $MID_i, N, RN_i, ID_i, RN_S$를 알아야한다. 제안한 프로토콜에서 공격자는 J_i나 AID_i를 알 수가 없다. 따라서 공격자는 위의 값을 계산 할 수가 없으므로 위장공격에 안전하다.

6.1.9 재전송 공격(replay attack)

재전송 공격은 통신채널(공개채널)로 전송되는 로그인 요청 메시지나 응답메시지를 공격자가 가로

Table 2. Comparison of security

	Kumari et al. [13]	Chaudhry et al. [14]	Nikooghadam et al. [15]	Proposed
Privileged-insider attack	O	O	×	O
Perfect forward secrecy	O	×	×	O
DoS attack	×	O	×	O
Anonymity	×	O	O	O
Insider attack	×	O	O	O
Guessing attack	×	×	O	O
Stolen smart card attack	O	O	O	O
Impersonation attack	O	O	O	O
Replay attack	O	O	O	O

O : Secure , × : Insecure

Table 3. Comparison of computation

	Registration	Login & Authentication	Password change	Total
Kumari et al. [13]	$4t_{h(.)}+5t_{\oplus}$	$14t_{h(.)}+12t_{\oplus}$	$6t_{h(.)}+7t_{\oplus}$	$24t_{h(.)}+24t_{\oplus}$
Chaudhry et al. [14]	$4t_{h(.)}+5t_{\oplus}$ $+1t_{E/D}$	$11t_{h(.)}+14t_{\oplus}$ $+1t_{E/D}$	×	$15t_{h(.)}+19t_{\oplus}$ $+2t_{E/D}$
Nikooghadam et al. [15]	$2t_{h(.)}+1t_{\oplus}$ $+1t_{E/D}$	$6t_{h(.)}+1t_{\oplus}$ $+6t_{E/D}$	$10t_{h(.)}+8t_{\oplus}$ $+6t_{E/D}$	$18t_{h(.)}+10t_{\oplus}$ $+13t_{E/D}$
Proposed	$5t_{h(.)}+4t_{\oplus}$ $+1t_{F}+1t_{E/D}$	$10t_{h(.)}+4t_{\oplus}$ $+1t_{F}+6t_{E/D}$	$7t_{h(.)}+4t_{\oplus}$ $+1t_{F}$	$22t_{h(.)}+12t_{\oplus}$ $+3t_{F}+7t_{E/D}$

$t_{h(.)}$: hash function, $h_{\oplus}$: exclusive-OR operation(XOR)
$t_{E/D}$: encryption/decryption operation, t_{F} : fuzzy operation

채거나 얻어서 서버로 다시 보내 사용자와 서버사이의 세션 키를 얻거나 아이디, 패스워드 및 중요정보를 공격자가 계산할 수 있는 공격이다. 재전송 공격을 효과적으로 예방할 수 있는 방법은 타임스탬프의 사용이다. 제안한 프로토콜은 사용자와 서버가 메시지를 받게 되면 timestamp freshness test를 하게 된다. 따라서 재전송 공격에 안전하다.

6.2 연산량 분석

연산량 분석은 Kumari[13], Chaudhry[14], Nikooghadam[15] 및 제안한 프로토콜을 등록, 로그인 & 인증 및 패스워드 변경 단계의 연산량을 비교하고 있다. 해시, XOR연산, 암호화/복호화 연산 및 퍼지추출을 분석하여 최종적인 연산을 Table 3으로 나타낸다.

7. 결 론

원격 사용자 인증에서 사용자가 서버에 접속하여 서버가 제공하는 서비스를 안전하게 이용하기 위해서는 권위 있는 내부자 공격에 안전하고 순방향 안전성을 제공하는 프로토콜이 요구된다.

본 논문은 원격 사용자 인증에서 사용자의 패스워드를 보호하며 순방향 안전성을 제공하는 인증 프로토콜을 제안하였다. 제안한 방식은 Nikooghadam [15] 등의 프로토콜의 취약점인 권위 있는 내부자 공격, DoS 공격 및 순방향 안전성의 문제점을 개선하였다. 권위 있는 내부자 공격에 대응하기 위해서 가상의 아이디와 패스워드를 만들어 사용하고 fuzzy 연산을 사용하여 생체정보를 암호화하였다. Nikooghadam 등의 로그인 단계에서는 올바른 유저가 접속했는지 검증하는 단계가 없으므로 DoS 공격에 취약하지만 제안한 프로토콜에서는 로그인 단계에서 사용자가 스마트카드를 넣으면 아이디와 패스워드 및 생체정보를 확인하는 검증 절차를 거치게 된다. 또한 순방향 안전성을 제공하기 위해서 기존의 마스터 키 x로 암호화 및 복호화 하지 않고 마스터 키 x를 이용하여 J_i와 AID_i를 계산하여 암호화 및 복호화 하는 방식을 사용하였다. 따라서 제안한 방식은 Nikooghadam 등의 취약점을 보완하고 XOR 연산, 해시 연산 및 암호화 및 복호화 연산을 사용하여 효율적인 프로토콜을 제안하였다.

