Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.2, Apr. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.2.329

다중 서버 환경에서 안전성이 향상된 스마트카드 기반 인증 프로토콜 설계에 관한 연구*

배 원 일,[1†] 곽 진[2‡]
[1]아주대학교 컴퓨터공학과 정보보호응용및보증연구실, [2]아주대학교 사이버보안학과

A Study on the Smartcard-Based Authentication Protocol Design with Advanced Security in the Multiple Server Environments*

Won-il Bae,[1†] Jin Kwak[2‡]
[1]ISAA Lab., Department of Computer Engineering, Ajou University,
[2]Department of Cyber Security, Ajou University

요 약

컴퓨터 네트워크 및 서비스 제공 서버 등의 급속한 성장으로 인해 자원의 효율성을 높이기 위한 다중 서버 아키텍처가 제안되었다. 이러한 다중 서버 환경의 스마트카드 기반 인증 프로토콜은 다양한 연구를 통해 지속적으로 발전되어 왔다. 최근 Chun-Ta Li 등은 Xiong Li 등의 인증 프로토콜이 사용자 가장 공격, 세션키 유출 공격에 취약하다는 문제점을 제기하고 취약점을 해결한 인증 프로토콜을 제안하였으나, Chun-Ta Li 등이 제안한 인증 프로토콜은 취약점 분석에 있어서 사용자 가장 공격 등에 문제가 있고 부적합한 인증 절차 문제를 가진다. 따라서 본 논문에서는 Xiong Li 등이 제안한 인증 프로토콜의 서비스 거부 공격 및 재전송 공격 등의 취약점을 해결하고 안전성이 향상된 다중 서버 환경의 스마트카드 기반 인증 프로토콜을 제안하고자 한다.

ABSTRACT

A multi-server architecture has been proposed to increase the efficiency of resources due to the rapid growth of computer networks and service providing servers. The smartcard-based authentication protocol in the multi-server environments has been continuously developed through various studies.

Recently, Chun-Ta Li et al proposed an authentication protocol that solves Xiong Li el al's authentication protocol vulnerability to user impersonation attack and session key disclosure attack. However, Chun-Ta Li et al's authentication protocol has a problem with user impersonation in the vulnerability analysis and has an unsuitable authentication process.

Therefore, this paper proposes a smartcard-based authentication protocol in the multi-server environments that solves the denial of service attack and replay attack vulnerabilities of the authentication protocol proposed by Xiong Li et al.

Keywords: Multi-Server Architecture, Smart card, Authentication protocol

I. 서 론

최근 컴퓨터 네트워크와 서비스 제공 서버 등의 발전으로 인해 하나의 서버에 대한 자원의 제약성을 높이기 위한 방안으로 다중 서버 환경이 제안되었다.

다중 서버 환경에서는 서비스를 제공받고자 하는

Received(02. 20. 2017), Modified(03. 20. 2017), Accepted(03. 21. 2017)
* 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2014R1A2A1A11050818).
† 주저자, wibae.isaa@gmail.com
‡ 교신저자, security@ajou.ac.kr(Corresponding author)

사용자가 원격으로 서버에 접속하기 때문에 데이터 도청 및 위변조 공격에 노출될 수 있고, 이를 방지하기 위한 안전한 인증 기술이 필요하다[1].

사용자가 스마트카드를 사용하여 다중 서버에서 제공하는 서비스를 이용할 수 있는 인증 프로토콜은 연산량을 줄이기 위하여 일방향 함수와 베타적 논리합을 이용한 연산으로 이루어져 있으며 이러한 인증 프로토콜은 지속적으로 연구가 진행되고 있다[2,3,4,5,6].

최근, Liao 등[7]은 사용자 익명성을 제공하기 위하여 다중 서버 환경의 인증 프로토콜을 제안하였다. 하지만 Hsiang 등[8]은 Liao 등[7]의 인증 프로토콜이 서버 스푸핑 공격과 내부자 공격에 취약하다는 점을 지적하였고, 이를 개선한 인증 프로토콜을 제안하였다. 그러나 Sood 등[9]과 Lee 등[10]은 Hsiang 등[8]의 인증 프로토콜이 여전히 서버 스푸핑 공격에 취약하고 검증자 유출 공격, 도난된 스마트카드 공격에 취약함을 지적하였으며, 이에 대한 안전성을 향상시킨 인증 프로토콜을 제안하였다.

이후, Xiong Li[11] 등은 Sood 등[9]의 인증 프로토콜이 검증자 유출 공격, 도난된 스마트 공격, 인증 단계의 부적합성의 문제점을 제기하고 새로운 인증 프로토콜을 제안하였으나, Chun-Ta Li 등[12]은 Xiong Li 등[11]의 논문이 사용자 위장 공격, 세션키 유출 공격, 서비스 거부 공격에 취약하다고 지적하여 개선된 인증 프로토콜을 제안하였다.

그러나, Chun-Ta Li 등[12]이 제안한 인증 프로토콜은 취약점 분석에 있어서 사용자 가장 공격 등에 문제가 있고 부적합한 인증 단계를 가지고 있다. 또한, 사용자가 스마트카드에 대한 읽기 및 쓰기 권한을 가지고 있어서, 공격자는 스마트카드 읽기 및 쓰기 권한을 이용하여 스마트카드 위변조가 가능하다는 문제점이 발생할 수 있다.

따라서 본 논문에서는 Chun-Ta Li 등[12]이 제안한 인증 프로토콜의 문제점을 해결하면서 Xiong Li 등[11]의 인증 프로토콜의 취약점을 모두 해결하는 안전성이 향상된 스마트카드 기반의 인증 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 Xiong Li 등[11]과 Chun-Ta Li 등[12]의 인증 프로토콜 및 취약점을 분석한다. 3장에서는 Xiong Li등[11]의 프로토콜을 개선하고 취약성을 보완한다. 4장에서는 개선된 프로토콜의 안전성 및 효율성을 분석하고, 마지막으로 5장에서 결론을 맺는다.

II. 관련 연구

2.1 Xiong Li 등의 인증 프로토콜

본 절에서는 Xiong Li 등[11]의 스마트카드를 이용한 동적 ID 기반 인증 프로토콜을 분석한다.

