# 재전송 공격에 안전한 개선된 강력한 패스워드 상호인증 프로토콜

# Improved Strong Password Mutual Authentication Protocol to Secure on Replay Attack

김준섭*, 곽진*

Jun-Sub Kim*, Jin Kwak*

## 요 약

개방형 네트워크에서 사용자 인증은 중요한 보안 기술이다. 특히, 패스워드 기반의 인증 방식은 분산된 환경에서 가장 널리 사용되고 있으며, 현재까지 많은 인증 방식들이 제안되고 있다. 그 중 하나인 SPMA 프로토콜은 NSPA 프로토콜에서 상호인증을 제공하지 않는 문제점으로 인하여 발생할 수 있는 취약성을 지적하며, 상호인증을 제공하는 강력한 패스워드 상호인증 프로토콜을 제안하였다. 하지만 SPMA 프로토콜은 재전송 공격에 대한 취약성을 가지고 있다. 따라서 본 논문에서는 SPMA 프로토콜의 재전송 공격에 대한 취약성을 분석하고, SPMA 프로토콜과 동일한 효율성을 제공하면서 재전송 공격에 안전한 개선된 강력한 패스워드 상호인증 프로토콜을 제안한다.

## Abstract

In public network, user authentication is important security technology. Especially, password-based authentication method is used the most widely in distributed environments, and there are many authentication methods. Their SPMA protocol indicates vulnerability about problem that NSPA protocol does not offer mutual authentication, and proposed Strong Password Mutual Authentication protocol with mutual authentication. However, SPMA protocol has vulnerability of replay attack. In the paper, we analyzed vulnerability to replay attack of SPMA protocol. And we also proposed Improved Strong Password Mutual Authentication protocol to secure on replay attack with same efficiency.

Key words : Password Protocol, Replay Attack, Mutual Authentication, Hash Function, Safety

## I. 서 론

사용자 인증은 개방형 네트워크상에서 안전한 통신을 보장하기 위한 중요한 보안 기술이다. 즉, 공격자가 시스템의 데이터를 읽고 클라이언트와 서버 간의 통신을 변경하거나 도청하더라도 인증 메시지에 대한 안전성과 기밀성을 보장하여 사용자 인증을 안전하게 수행할 수 있어야 한다.

1981년 Lamport [1] 이 일회용 패스워드 방식을 제안하였고, 이에 따라 여러 가지 패스워드 인증 프로

* 순천향대학교 정보보호학과(Dept. of Information Security Engineering., Soonchunhyang University)
· 제1저자 (First Author) : 김준섭 · 교신저자 (Corresponding Author) : 곽진
· 투고일자 : 2010년 5월 28일
· 심사(수정)일자 : 2010년 5월 일 (수정일자 : 2010년 6월 23일)
· 게재일자 : 2010년 6월 30일

토콜 등이 제안되고 있다 [2]-[11]. 하지만, 이 방식은 고도의 해시 오버헤드와 패스워드 재설정에 대한 어려움이 존재하고 있다. 이러한 문제를 해결하기 위해 Shimizu는 CINON(chained one-way data verification method) [12],[13] 방식을 제안하였다. CINON 방식은 난수를 메모리 장치(IC 카드) 등에 저장하여 안전성을 제공하지만, 이는 휴대의 불편함과 하드웨어의 고비용에 대한 문제점이 발생하였다. 이를 보완하기 위해 Shimizu는 난수를 저장하는 문제를 해결하기 위해 PERM(Privacy Enhanced information Reading and writing Management method) [14] 방식을 제안하였다.

2000년 Sandirigama 등은 CINON과 PERM이 중간자 공격에 성공할 수 있는 것을 지적하며, SAS(Simple And Secure) 패스워드 인증 프로토콜을 제안하였다 [15]. 하지만 Chen과 Ku는 SAS 패스워드 인증 프로토콜이 훔친 검증자 공격에 대해 취약하다는 것을 증명하였다 [16]. 또한 Lin 등은 SAS 패스워드 인증 프로토콜이 재전송 공격과 서비스 거부 공격에 대한 취약성을 지적하며, OSPA(Optimal Strong-Password Authentication) 프로토콜을 제안하였다 [17]. 그럼에도 불구하고 Chen과 Ku는 OSPA 프로토콜이 SAS 프로토콜과 마찬가지로 훔친 검증자 공격에 대해 취약하다는 것을 증명하였다 [16].

2003년 Lin 등은 OSPA 프로토콜의 안전성을 강화한 SE-OSPA(Security Enhancement for Optimal Strong-Password Authentication) 프로토콜을 제안하였다 [18]. 그러나 Ku 등은 SE-OSPA 프로토콜이 재전송 공격과 서비스 거부 공격에 대해 취약하다는 것을 증명하였다 [19]. 2006년 Lin 등은 SE-OSPA 프로토콜의 안전성과 효율성을 강화한 NSPA(New Strong-Password Authentication) 프로토콜을 제안하였다 [20]. 그러나 Yoon 등은 NSPA 프로토콜이 상호인증을 제공하지 않은 문제점으로 인해 서버로 위장한 공격과 서비스 공격에 대한 취약성을 지적하며, 상호인증을 제공하는 SPMA(Strong Password Mutual Authentication) 프로토콜을 제안하였다 [21].

하지만 SPMA 프로토콜은 공격자가 정당한 사용자로 가장하기 위해 이전의 인증 메시지를 도청하여 전송하는 재전송 공격에 대한 취약성이 존재한다. 즉, 공격자는 SPMA 프로토콜에서 서버가 수행하는 상호인증의 첫 번째 인증단계에서 재전송 공격에 대한 취약성을 이용하여 공격자를 정당한 사용자로 인증할 수 있는 문제를 발생시킨다. 그 후 서버는 상호인증을 위해 공격자에게 인증 메시지를 전송한다. 하지만 사용자 U로 가장한 공격자는 인증 메시지에 대한 무결성을 검증한 후 서버에게 인증 완료에 대한 메시지를 보내지 않기 때문에 정당한 사용자임을 입증하는 상호인증에 성공할 수 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 재전송 공격에 안전한 개선된 강력한 패스워드 상호인증(I-SPMA : Improved Strong Password Mutual Authentication) 프로토콜을 제안한다. I-SPMA 프로토콜은 재전송 공격, 위장 공격, 패스워드 추측 공격 등 다양한 공격에 대한 안전성을 제공하며, 기존의 SPMA 프로토콜과 동일한 효율성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 SPMA 프로토콜과 SPMA 프로토콜에 대한 재전송 공격을 분석하고, 3장에서는 개선된 강력한 패스워드 상호인증 프로토콜인 I-SPMA를 제안한다. 4장에서는 안전성을 분석하고 마지막으로 5장에서는 결론을 맺는다.

