# 다중서버 환경에서의 스마트카드를 이용한 상호 익명 인증

정회원 유 혜 정*

# Mutual Anonymous Authentication Using Smart Cards in Multi-server Environments

Hye-joung Yoo* ***Regular Member***

요 약

컴퓨터 네트워크 환경에서 사용자가 서버에서 제공되는 서비스를 요청할 때, 스마트카드를 이용한 사용자 인증 스킴은 사용자의 적격성을 입증하고 안전한 통신을 제공하는 매우 실용적인 기술이다. 이러한 사용자 인증 환경에서 네트워크와 정보기술의 발전으로 인하여 대부분의 서비스는 다중서버 환경으로 변화되었으며, 이는 점점 더 가속화 될 것이다. 그러나 아직까지 스마트카드를 이용한 다중서버 환경을 지원하는 상호 익명성 제공 인증에 대한 연구는 존재하지 않는다. 본 논문에서는 사용자를 인증하는 기술 중에서 다중서버 환경을 지원하며, 동시에 서버와 사용자 간의 상호 인증 및 상호 익명성을 제공하는 스마트카드를 이용한 사용자 인증 스킴을 제안하고자 한다. 스마트카드를 이용하여 계산량 측면에서 효율성을 취하고, 사용자는 한 번의 등록으로 등록한 서버에 관계없이 서버 군에서 제공하는 다양한 서비스를 안전하게 제공받을 수 있도록 하였으며, 사용자의 프라이버시 보호를 위해 원격 서버에게도 안전한 익명성을 제공받으면서 서버와의 상호 인증을 수행하도록 하였다.

**Key Words** : Smart cards, Authentication, Anonymity, Multi-server environments

ABSTRACT

In a network environment, when a user requests a server's service, a remote user authentication system using smart cards is a very practical solution to validate the eligibility of a user and provide secure communication. In these authentication schemes, due to fast progress of networks and information technology, most of provided services are in multi-server environments. However, there are no studies in multi-server authentication schemes using smart cards providing mutual anonymity so far. In this paper, we propose a novel user authentication scheme using smart cards providing mutual authentication and mutual anonymity for multi-server environments. Our proposed scheme achieves the low-computation requirement for smart cards and a user can use permitted various services in eligible servers by only one registration. Also, this scheme guarantees perfect mutual anonymity of participants.

## Ⅰ. 서 론

최근 컴퓨터 네트워크 사용의 증가로 인해 많은 사람들이 분산된 컴퓨터 환경에서 원격 서버에 접속하는 일이 빈번해지고 있다. 사용자가 정보 자산에 접근을 요청할 때 시스템과 관리자는 인증과정

* 세종사이버대학교 정보보호시스템학과(hjyoo@sjcu.ac.kr)
논문번호 : KICS2008-09-411, 접수일자 : 2008년 9월 19일, 최종논문접수일자 : 2008년 10월 17일

을 통하여 서버에서 제공하는 서비스의 접근 여부를 결정하게 된다.

일반적인 인증 시스템에서는 등록과 로그인 과정에서 사용자의 개인적인 정보를 전달하게 되는데, 이 과정에서 사용자의 익명성은 서비스를 요청하고 서비스에 접근하는 동안 제공되어야 할 바람직한 안전성 요소라고 할 수 있으며, 최근 스마트카드를 이용한 인증 시스템에서도 사용자의 익명성을 제공하는 인증기술에 대한 연구가 진행되고 있다.

그러나 지금까지 제안된 익명성을 제공하는 스마트카드를 이용한 인증 스킴들은 인증 단계에서 사용자의 익명성은 제공하지만 서버의 익명성은 제공하지 못한다. 서버를 알지 못하는 상태에서 사용자가 인증을 위해 자신의 정보를 제공한다는 것은 어떻게 보면 매우 부적절해 보일 수 있다. 그러나 네트워크 환경과 정보기술의 빠른 변화는 기존의 서버와 사용자의 개념이나 공급자와 소비자 개념에서 벗어나 모든 참여자가 공급자인 동시에 수요자가 되는 형태로 진화하고 있다. P2P(peer to peer) 서비스 제공 환경에서 사용자가 서비스 제공자인 동시에 사용자가 되는 경우를 예를 들 수 있다. 이러한 환경에서는 서버에 대한 익명성은 사용자의 익명성의 필요성과 더불어 동일하게 다루어져야 될 안전성 요소라 할 수 있다.

네트워크 구조가 다중서버 환경으로 변화하면서 단일서버가 아닌 다중서버 환경에 적합한 사용자 인증기술이 요구되었다. 단일서버 환경을 위한 사용자 인증 스킴을 다중서버 환경으로 확장하기 위해서는 하나의 서버에 대한 인증 환경을 그대로 복사하여 서비스 제공에 사용되는 모든 서버들에 대하여 반복적으로 적용하는 방법이 존재한다. 그러나 이 방법은 사용자가 각 서버에 대하여 반복적으로 등록해야 하며, 서비스를 요청할 때마다 각 서버에 대한 모든 아이디와 패스워드를 기억하여 그에 맞는 아이디와 패스워드를 입력해야만 하므로 매우 불편할 뿐 아니라 안전성 면에서도 매우 위험하다. 이 문제를 극복하기 위해서 다중서버 환경을 위한 사용자 인증 스킴이 제안되었으며, Lin et al.[1] 에서는 스마트카드를 이용한 안전한 다중서버 인증 스킴을 위한 조건을 안전성과 관련하여 다음과 같이 요약하고 있다:

1. 반복적인 등록 없이 한 번의 등록으로 다중서버 군에 로그인이 가능해야 한다.
2. 패스워드를 포함한 테이블을 사용하지 않는다.
3. 사용자들은 자신의 패스워드를 자유롭게 선택할 수 있다.
4. 재사용 공격에 강하다.