Xiong Li 등[11]의 인증 프로토콜은 사용자와 서비스 제공 서버, 인증 서버로 구성되어 있으며 사용자가 서비스 제공 서버에 로그인을 수행하는 것으로 사용자, 서비스 제공 서버, 인증 서버 간의 인증이 수행된다.

Xiong Li 등[11]의 인증 프로토콜에 사용된 파라미터들은 Table 1. 과 같다.

Xiong Li 등[11]의 인증 프로토콜은 등록 단계, 로그인 단계, 인증 및 키 동의 단계의 3단계로 구성

Table 1. Notations used in the Xiong Li et al's Protocol

Notations	Description
U_i	The ith user
S_j	The jth server
CS	The control server
ID_i	The identity of U_i
P_i	The password of U_i
SID_j	The identity of S_j
x	The master secret key chosen by CS
y	A secret number chosen by CS
b	A random number chosen by U_i for registration
CID_i	The dynamic identity generated by U_i for authentication
N_{i1}	A random number generated by U_i's smart card for session key agreement
N_{i2}	A random number generated by S_j for session key agreement
N_{i3}	A random number generated by CS for session key agreement
SK	A common session key shared among U_i, S_j and CS
$h(*)$	A collision free one-way hash function
$\oplus$	Exclusive OR operation
$\|$	Message concatenation operation

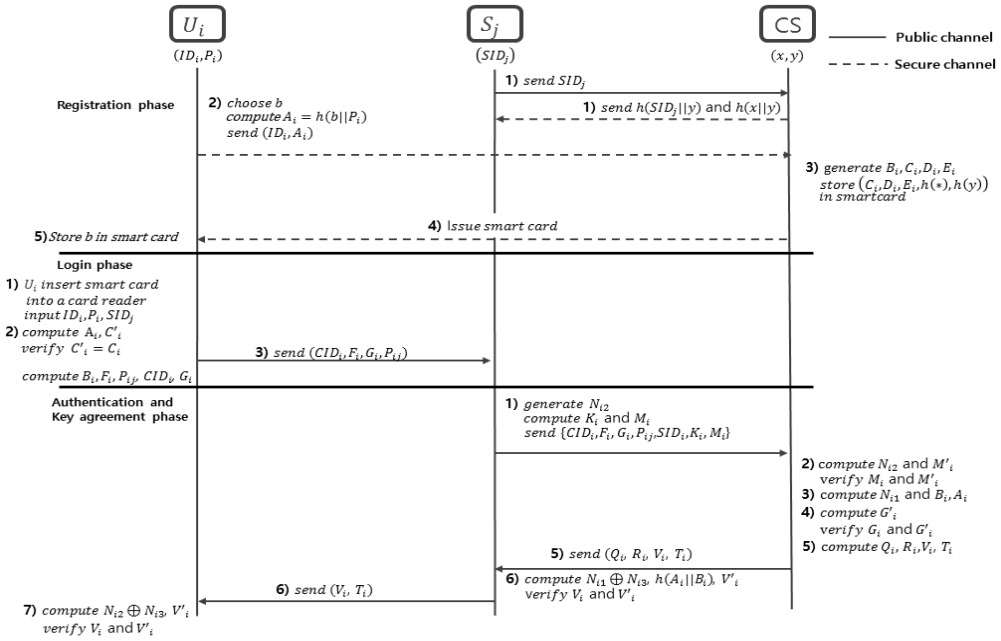

Fig. 1. Xiong Li et al's authentication protocol

되어 있으며 Fig.1.과 같다. CS는 신뢰할 수 있는 인증 서버로써 U_i와 S_j의 등록 및 인증을 담당한다.

2.1.1 등록 단계

등록 단계에서는 U_i, S_j가 CS에게 등록을 수행하고 CS는 U_i에게 스마트카드를 발행한다.

- ***Step 1.*** S_j는 CS에게 자신의 ID 값인 SID_j를 보안 채널로 전송하고 CS는 다음과 같은 식을 계산하여 S_j에게 보안 채널을 통해 전송한다.

$$h(SID_j \| y) \quad (1)$$

$$h(x \| y) \quad (2)$$

- ***Step 2.*** U_i는 사용자의 ID_i, P_i와 사용자가 선택한 b값을 선택하고 다음과 같은 식을 계산한 후, (ID_i, A_i)를 보안 채널을 통해 CS에게 전송한다.

$$A_i = h(b \| P_i) \quad (3)$$

- ***Step 3.*** CS는 B_i를 계산하고 스마트카드 발행을 위한 C_i, D_i, E_i를 계산하고 $(C_i, D_i, E_i, h(*), h(y))$를 U_i의 스마트카드에 저장한다.

$$B_i = h(ID_i \| x) \quad (4)$$

$$C_i = h(ID_i \| h(y) \| A_i) \quad (5)$$

$$D_i = B_i \oplus h(ID_i \| A_i) \quad (6)$$

$$E_i = B_i \oplus h(y \| x) \quad (7)$$

- ***Step 4.*** CS는 자신이 생성한 스마트카드 $(C_i, D_i, E_i, h(*), h(y))$를 U_i에게 보안 채널을 통해 발행한다.
- ***Step 5.*** U_i는 자신의 랜덤값 b를 스마트카드에 포함한 후에 $(C_i, D_i, E_i, h(*), h(y), b)$를 생성한다.

2.1.2 로그인 단계

로그인 단계에서는 사용자 U_i가 로그인을 수행하기 위하여 서비스 제공 서버 S_j에게 로그인 요청 메

시지를 전송하는 과정이다.