## Ⅱ. SPMA 프로토콜에 대한 재전송 공격 취약성

### 2-1 SPMA 프로토콜

**등록 단계** : 새로운 사용자 U가 서비스 접근을 위해 서버에 안전한 등록을 요청한다. 등록 단계는 안전한 채널을 통해 수행되며, 수행 절차는 다음과 같다.

1) 사용자 U는 클라이언트를 이용하여 PRNG(·)로부터 랜덤 넌스 N을 생성하고, 패스워드 검증자 $h(P \| N)$을 계산한다. 안전한 채널을 통하여 자신의 식별자 ID와 패스워드 검증자 $h(P \| N)$를 서버 S에게 전송한다.

2) 서버 S는 사용자 U의 식별자 ID가 기존에 가입

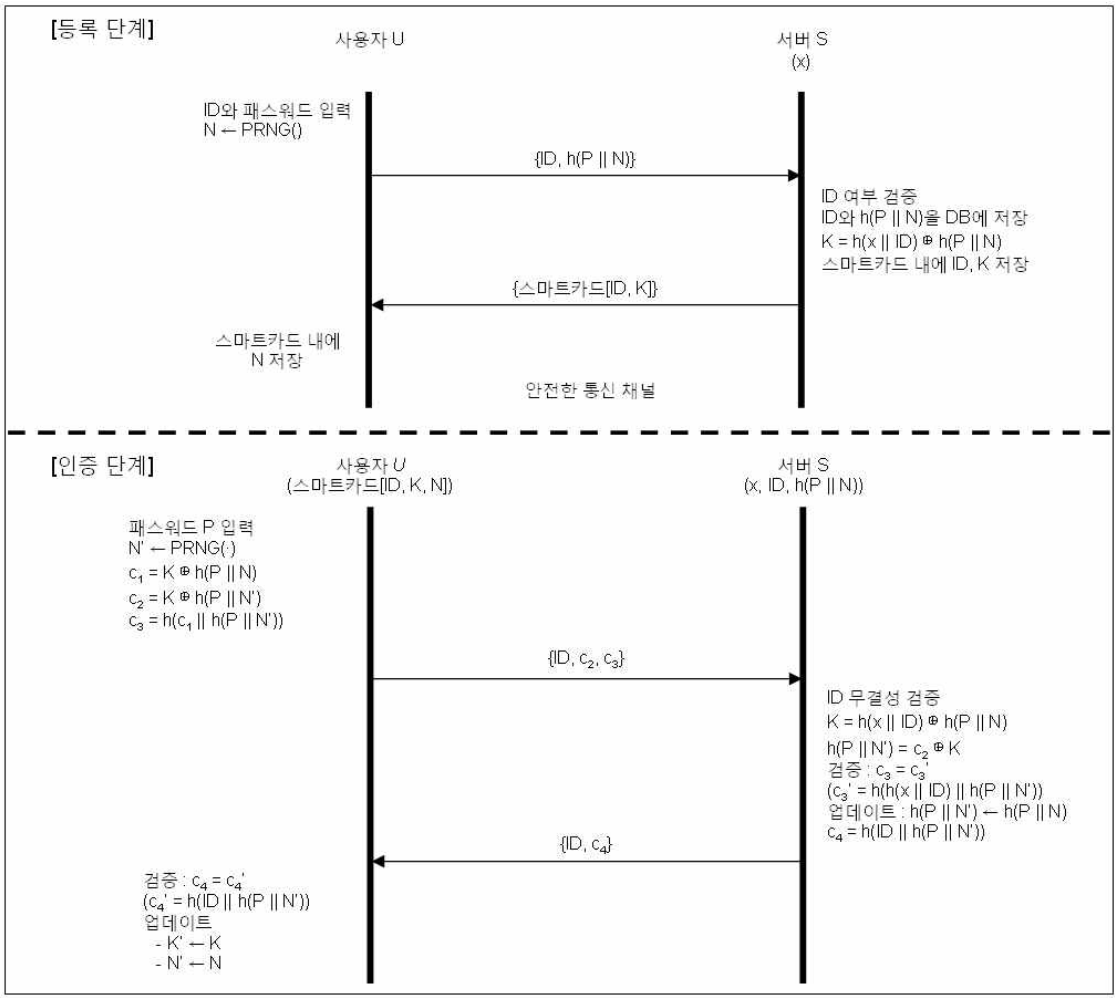


그림 1. SPMA 프로토콜의 등록 및 인증 단계
Fig 1. Registration & authentication phase of SPMA protocol.

된 사용자의 ID 인지 여부를 검증한다. 기존에 가입된 ID가 식별되지 않으면, 자신의 데이터베이스 내에 사용자 U의 식별자 ID와 패스워드 검증자 $h(P \| N)$를 저장한다. 이후 서버 S는 자신의 비밀키 x를 이용하여 $K = h(x \| ID) \oplus h(P \| N)$를 계산하여, 사용자 U를 위한 스마트카드에 ID와 K값을 저장하고, 안전한 채널을 통해 사용자 U에게 스마트카드를 발급한다.

3) 사용자 U는 발급받은 스마트카드와 자신의 클라이언트를 이용하여 랜덤 넌스 N을 스마트카드 내에 저장한다.

인증 단계 : 인증 단계는 사용자 U가 서버 S에 접속하기 위해 스마트카드와 패스워드를 입력을 하여 로그인을 요청한다. 수행 절차는 다음과 같다.

1) 사용자 U는 로그인 요청 메시지(challenge) 값인 메시지 $\{ID, c_2, c_3\}$을 서버 S에게 전송한다. $c_2$, $c_3$은 다음과 같이 계산된다.

$$c_1 = K \oplus h(P \| N) = h(x \| ID) \quad (1)$$

$$c_2 = K \oplus h(P \| N') \quad (2)$$

$$c_3 = h(c_1 \| h(P \| N')) \quad (3)$$

2) 서버 S는 사용자 U로부터 전송받은 로그인 요청 메시지 $\{ID, c_2, c_3\}$을 이용해 $c_2 \oplus K$ 연산을 수행하여 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 계산한다. $h(x \| ID)$와 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 이용하여 $c_3$의 무결성을 검증한다. 무결성이 검증되면 서버 S는 사용자 U를 인증하게

되고, 기존의 패스워드 검증자 $h(P \| N)$를 다음 세션을 위한 패스워드 검증자 $h(P \| N')$으로 업데이트한다. 서버 S는 사용자 U와 상호인증을 수행하기 위해 $c_4 = h(ID \| h(P \| N'))$을 계산한 후, 사용자 U에게 메시지 $\{ID, c_4\}$를 전송한다.