본 논문에서는 스마트카드를 이용한 효율적인 다중서버 환경에서의 속성 기반 사용자 인증 스킴을 제안하고자 한다. 제안된 스킴은 일반적인 스마트카드를 이용한 사용자 인증 스킴이 만족해야하는 성질을 만족하며, 또한 위에 언급된 Lin et al.[1]의 네 가지 조건을 포함한 다중서버 환경으로 확장 시 고려되어야 할 성질과 새로운 안전성 요소도 만족한다. 상호 익명성과 다중서버 환경에서 중요하게 고려되는 안전성 요소인 공모공격 불가능성 등을 제공하기 위해 기존의 그룹서명 기법의 특성을 인증 프로토콜에 도입, 사용자와 서버간의 인증 단계에서 사용자로부터 서버에게 전달되어지는 인증 메시지에 스마트카드에 저장된 사용자의 비밀 값에 대한 서버 군의 변형된 그룹서명을 포함시켜 서버가 사용자의 인증 메시지로부터 받은 비밀 값이 정당한지를 검증하는 방법으로 제안되어진다.

스마트카드를 이용한 상호 익명성을 제공하는 다중서버 환경 인증기법은 다양한 서비스를 제공받을 수 있는 분산 환경에서의 인증 시스템과 같은 익명성을 필요로 하는 많은 환경에서 활용할 수 있을 것으로 기대되며, P2P 환경에서 디지털 콘텐츠의 저작권 문제와 더불어 가장 큰 문제로 제기되는 사용자 상호간의 정보 노출 방지 등 프라이버시 침해에 대한 해결책을 제시할 수 있을 것이라 생각된다. 또한 상호 익명성과 더불어 사용자 공모 공격 불가능성과 기존 다중서버 환경 인증 스킴에서 가정하였던 중앙 관리자(CM; Central Manager) 등 서버 군을 위해 다중환경을 구성하는 특정 구성원에 대한 높은 의존도에 내포된 절대적 신뢰도를 낮추는 등 안전성 요소를 추가하면서도 Lin et al.[1]의 인증 스킴보다 효율적이다.

본 논문의 구성은 다음과 같다. 제 Ⅱ장에서는 제안된 시스템 구조와 관련된 연구들에 대해 간략하게 서술하고, 제 Ⅲ장에서는 그룹서명을 기반으로 다중서버 환경에 적합한 새로운 프로토콜을 제안한다. 다음으로 Ⅳ장에서는 제안된 프로토콜의 안전성과 효율성을 분석하고, 마지막으로 제 Ⅴ장에서 결론을 끝으로 본 논문을 마무리 짓고자 한다.

## Ⅱ. 용어 및 관련 연구

이번 장에서는 제안된 시스템 구조와 관련된 연구들에 대해 간략하게 알아본다. 먼저 본 논문에서

사용되고 있는 주요 용어에 대해 살펴보기로 한다.

### 2.1 용어

- $U_i$: $i$번째 사용자
- $ID_i$, $pw_i$: $U_i$의 아이디와 패스워드
- $\Sigma=\{S_1,S_2,\cdots,S_\eta\}$: 서버 군
- $G_1, G_2$: 위수가 $l$-비트 소수 $p$인 그룹
- $e$: 군 $G$에 대하여 $G_1\times G_2\rightarrow G$인 쌍일차 함수
- $h_1, h_2, H$: 일방향 해쉬 함수
- $\oplus$ : 배타적 논리합
- $(A_\nu, x_\nu)$ : 서버 $S_\nu\,(1\le\nu\le\eta)$의 비밀 키
- $T$, $\triangle T$ : 타임스탬프(time stamp), 타임스탬프 허용제한시간

### 2.2 관련 연구

단일서버 환경에서 일반적인 사용자 인증 시스템은 사용자와 서버, 두 개의 구성원으로 구성되며, 등록, 로그인 그리고 검증의 세 단계를 거치게 된다.

계산량과 통신량 측면에서 효율성을 취하고자 스마트카드에 기반 한 많은 인증 시스템이 제안되었으며, 스마트카드를 이용한 인증 시스템에서 사용자 익명성에 대한 연구는 동적 아이디(Dynamic identity)를 이용하여 사용자 익명성을 제공하는 방법으로 2004년 Das et al.[4]에 의해 처음으로 제안되었다.

Lin et al.[1]은 유클리드 평면의 기하학적 성질에 기반한 다중서버 환경을 위한 스마트카드를 이용한 효율적인 인증 스킴을 제안하였다. 이 스킴은 초기화, 등록, 로그인 그리고 인증의 네 단계로 구성되어 있으며, 사용자, 다양한 서버들 그리고 절대적 신뢰를 기반으로 공개/비밀 파라미터를 조직하는 중앙 관리자(CM; Central Manager)를 참여자로 가지고 있다. 이 스킴에서 등록을 원하는 새로운 사용자는 시스템을 구성하는 다양한 서버 중 자신이 원하는 서버에 자신의 아이디와 패스워드를 전송한다. 그러나 이때 해당 서버는 사용자의 로그인 정보를 직접 생성하는 것이 아닌 사용자의 아이디와 패스워드를 CM에게 전송하게 되고, CM은 이 정보를 통해 해당 서버를 위한 정보와 새로운 사용자를 위한 정보를 생성한다. 따라서 이 스킴은 논문에서 이야기하는 것처럼 시스템을 구성하는 다양한 서버 중에서 사용자가 원하는 서버와 등록 단계를 수행하는 것이 아닌 CM과 등록 단계를 거친다고 할 수 있다.