- ***Step 1.*** U_i는 자신의 스마트카드를 카드 리더기에 넣고 자신의 아이디 ID_i와 패스워드 P_i, 서비스 제공 서버 S_j의 아이디 SID_j를 함께 삽입한다.
- ***Step 2.*** 스마트카드는 $A_i = h(b \| P_i)$를 생성하고 A_i값을 이용하여 C'_i를 계산한다. 이후, 다음 식과 같이 스마트카드에 포함되어 있는 C_i와 스마트카드가 계산한 C'_i를 비교하여 일치하면 스마트카드의 정당한 소유자로 판단한다.

$$C'_i = h(ID_i \| h(y) \| A_i) \tag{8}$$

$$C_i = ? C_i' \tag{9}$$

- ***Step 3.*** 스마트카드의 정당한 소유자를 위한 검증이 끝나면 스마트카드는 랜덤값 N_{i1}를 선택한 후, 다음과 같이 B_i, F_i, G_i, P_{ij}, CID_i를 계산한 후 로그인 값 $(F_i, G_i, P_{ij}, CID_i)$를 S_j에게 공개된 채널을 통해 전송한다.

$$B_i = D_i \oplus h(ID_i \| A_i) \tag{10}$$

$$F_i = h(y) \oplus N_{i1} \tag{11}$$

$$CID_i = A_i \oplus h(B_i \| F_i \| N_{i1}) \tag{12}$$

$$P_{ij} = E_i \oplus h(h(y) \| N_{i1} \| SID_j) \tag{13}$$

$$G_i = h(B_i \| A_i \| N_{i1}) \tag{14}$$

2.1.3 인증 및 키 동의 단계

인증 및 키 동의 단계에서는 수신된 로그인 요청 메시지를 이용하여 사용자 U_i, 서비스 제공 서버 S_j, 인증 서버 CS 간의 검증을 수행한다.

- ***Step 1.*** S_j는 수신된 로그인 메시지를 이용하여 다음과 같은 K_i와 M_i를 계산한다. 이후, 로그인 요청메시지 $(F_i, G_i, P_{ij}, CID_i, SID_j, K_i, M_i)$를 생성하여 CS에게 전송한다. N_{i2}는 S_j가 선택한 랜덤값이다.

$$K_i = h(SID_j \| y) \oplus N_{i2} \tag{15}$$

$$M_i = h(h(x \| y) \| N_{i2}) \tag{16}$$

- ***Step 2.*** CS는 S_j에게 로그인 요청 메시지를 수신하고 N_{i2}를 계산한다. 계산된 N_{i2}값을 이용하여 M'_i를 계산하고 다음과 같이 수신된 로그인 요청 메시지의 M_i과 CS가 계산한 M'_i를 비교하여 일치하면 S_j는 CS에 의해 인증된다. 만약 일치하지 않으면 세션을 종료한다.

$$N_{i2} = K_i \oplus h(SID_j \| y) \tag{17}$$

$$M'_i = h(h(x \| y) \| N_{i2}) \tag{18}$$

$$M_i = ? M_i' \tag{19}$$

- ***Step 3.*** CS는 다음과 같이 N_{i1}와 B_i를 계산하고 계산된 B_i와 N_{i1}값을 이용하여 다음과 같은 식의 A_i를 생성한다.

$$N_{i1} = F_i \oplus h(y) \tag{20}$$

$$\begin{aligned} B_i &= P_{ij} \oplus h(h(y) \| N_{i1} \| SID_j) \oplus h(y \| x) \\ &= E \oplus h(y \| x) \end{aligned} \tag{21}$$

$$A_i = CID_i \oplus h(B_i \| F_i \| N_{i1}) \tag{22}$$

- ***Step 4.*** CS는 Step 3.에서 계산한 B_i, A_i, N_{i1} 값을 이용하여 다음과 같은 G'_i를 계산하고 로그인 요청 메시지의 G_i와 CS가 계산한 G'_i를 비교하여 일치하면 사용자 U_i는 CS에게 정당한 사용자로 인증된다.

$$G'_i = h(B_i \| A_i \| N_{i1}) \tag{23}$$

$$G_i = ? G_i' \tag{24}$$

- ***Step 5.*** CS는 랜덤값 N_{i3}을 생성하고 Q_i, R_i, V_i, T_i를 계산한다. CS는 계산된 상호 인증 메시지 (Q_i, R_i, V_i, T_i)를 S_j에게 전송한다.

$$Q_i = N_{i1} \oplus N_{i3} \oplus h(SID_j \| N_{i2}) \quad (25)$$

$$R_i = h(A_i \| B_i) \oplus h(N_{i1} \oplus N_{i2} \oplus N_{i3}) \quad (26)$$

$$V_i = h(h(A_i \| B_i) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3})) \quad (27)$$

$$T_i = N_{i2} \oplus N_{i3} \oplus h(A_i \| B_i \| N_{i1}) \quad (28)$$

- ***Step 6.*** S_j는 CS로부터 상호 인증 메시지를 수신하고 다음과 같은 $N_{i1} \oplus N_{i3}$, $h(A_i \| B_i)$, V'_i를 계산하고 상호 인증 메시지의 V_i값과 S_j가 생성한 V'_i를 비교하여 일치하면 CS는 S_j에게 정당한 인증 서버로 검증된다. S_j는 (V_i, T_i)를 U_i에게 전송한다.

$$N_{i1} \oplus N_{i3} = Q_i \oplus h(SID_j \| N_{i2}) \quad (29)$$

$$h(A_i \| B_i) = R_i \oplus h(N_{i1} \oplus N_{i3} \oplus N_{i2}) \quad (30)$$

$$V'_i = h(h(A_i \| B_i) \| h(N_{i1} \oplus N_{i3} \oplus N_{i2})) \quad (31)$$

$$V_i = ? \, V_i' \quad (32)$$

- ***Step 7.*** U_i는 S_j로부터 (V_i, T_i) 메시지를 수신하고 $N_{i2} \oplus N_{i3}$, V'_i를 계산한다. 이후, 다음과 같이 수신된 V_i와 V'_i를 비교하여 일치하면 CS와 S_j는 정당한 서비스 제공 서버로 검증된다. 따라서 U_i, S_j, CS는 공통 세션키인 $SK = h(h(A_i \| B_i) \| (N_{i1} \oplus N_{i2} \oplus N_{i3}))$를 생성할 수 있다.

$$N_{i2} \oplus N_{i3} = T_i \oplus h(A_i \| B_i \| N_{i1}) \quad (33)$$

$$V'_i = h(h(A_i \| B_i) \| h(N_{i2} \oplus N_{i3} \oplus N_{i1})) \quad (34)$$

$$V_i = ? \, V_i' \text{ (32번과동일)} \quad (35)$$

$$SK = h(h(A_i \| B_i) \| (N_{i1} \oplus N_{i2} \oplus N_{i3})) \quad (36)$$

2.2 Xiong Li 등의 인증 프로토콜 취약점 분석

Xiong Li 등[11]의 인증 프로토콜은 스마트카드 권한 문제, 서비스 거부 공격, 재전송 공격에 취약하며 관련된 분석 내용은 아래와 같다.