3) 사용자 U는 서버 S로부터 전송받은 $\{ID, c_4\}$를 이용하여 자신의 식별자 ID와 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 이용하여 $c_4$의 무결성을 검증한다. 무결성이 검증되면 스마트카드 내에 저장된 $K = h(x \| ID) \oplus h(P \| N)$를 $K' = h(x \| ID) \oplus h(P \| N')$으로 N을 N'으로 각각 업데이트한다.

### 2-2 재전송 공격 취약성

사용자 U가 서버 S로부터 n번째 로그인을 하기 전에, 공격자 A는 두 가지의 인증 메시지 $\{ID, c_2^{(n-1)}, c_3^{(n-1)}\}$과 $\{ID, c_2^{(n-2)}, c_3^{(n-2)}\}$를 도청한다. 그 후 공격자 A는 사용자 U가 n번째 로그인을 하기 전에 사용자 U로 가장하기 위해 메시지 $\{ID, c_2^{(n)}, c_3^{(n)}\}$을 메시지 $\{ID, c_2^{(n-1)}, c_3^{(n-2)}\}$로 교환한다. 공격자 A는 서버 S에게 메시지 $\{ID, c_2^{(n-1)}, c_3^{(n-2)}\}$를 전송한다.

$$c_2^{(n-1)} = K^{(n-1)} \oplus h(P \| N^{(n)}) \quad (4)$$

$$c_3^{(n-2)} = h(h(x \| ID) \ \| \ h(P \| N^{(n-1)})) \quad (5)$$

서버 S는 $c_2^{(n-1)}$와 $K^{(n)}$를 연산하여 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n-1)})$를 계산한다.

$$\begin{aligned} & c_2^{(n-1)} \oplus K^{(n)} \\ &= K^{(n-1)} \oplus h(P \| N^{(n)}) \oplus K^{(n)} \\ &= h(x \| ID) \oplus h(P \| N^{(n-1)}) \oplus h(P \| N^{(n)}) \oplus h(x \| ID) \\ &\quad \oplus h(P \| N^{(n)}) \\ &= h(P \| N^{(n-1)}) \end{aligned} \quad (6)$$

서버 S는 $h(x \| ID)$와 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n-1)})$를 이용하여 $c_3^{(n-2)}$의 무결성을 검증한다.

$$c_3' = h(h(x \| ID) \| h(P \| N^{(n-1)}) \quad (7)$$

$$c_3^{(n-2)} = c_3' \quad (8)$$

서버 S는 계산된 $c_3'$와 공격자로부터 받은 $c_3^{(n-2)}$이 동일하기 때문에 공격자 A를 인증하게 되고, 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n)})$를 $h(P \| N^{(n-1)})$로 업데이트한다. 그 후 서버 S는 $c_4^{(n-2)} = h(ID \| h(P \| N^{(n-1)})$를 계산하여 정당한 사용자임이 입증된 공격자 A에게 $\{ID, c_4^{(n-2)}\}$를 전송한다. 하지만 사용자 U로 가장한 공격자 A는 $c_4^{(n-2)}$에 대한 무결성 검증 후 서버 S에게 인증 완료에 대한 메시지를 보내지 않기 때문에 공격자 A는 정당한 사용자임을 입증하여 인증에 성공할 수 있으므로, SPMA 프로토콜은 재전송 공격에 취약하다.

## Ⅲ. I-SPMA 프로토콜

본 장에서는 SPMA 프로토콜의 재전송 공격에 안전한 개선된 강력한 패스워드 상호인증 프로토콜인 I-SPMA를 제안한다.

SPMA 프로토콜은 $h(x \| ID)$를 이용하여 인증 메시지에 대한 무결성을 검증하기 때문에 재전송 공격이 발생한다. 제안한 I-SPMA 프로토콜에서는 재전송 공격을 방지하기 위해 인증단계에서 $h(x \| ID)$를 이용하지 않고, 현재 세션의 패스워드 검증자, 다음 세션의 패스워드 검증자, 현재 세션의 $K(= h(x \| ID) \oplus$ 현재 세션의 패스워드 검증자)를 이용하여 인증 메시지에 대한 무결성을 수행한다.

제안한 I-SPMA 프로토콜은 등록 단계와 인증 단계로 구성된다. 등록 단계에서는 사용자가 서버에 요청하여 자신의 아이디와 패스워드를 담고 있는 스마트카드를 발급받으며, 인증 단계에서는 사용자가 서버에 자신의 아이디, 패스워드, 스마트카드를 이용하여 상호인증을 받게 된다. 표 1은 I-SPMA 프로토콜에서 사용하는 시스템 파라미터를 나타낸다.

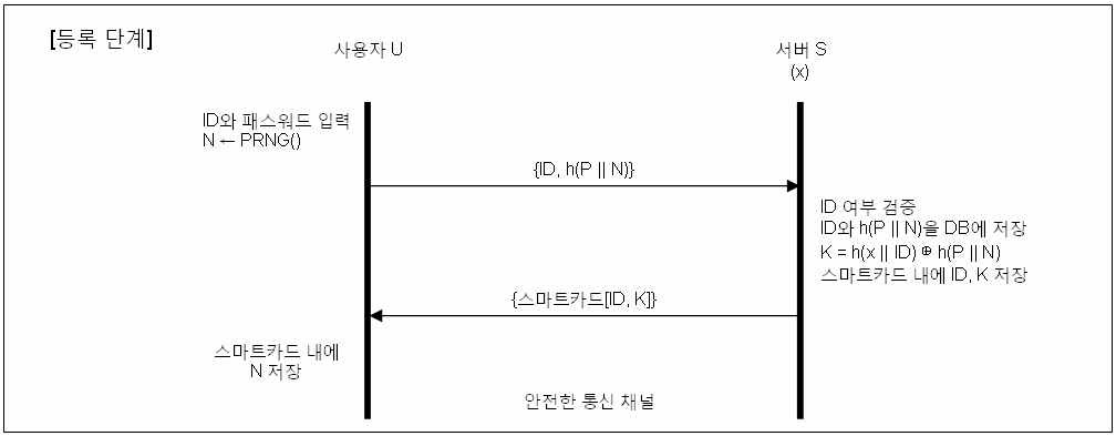


그림 2. 등록 단계
Fig 2. Registration phase.

