情報保護學會論文誌
第13券 第1號, 2003. 2

# 다중서버를 이용한 인증된 키교환 프로토콜

이 정 현*, 김 현 정*, 이 동 훈*

# Multi-Server Authenticated Key Exchange Protocol

Jung-hyun Lee*, Hyun-jeong Kim*, Dong hoon Lee*

요 약

본 논문에서는 사용자가 서버를 신뢰하지 않아도 되는 인증프로토콜을 설계하기 위하여 인증서버에 위탁되는 인증정보에 대한 두 가지 안전성인 "계산 불가능 안전성"과 "분산 안전성"을 정의한다. 또한 분산 안전성에 기반 하여 서버를 신뢰하지 않아도 되는 패스워드 기반 인증프로토콜인 MAP(Multiplex Server Authentication Protocol)을 제시하고, 이러한 MAP을 이용한 SSSO(Secure Single Sign On)가 서버를 신뢰하여야 하는 인증프로토콜을 이용한 SSO(Single Sign On)의 문제점을 해결함을 보인다.

ABSTRACT

In this paper, we define two security concepts, "non-computable security" and "distribution security", about authentication information committed to a authentication server without any trustee, and propose an authenticatied key exchange protocol based on password, satisfying "distribution security". We call it MAP(Muti-Server Authentication Protocol based on Password) and show that SSSO(Secure Single Sign On) using MAP solves a problem of SSO(Single Sign On) using authentication protocol based on password with a trustee.

**Keyword** : *Password, Trustee, SSO, Key Exchange Protocol, Authentication*

## Ⅰ. 서 론

구현된 인증 프로토콜의 안전성은 2가지 측면에서 고려해야 한다. 하나는 프로토콜의 인증과정에서의 안전성이고 다른 하나는 구현된 프로토콜의 인증서버에 대한 안전성을 고려할 수 있다. 전자는 사용자와 인증서버 사이에 프로토콜 진행과정 중 전달되는 메시지에 대한 프로토콜의 안전성이고, 후자는 인증과정에 필요한 사용자 인증정보에 대한 안전성이다. 기존의 패스워드 기반 인증프로토콜의 안전성 연구는 후자에 대해서는 안전하다고 가정하고, 전자에 대해서만 언급해 왔다. 하지만 이러한 가정은 인증서버를 사용자가 신뢰한다는 가정 하에서 시작된다.

본 논문에서는 후자에 대한 안전성을 고려한 패스워드 기반 인증 프로토콜을 제시한다. 후자에 대한 안전성은 인증서버에 대한 외부공격자와 내부공격자 두 종류의 공격자에 대한 안전성을 고려해야 한다. 외부공격자에 대한 안전성은 인증시스템 밖에서부터 인증 시스템에 침입해 인증서버에 저장된 사용자 인증정보를 알아내려는 공격자이고, 내부공격자는 인증서버에 저장된 인증정보를 가지고 사용자를 위장하려는 공격자이다. 외부 공격자에 대한 안전성은 프로토콜에 대한 안전성측면에서 고려해야 할 사항이 아닌 시스템에 대한 안전성에서 고려해야 할 사항이다. 따라서 본 논문에서는 외부공격자에 대한 안전성은 안전하다고 가정

* 고려대학교 정보보호기술연구센터(CIST)({moomoo46, khj}@cist.korea.ac.kr, donghlee@korea.ac.kr)

하고, 내부 공격자에 대하여 안전한 인증프로토콜을 제시한다.

인증프로토콜에서 인증서버는 사용자에게서 위탁받은 사용자 인증정보를 통해 인증과정을 수행한다. 이러한 인증프로토콜에서 사용자가 위탁하는 인증정보에 대한 안전성은 매우 중요하게 고려되어야 한다. 특히 패스워드 기반 인증프로토콜에서의 사용자는 자신이 기억하는 패스워드에 대한 인증정보를 인증서버에 위탁해 인증과정을 수행한다. 그러나 패스워드는 사용자가 기억할 수 있는 정보이어야 하기 때문에 비밀키에 비해 상대적으로 엔트로피가 매우 낮다는 문제점을 가지고 있다. 따라서 패스워드 기반 인증프로토콜의 경우 패스워드에 대한 인증정보 위탁은 더욱 더 안전하게 고려되어야 한다. 인증프로토콜에서 사용자가 서버와 인증과정에서 사용할 인증정보를 위탁할 때, 안전하지 않은 인증정보를 위탁한다면 사용자는 서버를 신뢰하여야 한다. 즉 내부 공격자에 대해 안전하지 않는 인증프로토콜이 된다.

본 논문에서는 내부 공격자에 대해 안전성을 재고하기 위해 사용자가 위탁하는 인증정보를 안전하게 생성하기 위한 두 가지 안전성 개념인 "계산 불가능 안전성" 과 "분산 안전성"을 정의한다. 특히 "분산 안전성"에 기반하여 내부 공격자에 대해 안전한 인증정보를 생성하는 패스워드 기반 분산 인증프로토콜인 *MAP*(Multi Server Authenticated Key Exchange Protocol)을 제시한다. 또한 서버를 신뢰하지 않아도 되는 *MAP*을 이용한 *SSSO*(Secure Single Sign On)가 서버를 신뢰하여야 하는 패스워드기반 인증프로토콜을 이용한 *SSO*(Single Sign On)의 문제점을 해결함을 보인다.