2.2.1 스마트 카드 권한 문제

등록 단계에서 사용자 U_i는 스마트카드에 대한 쓰기 권한을 가짐으로써 자신의 랜덤값 b를 스마트카드에 저장할 수 있다. 사용자에게 스마트카드 쓰기 권한을 부여하게 되면 인가된 사용자로 가장하는 공격자 또한 스마트카드에 대한 쓰기 권한을 가질 수 있다.

2.2.2 서비스 거부 공격

공격자 U_k가 S_j에게 로그인 요청 메시지를 전송할 때 많은 양의 동일한 로그인 요청 메시지를 전송하게 되면 서비스를 제공하는 서버 S_j의 서비스 가용성에 대한 문제가 발생할 수 있다.

$(n = 1,2,...,n.)$일 때, 공격자 U_{Kn}은 공격자가 선택한 랜덤값 N_{Kn}들을 생성한다. 이후, $F_{Kn} = N_{Kn} \oplus h(y)$, $P_{Kn} = E_i, \oplus h(h(y) \| N_{Kn} \| SID_j), CID_{Kn} = A_i \oplus h(B_i \| F_{Kn} \| N_{Kn}), G_{Kn} = h(B_i \| A_i \| N_{Kn})$을 계산하여 $(CID_{K1}, F_{K1}, G_{K1}, P_{K1}), (CID_{K2}, F_{K2}, G_{K2}, P_{K2}), ..., (CID_{Kn}, F_{Kn}, G_{Kn}, P_{Kn})$.의 로그인 요청 메시지를 전송한다. 즉, 공격자는 동일한 ID_K로 $N_{K1}, N_{K2}, ..., N_{Kn}$.을 생성하여 로그인을 수행할 수 있기 때문에 서비스 가용성 침해에 대한 문제가 발생할 수 있다.

2.2.3 재전송 공격

공격자는 이전 세션의 로그인 요청 메시지 $(F_i^{n-1}, G_i^{n-1}, CID_i, P_{ij}^{n-1}, K_i^{n-1}, M_i^{n-1}, SID_j)$를 안다고 가정했을 때, $N_{i2}^{n-1} = K_i^{n-1} \oplus h(SID_j \| y)$를 계산할 수 있다.

따라서 N_{i2}^{n-1}을 통해 $M'^{n-1}_i = h(h(x \| y) \| N_{i2}^{n-1})$을 계산할 수 있고, 생성된 M'^{n-1}_i은 로그인 요청 메시지의 M_i^{n-1}과 일치하기 때문에 S_j는 CS에 의해 인증될 수 있다. U_i에 대한 검증을 수행하기 위하여 $N_{i1}^{n-1} = F_i^{n-1} \oplus h(y)$을 통해 $B_i^{n-1} = P_{ij}^{n-1} \oplus h(h(y) \| N_{i1}^{n-1} \| SID_j) \oplus h(x \| y)$를 계산할 수 있으며 $A_i^{n-1} = CID_i \oplus h(B_i^{n-1} \| N_{i1}^{n-1} \| F_i^{n-1})$을 계산

하고 $G_i^{n-1} = h(B_i^{n-1} \| A_i^{n-1} \| N_{i1}^{n-1})$을 도출할 수 있다. 따라서 스마트카드의 G_i^{n-1}과 자신이 생성한 $G_i'^{n-1}$과 비교하여 일치하면 U_i는 CS에 의해 인증된다. 동일한 방법으로 공격자는 이전의 사용했던 로그인 응답 메시지 및 상호 인증 메시지를 이용하여 공통 세션키를 생성하고 U_i와 S_j, CS는 각 개체에 대한 인증을 수행할 수 있다.

2.3 Chun-Ta Li 등의 인증 프로토콜

Chun-Ta Li 등[12]의 인증 프로토콜은 기존의 Xiong Li 등[11]의 인증 프로토콜의 안전성의 문제를 제기하고 개선한 인증 프로토콜이다.

인증 방식은 Xiong Li 등의 인증 프로토콜과 동일하게 구성되어 있으며 본 절에서는 Chun-Ta Li 등[12]의 인증 프로토콜에 대한 부적합한 인증 절차와 취약점 분석에 있어서 사용자 가장 공격 등에 문제를 분석하기 위하여 인증 및 키 동의 단계의 일부분을 설명한다.

- ***Step 1.*** U_i는 로그인 요청 메시지 (TID_i, F_i, G_i)를 S_j에게 전송한다.
- ***Step 2.*** S_j는 S_j의 랜덤값 N_{i2}를 생성하고 자신의 로그인 요청 값인 K_i, M_i를 생성하고 $(TID_i, F_i, G_i, K_i, M_i)$를 CS에게 전송한다.
- ***Step 3.*** CS는 S_j의 랜덤값 N_{i2}을 계산하여 M'_i값을 도출한다. 이후, 로그인 요청 메시지의 M_i값과 자신이 계산한 M'_i을 비교하여 일치하면 S_j에 대한 검증이 이루어지게 된다.
- ***Step 4.*** CS는 검증 테이블에서 검증자 B_i를 검색하고 B_i와 로그인 요청메시지를 통해 N_{i1}을 도출할 수 있으며 이를 통해 G'_i를 계산한다. 이후, 자신이 계산한 G'_i와 로그인 요청 메시지의 G_i를 비교하여 일치하면 U_i는 정당한 사용자로 CS에게 검증된다.
- ***Step 5.*** CS는 로그인 요청 메시지에서 받은 TID_i 값을 TID'로 업데이트한다.

2.3.1 부적합한 인증

2.3절의 Step 2. 단계에서 S_j는 자신의 로그인 요청 값을 생성하고 $(TID_i, F_i, G_i, K_i, M_i)$를 CS에게 전송한다. 이후, CS는 N_{i2}을 계산하여 M'_i값을 도출함으로써 S_j에 대한 검증을 수행하지만 N_{i2}을 계산하는 과정에서 CS는 해당하는 서비스 제공 서버에 대한 j번째 SID를 알 수 없기 때문에 N_{i2} 값을 계산할 수 없고, 다중의 서비스 제공 서버 중 어떠한 서버가 인증을 요청하였는지 알 수 없다. 따라서 CS는 S_j에 대한 인증을 수행할 수 없다.