표 1. 시스템 파라미터
Table 1. System parameter.

| 기호 | 의미 |
|---|---|
| U | 원격 사용자(remote user) |
| S | 원격 서버(remote server) |
| ID | 원격 사용자의 식별자(identifier) |
| P | 원격 사용자의 패스워드(password) |
| N | 현재 세션을 위한 랜덤 넌스(random nonce) |
| N' | 다음 세션을 위한 랜덤 넌스(random nonce) |
| x | 원격 서버의 비밀키(secret key) |
| h(·) | 일방향 해시 함수(one-way hash function) |
| PRNG(·) | 의사난수생성기(Pseudo Random Number generator) |
| ⊕ | 배타적 논리합 연산(Exclusive OR operation) |
| \|\| | 연접 연산(concatenation operation) |
| A→B: X | X가 A에서 B로 전송 |

### 3-1 등록 단계

새로운 사용자 U가 서비스에 접근하기 위해 서버 S에 등록을 요청한다. 등록 단계는 그림 2와 같이 안전한 채널을 통해 수행되며, 수행 절차는 다음과 같다.

1) U→S: {ID, h(P‖N)}

사용자 U는 클라이언트를 이용하여 PRNG(·)로부터 랜덤 넌스 N을 생성하고 패스워드 검증자 h(P‖N)를 계산한다. 사용자 U는 서버 S에 등록하기 위해 자신의 식별자 ID와 패스워드 검증자 h(P‖N)를 안전한 채널을 통하여 서버 S로 전송한다.

2) S→U: {스마트카드[ID, K]}

서버 S는 사용자 U의 식별자 ID가 기존에 가입된 사용자의 ID인지를 검증한다. 기존에 가입된 사용자의 ID가 아님이 식별되면, 서버의 데이터베이스에 사용자 U의 식별자 ID와 패스워드 검증자 h(P‖N)를 저장한다. 서버 S는 자신의 비밀키 x를 이용하여 K = h(x‖ID) ⊕ h(P‖N)를 계산하고, 스마트카드 내에 사용자 U의 식별자 ID와 K값을 저장한다. 그 후 사용자 U에게 안전한 채널을 통해 스마트카드를 발급한다.

3) 사용자 U는 발급받은 스마트카드와 클라이언트를 이용하여 스마트카드 내에 랜덤 넌스 N을 저장한다.

### 3-2 인증 단계

그림 3은 제안한 I-SPMA 프로토콜의 인증 단계를 나타낸다. 사용자 U가 서버 S에 접속하기 위해 클라이언트 컴퓨터에 있는 로그인 장치에 스마트카드를 입력한 후, 자신의 패스워드 P를 입력한다. 스마트카드는 클라이언트를 이용하여 사용자 U가 서버 S로부터 인증을 수행할 수 있도록 다음과 같은 연산을 수행한다.

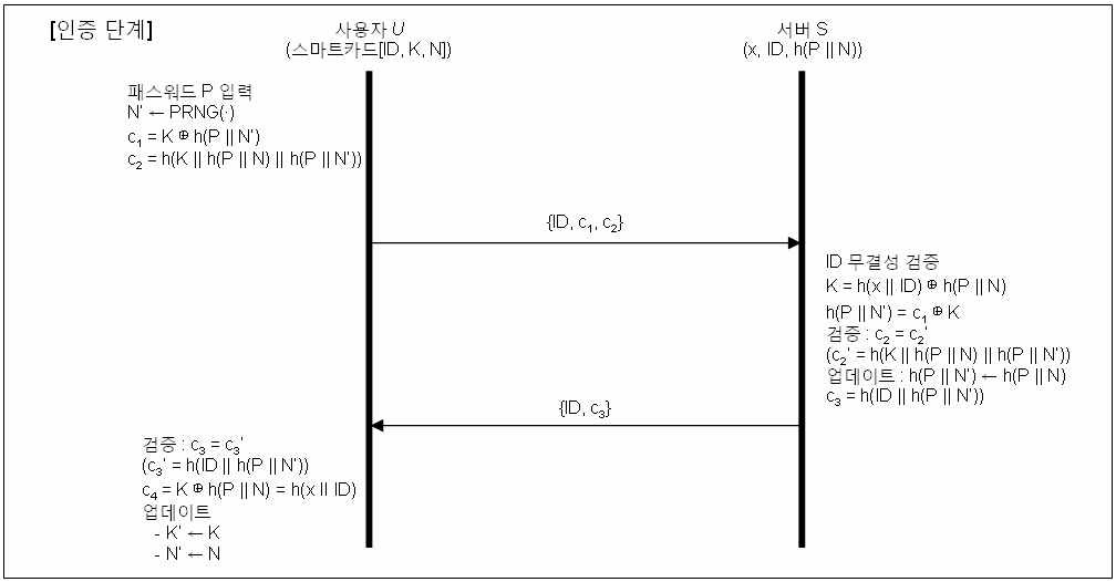


그림 3. 인증 단계
Fig 3. Authentication phase.

1) PRNG(·)로부터 다음 세션을 위한 랜덤 넌스 N'을 생성한다. $c_1$과 $c_2$를 다음과 같이 계산한다.

$$c_1 = K \oplus h(P \| N') \quad (9)$$

$$c_2 = h(K \| h(P \| N) \| h(P \| N')) \quad (10)$$

2) U → S: {ID, $c_1$, $c_2$}

사용자 U는 서버 S에게 로그인 요청을 위해 메시지 {ID, $c_1$, $c_2$}를 전송한다.

서버 S는 사용자 U로부터 전송받은 로그인 요청 메시지 {ID, $c_1$, $c_2$}를 이용하여 사용자 U의 신원을 식별하기 위해 다음과 같은 절차를 수행한다.

1) 사용자 U의 식별자 ID와 자신의 데이터베이스에 저장된 ID를 비교하여 등록된 ID인지를 검증한다. 만약 등록된 ID가 아니면, 서버 S는 현재 세션을 종료한다.

2) 등록된 ID임이 식별되면, 서버 S는 자신의 비밀키 x와 사용자 U의 ID를 이용하여 $h(x \| ID)$를 계산한다. 그 후 데이터베이스에 저장된 사용자 U의 패스워드 검증자 $h(P \| N)$와 $h(x \| ID)$를 이용하여 $K = h(x \| ID) \oplus h(P \| N)$를 계산한다.