W. Juang[5]은 계산량과 통신량을 줄인 스마트카드를 이용한 다중서버 패스워드 인증 스킴을 제안하였다. 이 스킴은 사용자, 서버 군 그리고 등록 센터(RC; Registration Center)를 참여자로 가지고 있으며, 이 중 RC는 모든 참여자의 절대적 신뢰성 가정을 바탕으로 로그인하는 사용자들의 정당성을 조사하여 자격 있는 사용자에 대해 스마트카드를 발급하는 역할을 수행한다. 이는 모든 사용자가 RC와 등록 단계를 수행함으로써 서버 군에서 제공하는 모든 서비스를 이용할 수 있도록 한 것이다. 그러나 이는 단일서버 환경의 인증 스킴에서 하나의 서버가 담당했던 등록과 인증 두 기능을 별도로 분리함으로써 RC의 모든 서버에 대한 지배구조를 형성하였으며, 특히 로그인을 요청하는 사용자에 대해 해당 서버가 사용자 로그인 정보의 정당성을 확인하기 위해 RC와의 통신이 필요하게 됨으로써 RC에 대한 병목현상이 발생할 수 있다.

본 논문에서는 안전하고 효율적이면서 다중서버 환경에 적합한 인증 스킴을 구성하기 위해 Boneh et al.[2]의 그룹서명을 적용하였다. 이 서명에 대한 안전성은 SDH(Strong Diffie-Hellman) 가정과 DL(Decision Linear) 가정에 기반 한다. 서버 군에 속하는 모든 서버는 등록과 인증을 사용자와 직접 수행함으로써 별도의 분리된 등록 센터 등의 기관이 필요치 않으며, 서버 군에 속하는 모든 서버는 동일한 권한을 가진다. 또한 정당한 서버 군에 속하는 서버에 의해 생성된 인증 값을 포함한 로그인 정보를 가진 정당한 사용자는 어떤 서버와 등록 단계를 수행했는지와 관계없이 서버 군에 속하는 모든 서버와의 인증과정을 통과하게 된다. 두 개의 다른 정당한 인증 값에 대해 같은 서버에 의해 생성되었는지 아닌지를 결정하는 것은 계산적으로 어려우며, 인증 값으로부터 서버의 신원을 알 수 없어 서버에 대한 익명성을 보장하고, 각기 다른 서버를 통해 등록을 한 사용자에 대해서 모두 동등한 자격을 가지고 서버 군이 제공하는 모든 서비스를 이용할 수 있게 된다. 그러나 서버가 공모 공격에 가담하였거나 해당 서버 군의 서비스를 제공하기 위한 인증 정책을 지키지 않는 등 인증 값에 문제가 발생했을 경우 누가 인증 값을 생성했는지 추적이 가능하다. 따라서 제안된 인증 스킴은 서버 군의 일부 서버들이 공모한 경우나 심지어 전체 서버 군이 공모 그룹에 포함되어 있는 경우일지라도 추적이 불가능한 유효한 인증 값을 생성할 수 없도록 함으로써 공모 공격에 강하도록 설계되었다. 또한 서버 군

에 속하는 정당한 서버만이 그룹을 대신하여 서명을 생성할 수 있어 외부자 서버 위장 공격이 불가능하며, 서버 군에 속하는 정당한 서버나 심지어 서버 군의 그룹 관리자라 할지라도 다른 서버 군의 일원을 대신해서 서명할 수는 없다.

## Ⅲ. 제안된 스킴

본 장에서는 다중서버 환경을 위한 새로운 사용자 인증 스킴을 제안하고자 한다. 본 스킴에는 사용자, 서버 그리고 그룹 관리자 GM의 세 참여자와 (1) 초기화 (2) 등록 (3) 로그인 (4) 검증의 네 단계로 구성된다. 각각의 정당한 사용자는 서버 군에 속하는 다양한 서버 중에서 그가 등록하고자 하는 서버와의 단 한 번의 등록으로 다중서버 환경을 구성하는 모든 서버들이 제공하는 서비스를 제공받을 수 있게 된다. 제안된 스마트카드를 이용한 다중서버 인증 스킴의 특성은 다음과 같다.

1. **사용자 인증**: 사용자는 아이디, 패스워드와 서버로부터 발급된 스마트카드를 이용하여 인증 메시지를 생성하게 된다. 이 때, 사용자는 인증 메시지를 통해 서버와 사용자 인증과정을 수행한다.
2. **단일 등록**: 서버 군에 속하는 서버에 의해 생성된 인증 값을 포함한 로그인 정보를 가진 정당한 사용자는 어떤 서버와 등록 단계를 수행 했는 지와 관계없이 서버 군에 속하는 모든 서버에 대해 로그인 요청이 가능하다.
3. **사용자 익명성**: 서버는 로그인 메시지로부터 인증 값을 검증함으로써 사용자의 정당성을 검증하게 된다. 이때, 서버는 로그인 메시지로부터 사용자의 개인정보를 얻을 수는 없다.
4. **서버 익명성**: 두 개의 다른 정당한 인증 값에 대해 같은 서버에 의해 생성 되었는지를 결정하는 것은 계산적으로 어려우며, 인증 값으로부터 서버의 신원을 알 수 없다.
5. **공모 공격 저항성**: 제안된 인증 스킴은 서버 군의 일부 서버들이 공모하여 공격한 경우나 심지어 전체 서버 군이 공모하였을 경우라도 추적이 불가능한 유효한 인증 값을 생성할 수 없다.
6. **부패한 서버에 대한 자격취소**: 서버가 공모 공격에 가담하였거나 해당 서버 군의 서비스를 제공하기 위한 인증 정책을 지키지 않는 등 인증 값에 문제가 발생했을 경우 오픈 단계를 거쳐 자격 취소를 위한 서버 검증 단계를 수행한 후 정당한 서버에 영향을 미치지 않고 해당 서버의 자격을 취소할 수 있다.
7. **위조 불가능성**: 서버 군에 속하는 정당한 서버만이 그룹을 대신하여 서명을 생성할 수 있어 내부자/외부자 서버 위장 공격이 불가능하며, 서버 군에 속하는 정당한 서버나 심지어 그룹 관리자라 할지라도 다른 서버 군의 일원을 대신해서 서명할 수는 없다.