2.3.2 부적절한 취약점 분석

Chun-Ta Li 등[12]은 Xiong Li 등[11]의 인증 프로토콜이 사용자 가장 공격과 세션키 유출 공격에 취약하다고 문제를 제기하였으나 잘못된 방법으로 취약점을 도출하였다.

공격자 U_K가 사용자 U_i의 스마트카드에 전력 분석 공격을 통해 스마트카드 내부 정보를 획득하였다고 가정하였을 때, 공격자 U_K는 스마트카드 $(C_K, D_K, E_K, h(), h(y), b)$를 가질 수 있다.

공격자 U_i는 자신의 ID_K와 P_K를 삽입하여 $A_K = h(b \| P_K), h(ID_K \| A_K)$를 계산하고 $B_K = D_K \oplus h(ID_K \| A_K)$를 도출할 수 있으며 B_K값을 통해 $h(y \| x)$를 도출하면 사용자 가장 공격과 세션키 유출 공격이 가능하다고 주장하였다.

하지만, $B_K = D_K \oplus h(ID_K \| A_K)$에서 $h(ID_k \| A_k)$는 공격자의 ID_k, P_k로 계산된 값이지만 D_K는 도난된 사용자 U_i의 스마트카드에 포함된 D_i와 동일하다. 즉, 사용자가 생성한 값 D_i와 공격자가 생성한 값 $h(ID_k \| A_k)$을 통해 공격자의 B_K값을 도출할 수 없으므로 $h(x \| y)$값 또한 도출할 수 없다. 따라서 Xiong Li 등[11]의 인증 프로토콜은 사용자 가장 공격과 세션키 유출 공격에 대해 안전하다.

III. 제안하는 인증 프로토콜

본 논문에서 제안하는 인증 프로토콜은 사용자, 서비스를 제공하는 서버 및 인증 서버로 구성되어 있으며 사용자가 서비스 제공 서버에 로그인을 하는 것으로 각각에 대한 인증을 수행한다.

제안하는 인증 프로토콜에서 사용되는 파라미터 값들은 Table 2.와 같다.

Table 2. Notations used in the proposed Protocol

Notations	Description
U_i	The ith user
S_j	The jth server
CS	The control server
ID_i	The identity of U_i
P_i	The password of U_i
SID_j	The identity of S_j
x	The master secret key chosen by CS
y	A secret number chosen by CS
Ts	A time stamp
N_{i1}	A random number generated by U_i's smart card for session key agreement
N_{i2}	A random number generated by S_j for session key agreement
N_{i3}	A random number generated by CS for session key agreement
SK	A common session key shared among U_i, S_j and CS
$h(*)$	A collision free one-way hash function
$\oplus$	Exclusive OR operation
$\|$	Message concatenation operation

3.1 인증 프로토콜

제안하는 인증 프로토콜의 수행 단계는 3단계로 이루어져 있으며 등록 단계, 로그인 단계, 인증 및 키 동의 단계 순으로 Fig.2.와 같이 수행된다. CS는 신뢰할 수 있는 인증 서버로써 U_i와 S_j의 등록 및 인증을 담당한다. 제안하는 인증 프로토콜은 타임스탬프를 사용하기 때문에 U_i와 S_j 및 CS는 시간 동기화를 수행한다.

3.1.1 등록 단계

등록 단계에서는 U_i, S_j가 CS에게 등록을 수행하고 CS는 U_i에게 스마트카드를 발행한다.

- ***Step 1.*** S_j는 CS에게 자신의 ID 값인 SID_j를 보안 채널로 전송하고 CS는 S_j에게 다음과 같은 식을 계산하여 보안 채널을 통해 전송한다.

$$h(SID_j \| y) \tag{37}$$

$$h(x \| y) \tag{38}$$

- ***Step 2.*** U_i는 사용자의 아이디 ID_i, 패스워드 P_i를 선택하고 UI_i를 계산하여 CS에게 등록 요청 메시지 (ID_i, UI_i)를 전송한다.

$$UI_i = h(ID_i \| h(P_i)) \tag{39}$$

- ***Step 3.*** CS는 사용자 검증자 값 $UserVer_i$를 생성하고 스마트카드 검증을 위한 $VerSC_i$와 스마트카드에 포함되는 값 A_i를 생성한다.

$$UserVer_i = h(ID_i \| x) \tag{40}$$

$$VerforSC_i = h(ID_i \| h(y) \| UI_i) \tag{41}$$

$$A_i = UserVer_i \oplus UI_i \oplus h(x) \tag{42}$$

- ***Step 4.*** CS는 $(VerforSC_i, A_i, h(*), h(y))$를 스마트카드에 저장한다.
- ***Step 5.*** CS는 사용자 정보 값인 UI_i와 사용자 검증자 값 $UserVer_i$ 및 상태 비트 값을 검증자 테이블에 저장한다. 만약 사용자가 등록을 하면 상태 비트 값은 1로 저장되며 등록이 되어 있지 않으면 0으로 저장된다. 검증자 테이블은 Table 3.과 같다.

Table 3. The verifier table

User information	User-verifier	Status-bit
UI_i	$UserVer_i$	0/1
UI_j	$UserVer_j$	0/1

- ***Step 6.*** CS_i는 생성한 스마트카드를 U_i에게 발행한다.