3) 서버 S는 수신한 $c_1$과 K를 이용하여 다음의 연산을 수행해 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 계산한다.

$$\begin{aligned} & c_1 \oplus K \\ &= K \oplus h(P \| N') \oplus K \\ &= h(P \| N') \end{aligned} \quad (11)$$

4) 서버 S는 전송받은 $c_2$에 대한 무결성을 검증하기 위해 K, 현재 세션을 위한 패스워드 검증자 $h(P \| N)$, 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 이용하여 다음의 연산을 수행한다.

$$c_2' = h(K \| h(P \| N) \| h(P \| N')) \quad (12)$$

$$c_2 = c_2' \quad (13)$$

5) 서버 S는 사용자 U로부터 전송받은 값($c_2$)과 정당한 사용자 U로부터 전송되었는지 확인하기 위하여 계산한 값($c_2'$)을 비교하여 동일하다면 사용자 U를 인증한다. 다음의 로그인을 위해 자신의 데이터베이스 내에 저장된 기존의 패스워드 검증자 $h(P \| N)$를 다음 세션을 위한 패스워드 검증자 $h(P \| N')$으로 업데이트 한다. 만약 비교 값이 동일하지 않으면, 로그인 요청을 거절하고 현재 세션을 종료한다.

6) S → U: {ID, $c_3$}

서버 S는 사용자 U와 상호인증을 수행하기 위해 다음의 연산을 수행한 후, 사용자 U에게 메시지 {ID, $c_3$}을 전송한다.

$$c_3 = h(ID \| h(P \| N')) \quad (14)$$

사용자 U는 서버 S로부터 전송받은 메시지 {ID, $c_3$}을 이용하여 상호인증을 위해 다음과 같은 절차를 수행한다.

1) 사용자 U는 전송받은 $c_3$에 대한 무결성을 검증하기 위해 자신의 식별자 ID와 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 이용하여 다음의 연산을 수행한다.

$$c_3' = h(ID \| h(P \| N')) \quad (15)$$

$$c_3 = c_3' \quad (16)$$

2) 사용자 U는 서버 S로부터 전송받은 값($c_3$)과 정당한 서버 S로부터 전송되었는지 확인하기 위하여 계산한 값($c_3'$)을 비교하여 동일하다면 서버 S가 자신을 올바르게 인증하였는지에 대한 상호인증 여부를 알게 된다.

3) 사용자 U는 K와 현재 세션을 위한 패스워드 검증자 $h(P \| N)$를 이용하여 다음과 같은 연산을 수행한다.

$$c_4 = K \oplus h(P \| N) = h(x \| ID) \quad (17)$$

4) 사용자 U는 계산한 $h(x \| ID)$를 이용하여 다음의 로그인을 위해 자신의 스마트카드 내에 저장된 $K = h(x \| ID) \oplus h(P \| N)$를 $K' = h(x \| ID) \oplus h(P \| N')$으로 현재 세션을 위한 랜덤 넌스 N을 다음 세션을 위한 랜덤 넌스 N'으로 각각 업데이트한다. 만약 비교 값이 동일하지 않으면, 상호인증 요청을 거절하고 현재 세션을 종료한다.

## Ⅳ. 안전성 분석

본 장에서는 제안한 I-SPMA 프로토콜에 대한 안전성을 패스워드 추측 공격, 재전송 공격, 위장 공격, 훔친 검증자 공격, 서비스 거부 공격, 상호 인증으로 구분하여 분석한다.

### 4-1 패스워드 추측 공격 (Password guessing attack)

공격자 A가 공개된 네트워크에서 사용자 U의 메시지 {ID, $c_1$, $c_2$, $c_3$}을 가로챘다고 가정한다. 공격자 A는 메시지 {$c_1$, $c_2$, $c_3$}을 가지고 있더라도 사용자 U의 패스워드를 추측할 수 없다.

즉, 공격자 A가 사용자 U의 패스워드를 추측하기 위해 메시지 {ID, $c_1$, $c_2$, $c_3$}을 가로챘다고 가정한다. 메시지 {$c_1$, $c_2$, $c_3$}은 다음과 같다.

$$c_1 = K \oplus h(P \| N') \quad (18)$$

$$c_2 = h(K \| h(P \| N) \| h(P \| N')) \quad (19)$$

$$c_3 = h(ID \| h(P \| N')) \quad (20)$$

공격자 A가 메시지 {$c_1$, $c_2$, $c_3$}을 가지고 있더라도 현재 세션을 위한 랜덤 넌스 N, 다음 세션을 위한 랜덤 넌스 N', 서버의 비밀키 x를 알아야만 패스워드 추측 공격을 수행하여 P를 획득할 수 있다. 하지만 일방향 해시 함수의 복구불가능성으로 인해 공격자는 메시지 {$c_1$, $c_2$, $c_3$}로부터 N, N', x를 알아 낼 수 없기 때문에 제안한 I-SPMA 프로토콜은 패스워드 추측 공격에 안전하다.

### 4-2 재전송 공격(Replay attack)

공격자 A가 공개된 네트워크에서 이전 로그인 단계에서 전송된 메시지 {$c_1$, $c_2$}를 도청한 것으로 가정한다. 사용자는 로그인을 할 때마다 새로 생성되는 랜덤 넌스 N'과 패스워드 검증자 $h(P \| N')$을 사용하기 때문에 공격자는 이전 로그인에서 전송된 메시지를 획득하여 다음 로그인에 그 메시지를 사용하더라도 서버로부터 인증을 받을 수 없다.