각 단계에 대한 자세한 사항은 다음과 같다.

<초기화 단계>

초기화 단계를 통하여 GM은 인증 스킴 구성을 위한 공개/비밀 파라미터를 설정하고 서버 군에 속하는 각 서버의 비밀 키를 생성한다.

1. 먼저 $G_2$에 대한 임의의 생성자를 선택하고, $G_2$에서 $G_1$으로 가는 동형함수(isomorphism) $\psi$에 대하여 $g_1 = \psi(g_2)$을 계산한다.
2. $G_1$에서 $1_{G_1}$이 아닌 임의의 원소 $h$을 선택하고, $Z_p^*$에서 $\xi_1, \xi_2, \gamma$을 선택한다.
3. $u^{\xi_1} = v^{\xi_2} = h$을 만족하는 $u, v \in G_1, w = g_2^{\gamma}$을 정한다.
4. $\gamma$을 이용하여, 서버 $S_\nu \in \Sigma$에 대하여 $x_\nu \in Z_p^*$와 $A_\nu = g_1^{1/(\gamma + x_\nu)} \in G_1$을 정한다.
5. $\Sigma$에 대하여 $Z_p$에서 임의의 원소 $t$을 선택한다.

이때 서버 군 $\Sigma$의 공개 키 $gpk$는 $(g_1, g_2, h, u, v, w)$가 되며, GM의 비밀 키 $gmsk$는 $(\xi_1, \xi_2)$가 된다. 또한 각각의 서버 $S_\nu$에 대한 비밀 키 $gsk[\nu], 1 \le \nu \le \eta$는 $(A_\nu, x_\nu)$이 되며, 서버 군에 속하는 모든 서버는 비밀 값 $t$을 공유하게 된다.

<등록 단계>

등록 단계를 통하여 $U_i$는 $ID_i$와 $pw_i$을 서버 $S_\nu$에 등록한다. $S_\nu$는 다음과 같은 인증 정보를 $U_i$스마트카드에 안전하게 저장한다. 사용자는 자신이 등록하기를 원하는 서버와 단 한 번의 등록 단계를 거침으로써 서버 군 $\Sigma$에 포함된 모든 서버들에 대해 다양한 서비스를 제공받을 수 있게 된다.

1. $U_i$는 안전한 채널을 통하여 $S_\nu$에게 $(ID_i, pw_i)$을 보낸다.
2. $(ID_i, pw_i)$을 전송받은 $S_\nu$는 $(A_\nu, x_\nu)$을 이용하여 $U_i$의 $ID_i$와 $pw_i$에 대한 인증 값 $Z_1$, $Z_2$ 그리고

$\sigma_i = (T_1, T_2, T_3, c_i, s_\alpha, s_\beta, s_{x_\nu}, s_{\delta_1}, s_{\delta_2})$을 계산한다.

(1) $Z_1 = t \oplus H(ID_i)$, $Z_2 = t \oplus H(pw_i)$

(2) $T_1 = u^\alpha$, $T_2 = v^\beta$, $T_3 = A_\nu \cdot h^{\alpha+\beta}$, $\alpha, \beta \in Z_p$.

(3) $R_1 = u^{r_\alpha}, R_2 = v^{r_\beta}, R_4 = T_1^{r_x} \cdot u^{-r_{\delta_1}}$, $R_3 = e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}$, $R_5 = T_2^{r_x} \cdot v^{-r_{\delta_2}}$, $r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \in Z_p$.

(4) $c_i = H(t, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$을 구한다.

(5) $s_\alpha = r_\alpha + c_i\alpha$, $s_\beta = r_\beta + c_i\beta$, $s_{x_\nu} = r_x + c_i x_\nu$, $s_{\delta_1} = r_{\delta_1} + c_i\delta_1$, $s_{\delta_2} = r_{\delta_2} + c_i\delta_2$.

3. $S_\nu$는 $U_i$의 스마트카드에 $Z_1$, $Z_2$, $\sigma_i$, $\triangle T$, $h_1, h_2$ 그리고 $H$을 저장하여 발급한다.

여기에서 각 서버는 등록 요청을 하는 사용자에 대한 인증 값 $\sigma_i$을 생성할 때, $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$의 값은 사용자와 관계없이 선행계산이 가능하며, 이를 통해 등록 단계의 효율성을 높일 수 있다. 더불어 $s_\alpha, s_\beta, s_{x_\nu}, s_{\delta_1}, s_{\delta_2}$의 계산은 환 군 $Z_p$에서의 덧셈과 곱셈 연산으로 구성되므로 인증 값을 생성하기 위한 계산량은 높지 않다고 할 수 있다.