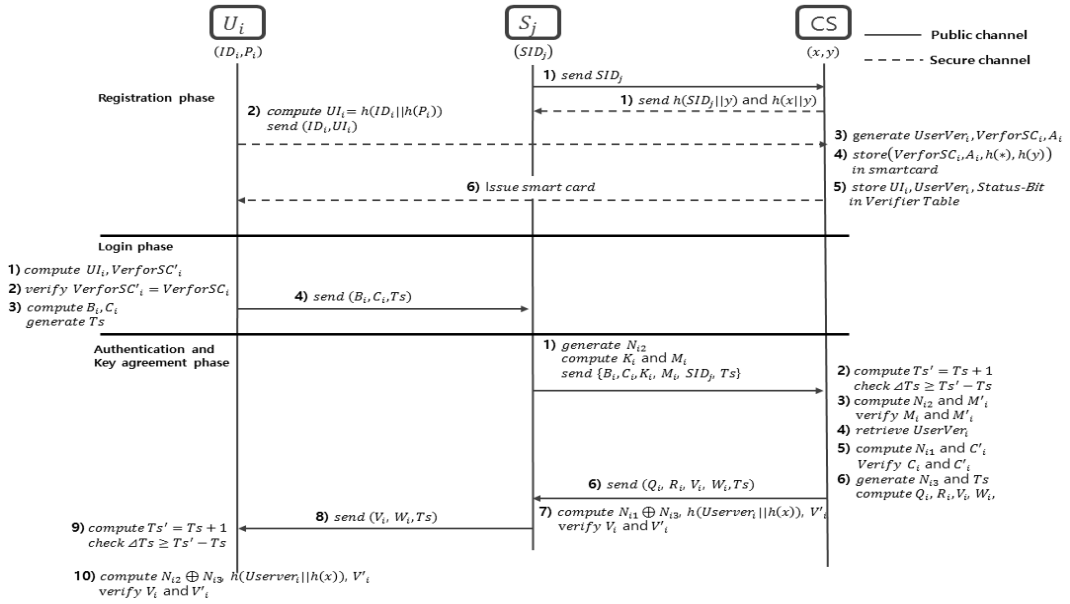

Fig. 2. The proposed authentication protocol

3.1.2 로그인 단계

로그인 단계에서는 사용자 U_i가 로그인을 수행하기 위하여 서비스 제공 서버 S_j에게 로그인 요청 메시지를 전송하는 과정이다.

- ***Step 1.*** U_i는 스마트카드를 카드 리더기에 삽입하고 자신의 ID_i, P_i를 입력한다. 스마트카드는 $UI_i = h(ID_i \| h(P_i))$를 계산하고 UI_i값을 이용하여 $VerforSC'_i = h(ID_i \| h(y) \| UI_i)$를 생성한다.

$$VerforSC'_i = h(ID_i \| h(y) \| UI_i) \tag{43}$$

- ***Step 2.*** 이후, 다음과 같이 자신이 생성한 $VerforSC'_i$와 스마트카드에 포함된 $VerforSC_i$를 비교하여 일치하면 정당한 스마트카드의 소유자로 검증된다.

$$VerforSC_i = ? \, VerforSC'_i \tag{44}$$

- ***Step 3.*** U_i는 랜덤값 N_{i1}을 선택하여 다음과 같이 B_i와 C_i를 계산하고 타임스탬프 Ts를 생성한다.

$$B_i = A_i \oplus UI_i \oplus N_{i1} \tag{45}$$

$$C_i = h(N_{i1} \| h(y)) \tag{46}$$

- ***Step 4.*** U_i는 S_j에게 로그인 요청 메시지 (B_i, C_i, Ts)를 전송한다.

3.1.3 인증 및 키 동의 단계

인증 및 키 동의 단계에서는 수신된 로그인 요청 메시지를 이용하여 사용자 U_i, 서비스 제공 서버 S_j, 인증 서버 CS 간의 검증을 수행한다.

- ***Step 1.*** S_j는 랜덤값 N_{i2}를 선택하고 다음과 같이 K_i와 M_i를 계산하여 CS에게 로그인 요청 메시지 $(B_i, C_i, K_i, M_i, SID_j, Ts)$를 전송한다.

$$K_i = h(SID_j \| y) \oplus N_{i2} \tag{47}$$

$$M_i = h(h(x) \| y \| N_{i2}) \tag{48}$$

- ***Step 2.*** CS는 $Ts' = Ts + 1$을 계산하여 $\triangle Ts \geqq Ts' - Ts$를 확인한다. Ts'는 서버가 로그인

메시지를 수신하였을 때의 타임스탬프이며, $\triangle Ts$는 로그인 메시지 전송 시간을 고려한 최소한의 인증 시간이다.

- **Step 3.** CS는 다음과 같이 N_{i2}의 계산을 통해 M'_i을 도출할 수 있으며, 로그인 요청 메시지의 M_i와 CS가 도출한 M'_i와 비교하여 일치하면 정당한 S_j로 인증된다.

$$N_{i2} = K_i \oplus h(SID_j \| y) \tag{49}$$

$$M'_i = h(h(x \| y) \| N_{i2}) \tag{50}$$

$$M_i = ? M'_i \tag{51}$$

- **Step 4.** CS는 UI_i에 일치하는 $UserVer_i$를 검증자 테이블에서 검색한다.
- **Step 5.** CS는 다음과 같이 N_{i1}를 계산하고 N_{i1}를 이용하여 C'_i를 도출할 수 있으며, 로그인 요청 메시지의 C_i와 CS가 도출한 C'_i를 계산하여 일치하면 정당한 U_i로 인증된다.

$$N_{i1} = B_i \oplus h(UserVer_i \| h(x)) \tag{52}$$

$$C'_i = h(h(y) \| N_{i1}) \tag{53}$$

$$C_i = ? C'_i \tag{54}$$

- **Step 6.** CS는 타임스탬프 Ts를 생성하고 랜덤값 N_{i3}을 선택한다. 이후, 다음과 같이 Q_i, R_i, V_i, W_i를 계산하여 상호 인증 메시지 (Q_i, R_i, V_i, W_i, Ts)를 S_j에게 전송한다.

$$Q_i = N_{i1} \oplus N_{i3} \oplus h(SID_j \oplus N_{i2}) \tag{55}$$

$$R_i = h(UserVer_i \| h(x)) \oplus h(SID_j \| N_{i2}) \tag{56}$$

$$V_i = h(h(UserVer_i \| h(x)) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3})) \tag{57}$$

$$W_i = N_{i2} \oplus N_{i3} \oplus h(UserVer_i \| h(x)) \tag{58}$$

- **Step 7.** S_j는 상호 인증 메시지를 수신하고 $N_{i1} \oplus N_{i3}$, $h(UserVer_i \| h(x))$을 계산하여 V'_i를 도출할 수 있으며 상호 인증 메시지의 V_i와 자신이 생성한 V_i'와 비교하여 일치하면 정당한 서버로 인증된다.