즉, 사용자 U가 서버 S로부터 n번째 로그인을 하

기 전에, 공격자 A는 {ID, $c_1^{(n-1)}$, $c_2^{(n-1)}$}과 {ID, $c_1^{(n-2)}$, $c_2^{(n-2)}$}를 도청할 수 있는 것으로 가정한다. 공격자 A는 메시지 {ID, $c_1^{(n-1)}$, $c_2^{(n-1)}$}과 메시지 {ID, $c_1^{(n-2)}$, $c_2^{(n-2)}$}를 도청한다. 그 다음 공격자 A는 사용자 U가 n번째 로그인을 하기 전에 사용자 U로 가장하기 위해 메시지 {ID, $c_1^{(n)}$, $c_2^{(n)}$}을 메시지 {ID, $c_1^{(n-1)}$, $c_2^{(n-2)}$}로 교환한다. 공격자 A는 서버 S에게 메시지 {ID, $c_1^{(n-1)}$, $c_2^{(n-2)}$}를 전송한다.

$$c_1^{(n-1)} = K^{(n-1)} \oplus h(P \| N^{(n)}) \quad (21)$$

$$c_2^{(n-2)} = h(K^{(n-2)} \| h(P \| N^{(n-2)}) \| h(P \| N^{(n-1)})) \quad (22)$$

서버 S는 $c_1^{(n-1)}$과 $K^{(n)}$를 연산하여 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n-1)})$를 계산한다.

$$\begin{aligned} & c_1^{(n-1)} \oplus K^{(n)} \\ &= K^{(n-1)} \oplus h(P \| N^{(n)}) \oplus K^{(n)} \\ &= h(x \| ID) \oplus h(P \| N^{(n-1)}) \oplus h(P \| N^{(n)}) \oplus h(x \| ID) \\ &\quad \oplus h(P \| N^{(n)}) \\ &= h(P \| N^{(n-1)}) \end{aligned} \quad (23)$$

서버 S는 $K^{(n)}$, 현재 세션을 위한 패스워드 검증자 $h(P \| N^{(n)})$, 위의 식에서 계산된 패스워드 검증자 $h(P \| N^{(n-1)})$를 이용하여 $c_2^{(n-2)}$의 무결성을 검증한다.

$$c_2' = h(K^{(n)} \| h(P \| N^{(n)}) \| h(P \| N^{(n-1)})) \quad (24)$$

$$c_2^{(n-2)} \neq c_2' \quad (25)$$

서버 S는 공격자 A로부터 전송받은 값($c_2^{(n-2)}$)과 정당한 사용자 U로부터 전송되었는지 확인하기 위하여 계산한 값($c_2'$)이 동일하지 않기 때문에 현재 세션을 종료한다. 따라서 제안한 I-SPMA 프로토콜은 재전송 공격에 안전하다.

### 4-3 위장 공격(Impersonation attack)

공격자 A는 사용자 U로 위장하기 위해 $c_{A_1} = K_A \oplus h(P_A \| N_A')$, $c_{A_2} = h(K_A \| h(P_A \| N_A) \| h(P_A \| N_A'))$를 계산하여 위조된 로그인 요청 메시지 {ID, $c_{A_1}$, $c_{A_2}$}를 서버 S에게 전송할 수 있다. 서버 S는 위조된 로그인 요청 메시지 {ID, $c_{A_1}$, $c_{A_2}$}를 전송받아 인증을 수행하지만, 위조된 로그인 요청 메시지 {ID, $c_{A_1}$, $c_{A_2}$}는 인증에 실패한다.

즉, 서버 S는 자신의 데이터베이스에 저장되어 있는 x와 사용자 U의 ID를 이용해 $h(x \| ID)$를 계산하고, $h(x \| ID)$와 현재 세션을 위한 패스워드 검증자 $h(P \| N)$를 이용해 K를 계산하여 다음 세션을 위한 패스워드 검증자 $K_A \oplus h(P_A \| N_A') \oplus K$를 획득하게 된다. 그 후 현재 세션에서 사용하고 있는 K, 현재 세션을 위한 패스워드 검증자 $h(P \| N)$, 위에서 계산된 다음 세션을 위한 패스워드 검증자 $K_A \oplus h(P_A \| N_A') \oplus K$를 이용해 다음과 같은 연산을 수행하여 전송받은 $c_{A_2}$의 무결성을 검증한다.

$$c_{A_2} \neq h(K \| h(P \| N) \| K_A \oplus h(P_A \| N_A') \oplus K) \quad (26)$$

$h(K_A \| h(P_A \| N_A) \| h(P_A \| N_A'))$와 서버에서 계산된 $h(K \| h(P \| N) \| K_A \oplus h(P_A \| N_A') \oplus K)$는 동일하지 않기 때문에 $c_{A_2}$는 무결성 검증에 실패한다. 따라서 제안한 I-SPMA 프로토콜은 위장 공격에 안전하다.

### 4-4 훔친 검증자 공격(Stolen-verifier attack)

공격자 A는 사용자의 n-1번째 로그인 후에 패스워드 검증자 $h(P \| N)$를 훔치고, 메시지 {$c_1$, $c_2$}를 도청한 것으로 가정한다. 사용자는 세션마다 K, 현재 세션을 위한 패스워드 검증자 $h(P \| N)$, 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 사용하기 때문에 공격자 A가 패스워드 검증자 $h(P \| N)$를 획득하여도 훔친 검증자 공격에 성공할 수 없다.

즉, 공격자 A는 사용자의 n-1번째 로그인 후에 패스워드 검증자 $h(P \| N^{(n)})$를 훔쳤다고 가정한다. 사용자의 n번째 로그인 절차 동안에 공격자 A는 메시지 {$c_1^{(n)}$, $c_2^{(n)}$}을 도청한다. 공격자 A는 K를 알아내기 위해 $c_1^{(n)}$과 패스워드 검증자 $h(P \| N^{(n)})$를 이용하여 다음을 계산한다.

$$\begin{aligned} & c_1 \oplus h(P \| N^{(n)}) \\ &= K^{(n)} \oplus h(P \| N^{(n+1)}) \oplus h(P \| N^{(n)}) \\ &= h(x \| ID) \oplus h(P \| N^{(n)}) \oplus h(P \| N^{(n+1)}) \oplus h(P \| N^{(n)}) \\ &= h(x \| ID) \oplus h(P \| N^{(n+1)}) = K^{(n+1)} \end{aligned} \quad (27)$$

그 후 사용자의 n+1번째 로그인 절차 동안에 공격자 A는 메시지 $\{c_1^{(n+1)}, c_2^{(n+1)}\}$을 도청한다. 공격자 A는 서버 S로부터 인증을 받기 위해 자신이 만든 $P_A$, 랜덤 넌스 $N_A^{(n+1)}$, 랜덤 넌스 $N_A^{(n+2)}$와 $K^{(n+1)}$를 이용하여 다음을 계산한다.

$$c_{A_1}^{(n+1)} = K^{(n+1)} \oplus h(P_A \| N_A^{(n+2)}) \quad (28)$$

$$c_{A_2}^{(n+1)} = h(K^{(n+1)} \| h(P_A \| N_A^{(n+1)}) \| h(P_A \| N_A^{(n+2)})) \quad (29)$$