<로그인 단계>

시스템에 로그인 하고자 할 때 사용자는 카드 리더기에 스마트카드를 넣은 후 자신의 패스워드를 입력한다. 서버 $S_\nu$와 등록 단계를 수행한 사용자는 등록한 서버 $S_\nu$뿐 아니라 $\Sigma$ 내에 있는 모든 서버에 대해 로그인 요청이 가능하다. $U_i$는 원하는 서비스를 제공할 수 있는 서버를 확인한 후, 해당 서버 $S_\theta \in \Sigma, 1 \le \theta \le \eta$와의 인증 절차를 수행한다.

1. $U_i$는 자신의 스마트카드를 리더기에 삽입하고 자신의 $pw_i$을 입력한다.
2. 스마트카드는 다음을 수행한다.
   (1) $Z_1 \oplus H(ID_i) = Z_2 \oplus H(pw_i) = t$인지를 체크하여 만족하지 않는다면 로그인은 거절된다.
   (2) 현재 타임스탬프 값 $T$을 이용하여 $h_1(T)^t$와 $h_2(T)^t$ 값을 계산한 후 다음의 계산을 수행한다. ① $\widetilde{T_1} = T_1 \cdot h_1(T)^t$, $\widetilde{T_2} = T_2 \cdot h_1(T)^t$ 그리고 $\widetilde{T_3} = T_3 \cdot h_1(T)^t$을 계산한다.
   ② $\widetilde{s_\alpha} = s_\alpha \cdot h_2(T)^t$, $\widetilde{s_\beta} = s_\beta \cdot h_2(T)^t$, $\widetilde{s_{x_\nu}} = s_{x_\nu} \cdot h_2(T)^t$, $\widetilde{s_{\delta_1}} = s_{\delta_1} \cdot h_2(T)^t$, $\widetilde{s_{\delta_2}} = s_{\delta_2} \cdot h_2(T)^t$, $\widetilde{c_i} = c_i \cdot h_2(T)^t$을 계산한다.
   ③ $mac$ 값을 다음과 같이 계산한다.
   $mac = H(T, \widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{c_i}, \widetilde{s_\alpha}, \widetilde{s_\beta}, \widetilde{s_{x_\nu}}, \widetilde{s_{\delta_1}}, \widetilde{s_{\delta_2}})$
   (3) 스마트카드는 로그인 메시지$(mac, \widetilde{c_i}, T, \widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{s_\alpha}, \widetilde{s_\beta}, \widetilde{s_{x_\nu}}, \widetilde{s_{\delta_1}}, \widetilde{s_{\delta_2}})$을 서버에게 전송한다.

<검증 단계>

서버 $S_\theta$는 사용자 $U_i$의 스마트카드로부터 로그인 메시지를 $T'$ 시간에 받은 후 다음과 같은 연산을 수행한다.

1. 전송 시 고려된 유예시간 $\triangle T$을 이용하여 현재 시간 $T'$와의 시간차를 확인하여 $|T - T'| \ge \triangle T$이면, 로그인 메시지를 거절한다. 만약 $|T - T'| < \triangle T$이면 다음 단계를 수행한다.
2. $mac = H(T, \widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{c_i}, \widetilde{s_\alpha}, \widetilde{s_\beta}, \widetilde{s_{x_\nu}}, \widetilde{s_{\delta_1}}, \widetilde{s_{\delta_2}})$을 만족하는지를 검증한다. 만족하지 않는다면 로그인 요청을 거절하고, 만족한다면 다음과정을 수행하다.
3. $S_\theta$는 $h_1(T)^{-t}$와 $h_2(T)^{-t}$을 계산한 후 로그인 메시지 $(mac, \widetilde{c_i}, T, \widetilde{T_1}, \widetilde{T_2}, \widetilde{T_3}, \widetilde{s_\alpha}, \widetilde{s_\beta}, \widetilde{s_{x_\nu}}, \widetilde{s_{\delta_1}}, \widetilde{s_{\delta_2}})$을 이용하여 다음의 검증을 수행한다.
   (1) $T_1' = \widetilde{T_1} \cdot h_1(T)^{-t}$, $T_2' = \widetilde{T_2} \cdot h_1(T)^{-t}$, $T_3' = \widetilde{T_3} \cdot h_1(T)^{-t}$, $s_\alpha' = \widetilde{s_\alpha} \cdot h_2(T)^{-t}$, $s_\beta' = \widetilde{s_\beta} \cdot h_2(T)^{-t}$, $s_{x_\nu}' = \widetilde{s_{x_\nu}} \cdot h_2(T)^{-t}$, $s_{\delta_1}' = \widetilde{s_{\delta_1}} \cdot h_2(T)^{-t}$, $s_{\delta_2}' = \widetilde{s_{\delta_2}} \cdot h_2(T)^{-t}$, $c_i' = \widetilde{c_i} \cdot h_2(T)^{-t}$을 계산한다.
   (2) $gpk = (g_1, g_2, h, u, v, w)$을 이용하여 다음을 구한다.
   $R_1' = u^{s_\alpha'} \cdot T_1'^{-c_i'}$, $R_2' = v^{s_\beta'} \cdot T_2'^{-c_i'}$,
   $R_3' = e(T_3', g_2)^{s_{x_\nu}'} \cdot e(h, w)^{-s_\alpha' - s_\beta'} \cdot e(h, g_2)^{-s_{\delta_1}' - s_{\delta_2}'} \cdot (e(T_3', w)/e(g_1, g_2))^{c_i'}$
   $R_4' = T_1'^{s_x'} \cdot u^{-s_{\delta_1}'}$, $R_5' = T_2'^{s_x'} \cdot v^{-s_{\delta_2}'}$
   (3) $c_i'$을 (2)에서 구한 값을 이용하여 $H(t, T_1', T_2', T_3', R_1', R_2', R_3', R_4', R_5')$와 같은 지를 체크하여 같지 않다면 로그인은 거절된다.
4. 검증을 모두 만족하게 되면 서버는 현재 타임스탬프 값 $T''$을 이용하여 $A = h_1(t, T'')^{t+1}$을 계산하여 $(T'', A)$을 스마트카드에 전송한다.
5. 스마트카드는 $T''$와 현재시간 $T'''$와의 시간차를 확인하여 $|T'' - T'''| < \triangle T$이면 다음 단계를 수행한다.
6. $h_1(t, T'')^{t+1} = A$인지를 검증하여 성립하면, 스마트카드는 서버와의 인증에 성공한 것으로 인식한다.