$$N_{i1} \oplus N_{i3} = Q_i \oplus h(SID_j \oplus N_{i2}) \tag{59}$$

$$h(UserVer_i \| h(x)) = R_i \oplus h(SID_j \| N_{i2}) \tag{60}$$

$$V'_i = h(h(UserVer_i \| h(x)) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3})) \tag{61}$$

$$V_i = ? V'_i \tag{62}$$

- **Step 8.** S_j는 응답 메시지 (V_i, W_i, Ts)를 사용자 U_i에게 전송한다.
- **Step 9.** S_j는 응답 메시지를 수신하고 타임스탬프 $Ts' = Ts + 1$을 계산하여 $\triangle Ts \geq Ts' - Ts$을 확인한다. Ts'는 서버가 로그인 메시지를 수신하였을 때의 타임스탬프이며, $\triangle Ts$는 로그인 메시지 전송 시간을 고려한 최소한의 인증 시간이다.
- **Step 10.** U_i는 $h(UserVer_i \| h(x))$, $N_{i2} \oplus N_{i3}$의 계산을 통해 V'_i를 도출할 수 있으며, 응답 메시지의 V_i와 U_i가 도출한 V'_i와 비교하여 일치하면 U_i에 의해 정당한 S_j와 CS로 인증된다. 따라서 U_i, S_j, CS는 다음과 같이 동일한 세션키 SK을 생성함으로써 인증을 수행할 수 있다.

$$h(UserVer_i \| h(x)) = A_i \oplus UI_i \tag{63}$$

$$N_{i2} \oplus N_{i3} = W_i \oplus h(UserVer_i \| h(x)) \tag{64}$$

$$V'_i = h(h(UserVer_i \| h(x)) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3})) \tag{65}$$

$$V_i = ? V'_i \text{(62번과동일)} \tag{66}$$

$$SK = h(h(UserVer_i \| h(x)) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3})) \tag{67}$$

3.1.4 패스워드 변경 단계

패스워드 변경 단계에서는 사용자가 자신의 패스워드를 새로운 패스워드로 변경하고자 할 때 수행하는 과정으로 5가지 단계로 구성되어 있다.

- ***Step 1.*** U_i는 자신의 패스워드를 변경하기 위해 스마트카드를 카드 리더기에 삽입하고 자신의 ID_i, P_i를 입력한다.

- ***Step 2.*** 스마트카드는 $UI_i = h(ID_i \| h(P_i))$를 계산하고 UI_i값을 이용하여 $VerforSC'_i = h(ID_i \| h(y) \| UI_i)$를 생성한다.

$$VerforSC'_i = h(ID_i \| h(y) \| UI_i) \quad (68)$$

- ***Step 3.*** 이후, 다음과 같이 자신이 생성한 $VerforSC'_i$와 스마트카드에 포함된 $VerforSC_i$를 비교하여 일치하면 정당한 스마트카드의 소유자로 검증된다.

$$VerforSC_i = ? \, VerforSC'_i \quad (69)$$

- ***Step 4.*** 정당한 스마트카드의 소유자로 검증되면, 사용자는 변경하고자 하는 새로운 패스워드 NwP_i를 삽입하고 스마트카드는 새로운 $NwUI_i = h(ID_i \| h(NwP_i))$를 생성한다.

$$NwUI_i = h(ID_i \| h(NwP_i)) \quad (70)$$

- ***Step 5.*** 생성된 $NwUI_i$를 이용하여 새로운 $NwVerforSC_i$를 계산할 수 있으며 기존의 $VerforSC_i$와 대체되어 스마트카드에 저장함으로써 패스워드 변경이 이루어진다.

$$NwVerforSC_i = h(ID_i \| h(y) \| NwUI_i) \quad (71)$$

IV. 안전성 및 효율성 분석

4.1 안전성 분석

4.1.1 스마트카드 권한 문제

제안하는 인증 프로토콜에서는 인증 서버인 CS가 스마트카드에 대한 쓰기 권한을 가지고 있으며 사용자가 스마트카드에 대한 쓰기 권한을 가지고 있지 않다. 따라서 인가된 사용자로 가장하는 공격자에 의한 스마트카드에 대한 위변조를 방지할 수 있다.

4.1.2 사용자 가장 공격

공격자가 이전 세션의 공개된 채널을 통해 로그인 요청 메시지 (B_i, C_i, Ts)를 안다고 가정할 때 악의적인 공격자는 랜덤값 N_{i1}과 $A_i, UI_i, h(y)$를 계산할 수 없기 때문에 사용자로 가장하여 현재 세션의 로그인 요청 메시지 B_i, C_i, Ts를 도출할 수 없다. 따라서 제안하는 인증 프로토콜은 사용자 가장 공격에 안전하다.

4.1.3 세션키 유출 공격

공격자가 이전 세션의 공개된 채널을 통해 $B_i, C_i, K_i, M_i, SID_j, Ts$와 Q_i, R_i, V_i, W_i를 안다고 가정할 때, 공격자는 세션키를 도출할 수 없다.

공격자는 공개된 B_i를 통해 N_{i1}를 도출할 수 없으며, 공개된 W_i를 통해 $N_{i2} \oplus N_{i3}$을 도출할 수 없으므로, 공격자는 공통 세션키인 $h(h(UserVer_i \| h(x)) \| h(N_{i1} \oplus N_{i2} \oplus N_{i3}))$을 생성할 수 없다. 따라서 제안하는 인증 프로토콜은 세션키 유출 공격에 안전하다.

4.1.4 재전송 공격

제안하는 인증 프로토콜에서는 U_i,S_j와 CS간의 로그인 요청 메시지 및 인증 메시지에 대해 타임스탬프를 사용한다. 따라서 공격자가 전송되는 메시지에 대한 위변조 공격을 수행하기 위해 세션에 참여하였을 경우, 타임스탬프 Ts의 값이 변하기 때문에 재전송 공격에 대응할 수 있다.

4.1.5 서비스 거부 공격

공격자 U_k가 자신의 스마트카드와 ID_k, P_k, N_{i1}를 이용하여 많은 양의 동일한 로그인 요청 메시지를 생성하여 전송할 경우, 서비스를 제공하는 서버 S_j와 인증 서버 CS에 대한 서비스 가용성에 대한 문제가 발생할 수 있다.

하지만 제안하는 인증 프로토콜에서는 검증 테이블을 사용하여 상태 비트 값이 1로 저장되어 있다면 해당하는 로그인 요청 메시지를 받지 않기 때문에 서비스의 가용성을 침해하는 서비스 거부 공격에 안전하다.