REFERENCES

[1] H.M. Sun, “An Efficient Remote Use Authentication Scheme Using Smart Cards,” *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, pp. 958-961, 2000.

[2] B.L. Chen, W.C. Kuo, and L.C. Wuu, “Robust Smart-Card-Based Remote User Password Authentication Scheme,” *International Journal of Communication Systems*, Vol. 27, No. 2, pp. 377-389, 2014.

[3] S.Y. Lee, K.S. Park, Y.H. Park, and Y.H. Park, “Symmetric Key-Based Remote User Authentication Scheme with Forward Secrecy,” *Journal of Korea Multimedia Society*, Vol. 19, No. 3, pp. 585-594, 2016.

[4] Y.F. Chang, W.L. Tai, and H.C. Chang, “Un-

traceable Dynamic-Identity-Based Remote User Authentication Scheme with Verifiable Password Update," *International Journal of Communication Systems*, Vol. 27, No. 11, pp. 3430-3440, 2014.

[5] G. Yang, D.S. Wong, H. Wang, and X. Deng, "Two-Factor Mutual Authentication Based on Smart Cards and Passwords," *Journal of Computer and System Sciences*, Vol. 74, No. 7, pp. 1060-1172, 2008.

[6] Q. Jiang, J. Ma, X. Lu, and Y. Tian, "An Efficient Two-Factor User Authentication Scheme with Unlinkability for Wireless Sensor Networks," *Peer-to-Peer Networking and Applications*, Vol. 8, No. 6, pp. 1070-1081, 2015.

[7] H. Arshad and M. Nikooghadam, "Three-Factor Anonymous Authentication and Key Agreement Scheme for Telecare Medicine Information Systems," *Journal of Medical Systems*, Vol. 38, No. 12, pp. 1-12, 2014.

[8] A.K. Das, "A Secure and Robust Temporal Credential-Based Three-Factor User Authentication Scheme for Wireless Sensor Networks," *Peer-to-Peer Networking and Applications*, Vol. 9, No. 1, pp. 223-244, 2016.

[9] A.T.B. Jin, D.N.C. Ling, and A. Goh, "Biohashing: Two Factor Authentication Featuring Fingerprint Data and Tokenised Random Number," *Pattern Recognition*, Vol. 37, No. 11, pp. 2245-2255, 2004.

[10] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys form Biometrics and Other Noisy Data," *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques*, pp. 523-540, 2004.

[11] X. Boyen, "Reusable Cryptographic Fuzzy Extractors," *Proceedings of the 11th ACM Conference on Computer and Communications Security*, pp. 82-91, 2004.

[12] D. Wang, P. Wang, C.G. Ma, and Z. Chen, "Robust Smart Card Based Password Authentication Scheme against Smart Card Security Breach," *Cryptology Eprint Archive,* pp. 1-35, 2012.

[13] S. Kumari, M.K. Khan, and X. Li, "An Improved Remote User Authentication Scheme with Key Agreement," *Computers & Electrical Engineering*, Vol. 40, No. 6, pp. 1997-2012, 2014.

[14] S.A. Chaudhry, M.S. Farash, H. Naqvi, S. Kumari, and M.K. Khan, "An Enhanced Privacy Preserving Remote User Authentication Scheme with Provable Security," *Security and Communication Networks*, Vol. 8, No. 18, pp. 3782-3795, 2015.

[15] M. Nikooghadam, R. Jahantigh, and H. Arshad, "A Lightweight Authentication and Key Agreement Protocol Preserving User Anonymity," *Multimedia Tolls and Applications*, pp. 1-23, 2016.

이 성 엽

2015년 2월 대구한의대학교 IT콘텐츠학과 학사
2015년 3월～2017년 2월 경북대학교 대학원 전자공학부 석사
관심분야 : 정보보호, 무선통신보안, 네트워크보안

박 요 한

2006년 2월 경북대학교 전자전기컴퓨터 학부 학사
2008년 2월 경북대학교 전자공학과 석사
2008년 3월～2013년 2월: 경북대학교 전자전기컴퓨터학부 박사
2013년 8월～2014년 7월 National University of Singapore 박사후연구원
2014년 9월～2016년 2월 경북대학교 산업전자공학과 시간강사
2016년 3월～2017년 2월 경북대학교 전자공학부 박사후연구원
2017년 3월～현재 나사렛대학교 IT학부 조교수
관심분야 : 정보보호, 무선통신보안, 네트워크보안

박 영 호

1989년 2월 경북대학교 전자공학과 학사
1991년 2월 경북대학교 전자공학과 석사
1995년 2월 경북대학교 전자공학과 박사
1996년～2008년 상주대학교 전자전기공학부 교수
2003년～2004년 Oregon State Univ. 방문교수
2008년～2014년 경북대학교 산업전자공학과 교수
2014년～현재 경북대학교 전자공학부 교수
관심분야 : 정보보호, 네트워크보안, 모바일 컴퓨팅