<오픈 단계>

각 서버 중에서 잘못된 인증을 수행하거나 공모 공격에 가담한 부패한 서버에 대해서는 오픈 단계를 통해 자격 취소를 위한 서버 검증 단계를 수행하게 된다. 이는 일상적인 인증 스킴에서는 제외되고, 서버에 관련된 문제가 발생하였을 경우 수행하게 된다. 이 단계는 서버 군 공개 키 $gpk$와 해당 서버 군의 그룹 관리자(GM)의 비밀 키 $gmsk$을 사용하며, 정당한 절차를 통하여 특정 서버에 등록한 후 추후 로그인 요청을 거절당하는 등의 문제가 발생했을 경우 사용자의 아이디 $ID_i$와 패스워드 $pw_i$ 그리고 이에 대한 인증 값 중 하나인 $\sigma_i$을 이용해 문제가 된 서버의 신원을 추적하게 된다.

1. 먼저 $\sigma_i$가 $ID_i$와 $pw_i$에 대한 정당한 서명인지를 확인한다.
2. $A_\nu = T_3/(T_1^{\xi_1} \bullet T_2^{\xi_2})$을 계산하여, $A_\nu$을 이용하여 서버의 신원을 확인한다.

## Ⅳ. 분 석

이 장에서는 제안된 스킴의 안전성과 효율성을 분석한다.

### 4.1 안전성 분석

안전성의 기본요소는 Fan et al.[6]에서 분류된 조건 중 제안된 인증 스킴에서 요구되는 것과 그 외의 중요한 안전성 요소를 모두 고려하였다. 제안된 스킴은 가장 공격, 오프라인 패스워드 공격 그리고 재사용 공격에 대해서 안전하고 전방향 안전성 성질을 만족한다. 또한 외부 공격자와 악의적인 서버에 대한 사용자의 익명성과 내·외부 공격자에 대한 서버의 익명성을 제공하며, 다중서버 환경에서 매우 중요한 안전성 요소인 공모 공격에 대해 강인성을 갖는다. 이때 본 논문에서는 스마트카드의 temper-resistant 성질을 기반으로 공격자가 사용자의 스마트카드를 오염시키거나 스마트카드로부터 정보를 얻는 공격은 고려하지 않는다. 그러나 공격자는 네트워크의 정보를 모두 얻을 수 있는 것으로 가정한다. 다음은 안전성에 대한 자세한 설명이다.

#### 4.1.1 은밀한 검증자 공격(stolen-verifier attack)

제안된 프로토콜은 검증 테이블을 필요로 하지 않는다. 따라서 어느 누구도 서버로부터 검증할 수 있는 정보를 얻을 수는 없다. 그러므로 제안된 스킴은 은밀한 검증자 공격에 대해서 안전하다.

#### 4.1.2 사용자 가장 공격(user-impersonation attack)

로그인과 검증 단계에서 획득한 메시지를 이용하여 사용자 가장 공격이 가능하기 위해서는 서버 군의 정당한 일원인 서버만이 생성 가능한 서버 군 서명 값들에 대한 정보를 획득하는 것이 가능해야만 한다. 그러나 본 논문에서 제안한 인증 방법은 공격자가 로그인 메시지에 포함된 정보의 타임스탬프 값들 $T, T''$에 대한 해쉬 값은 얻을 수 있지만 정당한 그룹의 멤버인 서버들만이 공유하고 있는 비밀 값 $t$에 대한 $h_1(T)^t$ 와 $h_2(T)^t$ 값을 계산하는 것이 불가능하며, $h_1(T)^t$와 $h_2(T)^t$ 값에 대한 정보가 주어졌을 때 $t$을 구하는 문제는 DLP(Discrete Logarithm Problem)의 어려움에 기반을 두고 있다.

#### 4.1.3 서버 가장 공격(server-impersonation attack)

정당한 그룹의 멤버인 서버가 아니라면 주어진 답변 메시지로부터 $h_1(t, T'')^{t+1}$ 값을 이용하여 $t$ 또는 $t+1$ 값을 획득하는 것은 불가능하다. $h_1(t, T'')^{t+1}$ 값이 주어졌을 때 $t$을 구하는 문제는 DLP의 어려움에 기반을 두고 있으며, $h_1(t, T'')$ 값도 알 수 없기 때문에 이산대수문제보다 어려운 문제라고 볼 수 있다. 따라서 서버의 정당한 메시지 없이 임의의 시간 $T^*$에 $h_1(t, T^*)^{t+1}$ 값을 대답하는 것은 $t$에 대한 정보 없이는 불가능하다.