Table 4. Vulnerability comparison of our protocol and other related protocols

Vulnerability	[11]	[12]	ours
Problems with smart card permissions	×	×	○
Impersonation attack	○	○	○
Session key disclosure attack	○	○	○
Replay attack	×	×	○
Many logged-in user's attack	×	○	○
Authentication suitability	○	×	○

4.2 효율성 분석

본 절에서는 Xiong Li 등[11]의 인증 프로토콜과 본 논문에서 제안하는 인증 프로토콜의 효율성을 비교 분석하였으며 Table 5.와 같다.

Xiong Li 등[11]의 인증 프로토콜에서 로그인 단계와 인증 및 키 동의 단계의 연산량을 분석한 결과 28번의 XOR 연산과 33번의 해쉬 암호화 연산이 필요한 반면에 본 논문에서 제안하는 인증 프로토콜은 22번의 XOR 연산과 32번의 해쉬 암호화 연산이 필요하다.

따라서 제안하는 인증 프로토콜은 Xiong Li 등[11]의 인증 프로토콜의 연산량보다 적은 연산량으로 인증 프로토콜을 수행할 수 있기 때문에 지속적으로 경량화를 요구하는 스마트기반 환경에서 효율적으로 활용될 수 있다.

Table 5. Performance Comparisons of our protocol and Xiong Li et al's protocol

Phase	Entity	[11]	ours
Login phase	U_i	$4X8h$	$2X6h$
	S_j	-	-
	CS	-	-
Authentication and key agreement phase	U_i	$3X4h$	$4X4h$
	S_j	$7X8h$	$6X9h$
	CS	$14X13h$	$10X13h$

X : Exclusive OR h : Hash function

V. 결 론

다중 서버 환경에서 사용자는 원격으로 서버에 접속하여 서비스를 제공받는다. 하지만 악의적인 공격자에 의해 데이터 도청 및 위변조 공격에 노출될 수 있다. 따라서 서비스를 제공받고자 하는 사용자는 원격으로 서버에 접속하기 때문에 데이터 도청 및 위변조 공격에 노출될 수 있음으로 이를 방지하기 위한 안전한 인증 기술이 필요하다

Chun-Ta Li 등[12]의 인증 프로토콜은 Xoing Li등[11]의 인증 프로토콜에 대한 문제점을 제기하고 개선된 인증 프로토콜을 제안하였지만 Chun-Ta Li 등[12]은 잘못된 방식으로 취약점을 분석하였으며 부적합한 인증 절차를 가진다.

따라서 본 논문에서는 Xoing Li 등[11]의 인증 프로토콜에 대한 새로운 문제점을 제기하고 이에 대한 안전성 및 효율성이 향상된 인증 프로토콜을 제안하였다.

본 논문에서 제안한 인증 프로토콜은 스마트카드를 이용한 키 교환 및 다양한 응용 분야에서 활용될 수 있을 것으로 기대한다.

References

[1] Ashish Singh, Kakali Chatterjee, "An Efficient Three Factor Based Remote User Authentication Protocol for Distributed Networks" Computer Information

Systems and Industrial Management, Vol 9842, pp 682-693, Sept. 2016.

〔2〕 R. Abdellatif, H. K. Aslan and S. H. Elramly, "New real time multicast authentication protocol", International Journal of Network Security, Vol 12, pp. 13-20, 2011.

〔3〕 C. C. Chang, H. L. Wu, Z. H. Wang, and Q. Mao, "An efficient smart card based authentication scheme using image encryption", Journal of Information Science and Engineering, Vol 29, pp. 1135-1150, Nov. 2013.

〔4〕 E. El-Emam, M. Koutb, H. Kelash and O. S. Faragallah, "An authentication protocol based on Kerberos 5", International Journal of Network Security, Vol 12, pp. 159-170, May. 2011,

〔5〕 D. He, J. Chen and J. Hu, "Weaknesses of a remote user password authentication scheme using smart card", International Journal of Network Security, Vol 13, pp. 58-60, Feb. 2011.

〔6〕 M. S. Hwang, S. K. Chong and T. Y. Chen, "Dos-resistant ID-based password authentication scheme using smart cards", Journal of Systems and Software, Vol 83, pp. 163-172, Jan. 2010.

〔7〕 Y.P Liao, S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment" Computer Standards and Interfaces, Vol 31, pp. 24-29, Oct. 2009.

〔8〕 H. C. Hsiang, W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment", Computer Standards and Interfaces, Vol 31, pp. 1118-1123, Nov. 2009.

〔9〕 S. K. Sood, A. K. Sarje and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture", Journal of Network and Computer Applications, Vol 34, pp. 609-618, Mar. 2011.

〔10〕 C. C. Lee, T. H. Lin and R. X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards", Expert Systems with Applications, Vol 38, pp. 13863-13870, Oct. 2011.

〔11〕 X. Li, Y. Xiong, J. Ma, W. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards", Journal of Network and Computer Applications, Vol 35, pp. 763-769, Mar. 2012.

〔12〕 Chun-Ta Li, Cheng-Chi Lee, Chi-Yao Weng, Chun-I Fan, "A Secure Dynamic Identity Based Authentication Protocol with Smart Cards for Multi-Server Architecture", Journal of Information Science and Engineering 31, 1975-1992, Oct. 2014.

〈저자소개〉

배 원 일 (Won-il Bae) 학생회원
2016년 2월: 목원대학교 컴퓨터공학과 학사
2016년 3월~현재: 아주대학교 컴퓨터공학과 석사과정
〈관심분야〉 클라우드 컴퓨팅 보안, 암호프로토콜, 개인정보보호

곽 진 (Jin Kwak) 종신회원
2000년 8월: 성균관대학교 학사
2003년 2월: 성균관대학교 석사
2006년 2월: 성균관대학교 박사
2006년 4월~2006년 11월: 일본 큐슈대학교 방문연구원
2006년 8월~2006년 11월: 일본 큐슈시스템정보기술연구소 특별연구원
2006년 11월~2007년 2월: 정보통신부 정보보호기획단 개인정보보호팀 통신사무관
2007년 3월~2015년 2월: 순천향대학교 정보보호학과 교수
2008년 1월~현재: 한국정보보호학회 상임이사
2011년 1월~현재: 한국정보처리학회 이사
2015년 3월~현재: 아주대학교 사이버보안학과 교수
〈관심분야〉 자동차 보안, 암호프로토콜, 응용시스템보안, 클라우드 컴퓨팅 보안, 개인정보보호, 정보보호제품평가