공격자 A는 서버 S에 메시지 $\{ID, c_{A_1}, c_{A_2}\}$를 전송한다. 서버 S는 $c_{A_1}^{(n+1)}$과 $K^{(n+1)}$를 연산하여 다음 세션을 위한 패스워드 검증자 $h(P_A \| N_A^{(n+2)})$를 계산한다.

$$\begin{aligned} & c_{A_1}^{(n+1)} \oplus K^{(n+1)} \\ &= K^{(n+1)} \oplus h(P_A \| N_A^{(n+2)}) \oplus K^{(n+1)} \\ &= h(x \| ID) \oplus h(P \| N^{(n+1)}) \oplus h(P_A \| N_A^{(n+2)}) \oplus h(x \| ID) \oplus h(P \| N^{(n+1)}) \\ &= h(P_A \| N_A^{(n+2)}) \end{aligned} \quad (30)$$

서버 S는 $K^{(n+1)}$, 현재 세션을 위한 패스워드 검증자 $h(P \| N^{(n+1)})$, 위의 식에서 계산된 패스워드 검증자 $h(P_A \| N_A^{(n+2)})$를 이용하여 $c_{A_2}^{(n+2)}$의 무결성을 검증한다.

$$c_2' = h(K^{(n+1)} \| h(P \| N^{(n+1)}) \| h(P_A \| N_A^{(n+2)})) \quad (31)$$

$$c_{A_2}^{(n+2)} \neq c_2' \quad (32)$$

서버 S는 공격자 A로부터 전송받은 값($c_{A_2}^{(n+2)}$)과 정당한 사용자 U로부터 전송되었는지 확인하기 위하여 계산한 값($c_2'$)이 동일하지 않기 때문에 현재 세션을 종료한다. 공격자가 공격에 성공하기 위해서는 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n+1)})$을 알아야 되지만, 제안한 프로토콜에서는 다음 세션을 위한 패스워드 검증자 $h(P \| N^{(n+1)})$을 알 수 없다. 따라서 제안한 I-SPMA 프로토콜은 훔친 검증자 공격에 안전하다.

### 4-5 서비스 거부 공격 (Denial of Service attack)

인증을 수행하는 과정에서 다음 세션을 위한 패스워드 검증자 $h(P \| N')$를 이용하여 다음 세션을 위한 패스워드 검증자 $h(P \| N')$가 포함되어 있는 $\{c_2, c_3\}$에 대한 무결성 검증을 수행한다. 따라서 공격자 A가 메시지 $\{c_1, c_2\}$를 도청하여 서비스 거부 공격을 수행한다고 해도 공격에 성공할 수 없다.

즉, 사용자 U가 서버 S로부터 n번째 로그인을 할 때 공격자 A는 메시지 $\{c_1, c_2\}$를 가로챈 것으로 가정한다. 공격자 A는 $c_1$과 다음 세션을 위한 패스워드 검증자의 동일한 크기 R을 이용하여 다음을 계산한다.

$$c_1 = K \oplus R \quad (33)$$

$$c_2 = h(K \| h(P \| N) \| h(P \| N')) \quad (34)$$

공격자 A는 서버 S에게 메시지 $\{c_1, c_2\}$를 전송한다. 서버 S는 $c_1$과 K를 연산하여 다음 세션을 위한 패스워드 검증자의 동일한 크기 R을 계산한다.

$$\begin{aligned} & c_1 \oplus K \\ &= K \oplus R \oplus K \\ &= R \end{aligned} \quad (35)$$

서버 S는 K, 현재 세션을 위한 패스워드 검증자 $h(P \| N)$, 위의 식에서 계산된 R을 이용하여 $c_2$의 무결성을 검증한다.

$$c_2' = h(K \| h(P \| N) \| R) \quad (36)$$

$$c_2 \neq c_2' \quad (37)$$

서버 S는 공격자 A로부터 전송받은 값($c_2$)과 정당한 사용자 U로부터 전송되었는지 확인하기 위하여 계산한 값($c_2'$)이 동일하지 않기 때문에 현재 세션을 종료한다. 만약 공격자 A가 서비스 거부 공격을 성공하기 위해서는 다음과 같이 계산되어야 한다.

$$c_1 = K \oplus R \quad (38)$$

$$c_2 = h(K \| h(P \| N) \| R) \quad (39)$$

위의 식처럼 $c_2$를 만들기 위해서는 $h(x \| ID)$와 현재 세션을 위한 패스워드 검증자 $h(P \| N)$를 알고 있

표 2. 안전성 비교·분석
Table 2. Comparison·analysis of safety.

| 구분 | SE-OSPA 프로토콜 | NSPA 프로토콜 | SPMA 프로토콜 | I-SPMA 프로토콜 |
|---|---|---|---|---|
| 패스워드 추측 공격 | ○ | ○ | ○ | ○ |
| 재전송 공격 | × | ○ | × | ○ |
| 위장 공격 | ○ | ○ | ○ | ○ |
| 훔친 검증자 공격 | ○ | ○ | ○ | ○ |
| 서비스 거부 공격 | × | ○ | ○ | ○ |
| 상호인증 | × | × | ○ | ○ |

어야 한다. 하지만 공격자 A는 $h(x \| ID)$와 현재 세션을 위한 패스워드 검증자 $h(P \| N)$를 알아낼 수 없다. 따라서 제안한 I-SPMA 프로토콜은 서비스 거부 공격에 안전하다.

### 4-6 상호인증(Mutual authentication)

서버 S는 사용자 U로부터 전달받은 로그인 요청 메시지 $c_1 = K \oplus h(P \| N')$에서 다음 세션을 위한 패스워드 검증자 $h(P \| N')$을 계산한다. 그 후 K와 현재 세션을 위한 패스워드 검증자 $h(P \| N)$를 이용해 $c_2 = h(K \| h(P \| N) \| h(P \| N'))$에 대한 무결성을 검증하여 사용자를 인증한 후 다음 세션을 위한 패스워드 검증 $h(P \| N')$으로 업데이트한다. 또한 사용자 U도 서버 S로부터 전달받은 인증 메시지 $c_3 = h(ID \| h(P \| N'))$가 정당한 서버로부터 전송된 메시지인지 여부를 확인하기 위해 다음 세션을 위한 패스워드 검증자 $h(P \| N')$와 ID를 이용하여 $c_3$에 대한 무결성을 검증한 후 다음 세션을 위한 패스워드 검증자 $h(P \| N')$으로 업데이트한다. 따라서 제안한 I-SPMA 프로토콜은 상호인증을 제공한다.