#### 4.1.4 오프라인 패스워드 공격(offline password attack)

정당한 사용자의 로그인 메시지는 사용자의 패스워드 정보를 포함하지 않는다. 단지, 로그인 메시지 생성 단계에서 자신의 아이디와 패스워드 정보 그리고 스마트카드를 이용하여 정당한 로그인 메시지를 생성하게 된다. 따라서 공격자는 로그인 메시지로부터 사용자의 패스워드 정보를 얻을 수 없다.

#### 4.1.5 사용자의 익명성(user anonymity)

정당한 사용자의 로그인 메시지는 사용자의 패스워드 정보를 포함하지 않고 로그인 단계에서 서버로부터 부여받은 $mac$ 값에 대한 정당성에 대한 검증만을 수행한다. 따라서 로그인 메시지는 사용자의 아이디나 패스워드 정보를 포함하지 않기 때문에 서버와 외부 공격자 모두에게 사용자의 익명성을 제공하게 된다.

#### 4.1.6 서버의 익명성(server anonymity)

모든 단계에서 내·외부 공격자가 알 수 있는 서버에 대한 정보를 포함하고 있는 메시지는 $\sigma_i$에서 $s_{x_v}$가 유일하며, 이로부터 서버의 신원을 확인하는 것은 어려운 일이다. 더욱이 추가 과정(오픈 단계) 없이 사용자가 등록한 서버가 어느 서버인지를 확인하는 것은 불가능하다.

#### 4.1.7 재사용 공격(replay attack)

정당한 사용자에 의해서 생성된 로그인 메시지를 그대로 이용할 경우 인증 단계에서 검증하는 인증 시간과 인증 메시지에 포함된 시간과의 차이를 검증하는 타임스탬프 체크를 이용하는 방법에 의해서 본 논문에서 제안된 스킴은 재사용 공격에 대한 안전성을 제공하게 된다.

#### 4.1.8 공모 공격(collusion attack)

서버가 공모 공격에 가담하였거나 서비스를 제공하기 위한 인증 정책을 지키지 않는 등 인증 값에 문제가 발생하였을 경우 SDH(Strong Diffie-Hellman) 가정 하에 인증 값 $\sigma_i$을 통하여 누가 인증 값을 생성했는지 추적이 가능하다.

### 4.2 효율성 분석

본 논문에서 제안된 스킴은 스마트카드를 기반으로 구성되었기 때문에 계산상의 효율성 또한 중요하다고 할 수 있다. 이전에 제안된 스마트카드를 이용한 익명 인증 스킴 중 최근 결과인 Das et al.[4], Yoon et al.[3] 그리고 Chien et al.[7]의 스킴과 다중서버 인증 스킴인 Lin et al.[1]와 W. Juang[5]와의 비교를 통해서 효율성을 분석하고자 한다.

제안된 스킴에서 인증 값 $\sigma_i = (T_1, T_2, T_3, c_i, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$은 $G_1$의 세 원소와 여섯 개의 $Z_p$ 원소로 구성된다. $p$을 SDH에 기반한 안전성을 제공할 수 있는 170비트 소수라 하고 각 원소가 171비트인 $G_1$ 그룹을 사용한다면, $\sigma_i$의 전체 길이는 1533비트가 된다. 따라서 스마트카드에 인증 값을 저장하기 위해 필요한 메모리는 다중서버 환경을 구성하는 서버 수 $k$와 관계없이 약 1900비트가 된다. 이는 서버 익명성과 사용자 공모 공격 불가능성 등 안전성 요소를 추가하면서도 Lin et al.[1] 인증 스킴이 이산대수문제에 기반하여 1024비트 소수를 사용한다고 가정할 때 필요한 $(4k+1) \cdot 1024$비트보다 매우 효율적임을 알 수 있다. 그러나 이는 W. Juang[5]의 256비트보다는 더 많은 메모리를 요구하는데 W. Juang의 스킴은 등록 센터에 대한 의존도가 매우 높고, 등록을 위해 별도의 기관 등록 센터가 필요하며, 모든 사용자는 등록 단계를 등록 센터와 거치게 된다. 따라서 스마트카드의 저장용량이 적어도 8Kbyte~256Kbyte인 점을 감안하면, 중앙 관리자의 의존도를 낮추면서 스마트카드에 1900비트의 저장량은 타당하다고 할 수 있다.

본 스킴은 쌍일차 함수 $e(h,w), e(h,g_2)$ 그리고 $e(g_1,g_2)$는 먼저 선행계산이 가능하다. 각 서버는 $e(A_v, g_2)$을 저장할 수 있으며, 인증 값 생성 시 쌍일차 함수 값을 구하지 않고 $e(T_3, g_2)$을 계산한다. 따라서 등록 단계에서 인증 값을 생성하는 데는 쌍일차 함수 계산 없이 여덟 번의 지수연산(exponentiation)만을 수행하면 된다. 그러나 $T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5$의 값은 사용자와 관계없이 사용자의 등록 요청을 받기 전에 선행계산이 가능하며, 이를 통해 등록 단계의 효율성을 높일 수 있다. 더불어 $s_\alpha, s_\beta, s_{x_v}, s_{\delta_1}, s_{\delta_2}$의 계산은 환 군 $Z_p$에서의 덧셈과 곱셈 연산으로 구성되므로 인증 값을 생성하기 위한 계산량은 높지 않다고 할 수 있다. 로그인 요청을 받은 서버는 $e(T_3, g_2)^{s_x}$와 $e(T_3, w)^c$을 $e(T_3, w^c \cdot g_2^{s_x})$ 하나로 정리하여 $R_3$을 효율적으로 얻어낼 수 있다. 따라서 로그인 요청에 대해 검증을 수행하는 서버는 여덟 개의 지수연산을 수행하면 된다. 이는 다중서버 환경을 구성하는 서버 수에 관계없이 고정적이다. Lin et al.[1] 인증 스킴의 경우 $k$개의 서버로 구성된 다중서버 환경에서 등록을 위해서는 $5k$번의 지수연산이 필요하고, 인증과정에서는 아홉 번의 지수연산을 요구한다.