## Ⅴ. 결 론

본 논문에서는 기존에 제안된 SPMA 프로토콜이 재전송 공격에 대한 취약성을 분석하였고, 재전송 공격에 안전한 개선된 강력한 패스워드 상호인증 프로토콜인 I-SPMA를 제안하였다. I-SPMA 프로토콜은 패스워드 추측 공격, 재전송 공격, 위장 공격, 훔친 검증자 공격, 서비스 거부 공격에 안전하다는 것을 증명하였다. 또한 제안한 I-SPMA 프로토콜은 SPMA 프로토콜과 동일한 효율성 및 상호인증을 제공하기 위해 7번의 해시 연산과 4번의 배타적 논리합 연산을 수행하였다. 따라서 본 논문에서 제안한 I-SPMA 프로토콜은 개방형 네트워크상에서 SPMA 프로토콜과 동일한 효율성 및 상호인증을 제공하면서 보안성이 향상된 패스워드 기반의 인증 방식에 사용할 수 있을 것으로 기대된다.

## 참 고 문 헌

[1] L. Lamport, “Password Authentication with Insecure Communication," *Communication of ACM, Vol. 24, no. 11*, pp. 770-772, November 1981.

[2] C. K. Chan and L. M. Cheng, “Cryptanalysis of timestamp-based password authentication scheme," *Computer & Security, Vol. 21, no. 1*, pp. 74-76, 2002.

[3] H. Y. Chien, J. K. Jan, and Y. M. Tseng, “A modified remote login authentication scheme based on geometric approach,” *The Journal of Systems and Software 55*, pp. 287-290, 2001.

[4] M. S. Hwang, “A remote password authentication scheme based on the digital signature method”, *International Journal of Computer Mathematics, Vol. 70, no. 4*, pp. 657-666, 1999.

[5] M. S. Hwang, C. C. Lee, and Y. L. Tang, “An Improvement of SPLICE/AS in WIDE Against Guessing Attack,” *Institute of Mathematics and Informatics, Vol. 12, no. 2*, pp. 297-302, 2001.

[6] C. C. Lee, M. S. Hwang, and W. P. Yang, “A

Flexible Remote User Authentication Scheme Using Smart Cards," *ACM Operating Systems Review, Vol. 36, no. 3*, pp. 46-52, 2002.

[7] C. C. Lee, L. H. Li, and M. S. Hwang, "A Remote User Authentication Scheme Using Hash Functions," *ACM SIGOPS Operating Systems Review, Vol. 36. no. 4*, pp. 23-29, October 2002.

[8] L. H. Li, I. C. Lin, and M. S. Hwang, "A Remote Password Authentication Scheme for Multiserver Architecture Using Neural Networks," *IEEE Transactions on Neural Networks, Vol. 12, no. 6*, pp. 1498-1504, November 2001.

[9] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A Simple Remote User Authentication Scheme," *Mathematical and Computer Modelling, Vol. 36, no. 1*, pp. 103-107, July 2002.

[10] J. J. Shen, C. W. Lin, and M. S. Hwang, "A Modified Remote User Authentication Scheme Using Smart Cards," *IEEE Transactions on Consumer Electronics, Vol. 49, no. 2*, pp. 414-416, May 2003.

[11] W. H. Yang and S. P. Shied, "Password Authentication Scheme with Smart Cards," *Computers & Security, Vol. 18, no. 8*, pp 727-733, 1999.

[12] A. Shimizu, "A dynamic password authentication method by one-way function," *IEICE Transactions on Communications, Vol. J73-D-I, no. 7*, pp. 630-636, July 1990.

[13] A. Shimizu, "A dynamic password authentication method by one-way function," *System and Computers in Japan, Vol. 22, no. 7*, pp. 32-40, July 1991.

[14] A. Simizu, T. Horioka, and H. Inagaki, "A password authentication method for contents communication on the internet," *IEICE Transactions on Communications, Vol. E81-B, no. 8*, pp. 1666-1673, August 1998.

[15] M. Sandirigama, A. Shimizu, and M. T. Noda, "Simple and Secure Password Authentication Protocol," *IEICE Transactions on Communications, Vol. E83-B, no. 6*, pp. 1363-1365, June 2000.

[16] C. M. Chen and W. C. Ku, "Stolen-Verifier Attack on Two New Strong-Password Authentication Protocols," *IEICE Transactions on Communications, Vol. E85-B, no. 11*, pp. 2519-2521, November 2002.

[17] C. L. Lin, H. M. Sun, and T. Hwang, "Attacks and Solutions on Strong-Password Authentication," *IEICE Transactions on Communications, Vol. E84-B, no. 9*, pp. 2622-2627, September 2001.

[18] C. W. Lin, J. J. Shen, and M. S. Hwang, "Security Enhancement for Optimal Strong-Password Authentication Protocol," *ACM SIGOPS Operating Systems Review, Vol. 37, no. 2*, pp. 7-12, April 2003.

[19] W. C. Ku, H. C. Tsai, and S. M. Chen, "Two simple attacks on Lin-Shen-Hwang's strong-password authentication protocol," *ACM SIGOPS Operating Systems Review, Vol. 37, no. 4*, pp. 26-31, October 2003.

[20] C. W. Lin, C. S. Tsai, and M. S. Hwang, "A New Strong-Password Authentication Scheme Using One-Way Hash Functions," *Journal of Computer and Systems Sciences International, Vol. 45, no. 4*, pp. 623-626, January 2006.

[21] 윤은준, 홍유식, 김천식, 유기영, "강력한 패스워드 상호인증 프로토콜," *전자공학회논문지, 제46권 CI편, 제1호*, 11-19쪽, 2009년 1월

## 김 준 섭 (金俊燮)

2010년 2월 : 순천향대학교 정보보호학과(공학사)
2010년 3월~현재 : 순천향대학교 정보보호학과 석사과정
관심분야 : 정보보호, 정보보호제품 평가, ID 관리, 클라우드 컴퓨팅 등

## 곽 진 (郭鎭)

1994년~2006년 : 성균관대학교 학사, 석사, 박사
2006년 4월~2006년 11월 : 일본 큐슈대학교 시스템정보공학부 방문연구원
2006년 8월~2006년 11월 : 일본 큐슈 시스템정보기술연구소 특별연구원
2006년~2007년 2월 : 정보통신부 정보보호기획단 개인정보보호팀 통신사무관
2007년 2월~현재 : 순천향대학교 정보보호학과 교수
관심분야 : 암호프로토콜, RFID 시스템 응용 보안, 개인정보보호, 정보보호제품 평가, 클라우드 컴퓨팅 등