본 스킴의 기능과 효율성에 대한 내용은 표 1에 요약하였으며, 이를 통해 알 수 있듯이 제안된 스킴은 등록, 로그인 그리고 검증 단계에서 비교적 효율적이며, 더불어 등록을 수행하는 서버에 대해서는 지수연산을 위한 기저를 고정시키면 선행계산을 통해 좀 더 계산의 효율성을 높일 수 있다.

## V. 결 론

본 논문에서 제안된 스킴은 스마트카드를 이용한 다중서버 환경을 위한 상호 익명성을 제공하는 최초의 인증 프로토콜로 안전하고 효율적인 인증방법을 제공한다는 측면에서 그 의미가 크다고 할 수 있다. 서버 군에 속하는 모든 서버는 등록과 인증을 사용자와 직접 수행함으로써 별도의 분리된 등록

표 1. 기능 및 효율성 분석

| 프로토콜 | 계산량 | | | | 상호 익명성 | 상호 인증 | 단일 등록 | 등록 기관의 필요성 |
|---|---|---|---|---|---|---|---|---|
| | 등록 | 로그인 | 인증 | 총계 | | | | |
| Our scheme | 2H | 3H+2E | 2H+8E | 7H+10E | **Yes** | **Yes** | **Yes** | **No** |
| Das et al. schem[4] | 2kH | 5H | 3H | (2k+8)H | No | No | No | **No** |
| Yoon et al. scheme[3] | 2kH+kE | 5H+1E | 4H+3E | (2k+9)H+(k+4)E | No | No | No | **No** |
| Chien et al. scheme[7] | 2kH | 1H+1E+1S | 3H+2E+2S | (2k+4)H+3E+3S | No | **Yes** | No | **No** |
| Lin et al. scheme[1] | 5kE | 4E | 9E | (5k+13)E | No | No | **Yes** | Yes |
| W. Juang's scheme[5] | 1H | 4H+7S | 3H+5S | 8H+12S | No | **Yes** | **Yes** | Yes |

H: 해쉬 함수의 계산량, E: 지수계산
S: 대칭키 암호/복호, k: 서버 군을 이루는 서버의 개수

센터 등의 기관이 필요치 않으며, 서버 군에 속하는 모든 서버는 동일한 권한을 가진다. 또한 정당한 서버 군에 속하는 서버에 의해 생성된 인증 값을 포함한 로그인 정보를 가진 정당한 사용자는 어떤 서버와 등록 단계를 수행했는지와 관계없이 서버 군에 속하는 모든 서버와의 인증과정을 통과하게 된다. 사용자는 한 번의 등록으로 서버 군에서 제공하는 다양한 서비스를 제공받을 수 있으며, 이때 통신하는 모든 메시지로부터 사용자의 아이디나 패스워드같은 개인정보와 서버에 대한 정보를 얻을 수 없게 되어 사용자는 외부 공격자뿐만 아니라 서버에 대해서 그리고 서버는 내·외부 공격자에 대해 타당한 익명성을 제공받게 된다. 본 스킴은 타당한 효율성을 유지하면서 다중서버 환경에 필요한 기능과 안전성을 추가하였다.

## 참 고 문 헌


〔1〕 I. Lin, M. Hwang, L. Li, "A New Remote User Authentication Scheme for Multi-server Architecture," *Future Generation Computer Systems*, 19, pp.13-22, 2003.

〔2〕 D. Boneh, X. Boyen, H. Shacham, "Short Group Signatures," *Advances in Cryptology, Crypto 2004*, 3152, pp.41-55, 2004.

〔3〕 E. Yoon, E. Ryu, K. Yoo, "Efficient Remote User Authentication Scheme based on Generalized ElGamal Signature Scheme," *IEEE Transactions on Consumer Electronics*, 50, pp.568-570, 2004.

〔4〕 M. Das, X. Saxena, V. Gulati, "A Dynamic ID-based Remote User Authentication Scheme," *IEEE Transactions on Consumer Electronics*, 50, pp.629-631, 2004.

〔5〕 W. Juang, "Efficient Multi-server Password Authenticated Key Agreement Using Smart Cards," *IEEE Transactions on Consumer Electronics*, 50, pp.251-255, 2004.

〔6〕 C. Fan, Y. Chan, Z. Zhang, "Robust Remote Authentication Scheme with Smart Cards," *Computers and Security 2005*, 24, pp.619-628, 2005.

〔7〕 H. Chien, C. Chen, "A Remote Authentication Scheme Preserving User Anonymity," *IEEE AINA'05*, 2, pp.245-248, 2005.



**유 혜 정** (Hye-joung Yoo) 정회원

1999년 2월 고려대학교 수학과 졸업
1999년 2월 고려대학교 수학과 석사
2002년 8월 고려대학교 수학과 박사
2004년 1월~현재 세종사이버대학교 정보보호시스템학과 조교수
<관심분야> 암호프로토콜, 컨텐츠보안