2절에서는 지금까지의 패스워드 기반 프로토콜의 연구를 설명하고, 3절에서는 서버를 신뢰하지 않아도 되는 패스워드 기반 인증프로토콜을 설계하기 위한 인증정보에 대한 두 가지 안전성 개념인 "계산 불가능 안전성"과 "분산 안전성"을 정의한다. 또한 4절에서는 *CDL*(Credential Down Load) 프로토콜을 이용하여 서버를 신뢰하지 않아도 되는 패스워드 기반 인증된 키교환 프로토콜을 설명한다. 그리고 5절에서는 서버를 신뢰하지 않을 뿐만 아니라 *CDL*프로토콜을 이용한 서버를 신뢰하지 않아도 되는 인증된 키교환 프로토콜보다 효율적인 *MAP*을 제시하고, 6절에서는 *MAP*의 안전성에 대하여 설명한다. 또한, 7절에서는 기존의 패스워드 기반 인증프로토콜을 이용한 *SSO*의 문제점을 지적하고, *MAP*을 이용한 *SSSO*가 이러한 문제점을 해결함을 보인다.

## II 기존 연구

패스워드 기반 인증프로토콜이란 어떤 서버에게 인증을 받으려는 사용자가 스마트카드와 같은 추가적인 장비 없이 오직 자신이 기억하고 있는 한 개의 패스워드(long term secret)만을 가지고 인증 받을 수 있도록 하는 프로토콜[1]이다. 이러한 편리성 때문에 패스워드 기반 프로토콜은 이동 사용자 서비스(key roaming service)등 각 분야에 널리 사용되고 있다. 지금까지 패스워드 기반 프로토콜은 크게 인증된 키교환 프로토콜과 *CDL*(Credential Down Load) 프로토콜 두 분야로 연구되고 있다[2].

### 2.1 패스워드 기반 인증된 키교환 프로토콜

이 카테고리에 속한 프로토콜은 사용자가 패스워드만을 이용하여 서버와 서로 인증된 키(세션키)를 교환·공유하는 프로토콜이다. 이러한 패스워드 기반 인증된 키교환 프로토콜은 대칭구조와 비대칭구조 프로토콜로 분류될 수 있다[3]. 대칭구조 프로토콜은 사용자와 서버가 각각 같은 값을 이용하여 서로 정당한 사용자 그리고 정당한 서버라는 것을 증명하는 프로토콜이고, 비대칭구조 프로토콜은 사용자와 서버가 각각 다른 값을 이용하여 서로 정당한 사용자 그리고 정당한 서버라는 것을 증명하는 프로토콜이다. 대칭구조의 경우 서버가 저장하고 있는 사용자 인증정보 파일(패스워드 파일)을 공격자가 얻게 되면 그 공격자는 항상 사용자 위장공격과 서버 위장공격을 할 수 있다. 이러한 대칭구조의 패스워드 기반 인증된 키교환 프로토콜은 *EKE*[4], *SPEKE*[1]등이 있다.

이러한 문제점을 개선하기 위해 제시된 프로토콜이 비대칭구조 프로토콜이다. 비대칭구조 프로토콜은 서버가 저장하는 사용자 인증정보 파일(패스워드 검증 파일)을 공격자가 얻더라도 공격자는 곧바로 사용자 위장공격을 할 수 없다. 하지만 여전히 공격자는 오프라인 사전공격을 통해 사용자를 위장할 수 있다. 결국 비대칭구조에서 서버의 사용자 인증정보 파일이 공격자에게 알려졌을 경우 공격자는 서버를 항상 위장할 수 있는 점에서 대칭 구조와 같지만 사용자를 위장하기 위해 먼저 사전공격을 수행해야 한다는 점에서 대칭구조보다 좀 더 안전성을 확보할 수 있다. 이러한 비대칭 구조 프로토콜의 예는 *A-EKE*[5], *B-SPEKE*[6], *SRP*[7], *AuthA*[8], *OKE*[9] 등이 있다.

비대칭 구조에 대한 또 다른 기법은 서버의 비밀

키를 이용하는 *AMP*[3]를 들 수 있다. *AMP*는 서버에서 안전하게 관리되는 서버 비밀키를 사용자 인증정보를 생성할 때 이용하여 사용자 인증정보 파일이 공격자에게 알려져도 사전공격 및 서버 위장공격이 이루어질 수 없도록 하는 구조이다. 하지만 *AMP*는 서버위장 공격과 사용자 인증정보 파일에 대한 사전공격을 막는 실용적인 프로토콜이라 할 수 없다. 그 이유는 *AMP*가 주장하는 안전성을 확보하기 위해 서버의 비밀키는 서버와 분리된 보호모듈 안에만 저장되어 있어야 하기 때문에 서버의 사용자 인증연산이 분리된 보호모듈 내에서 수행되어야 한다. 이는 연산 속도에 영향을 미치며 결과적으로 사용자가 많은 시스템에서는 *AMP* 저자가 지적한 것처럼 프로토콜 진행 과정에서 병목현상이 발생하게 된다. 이러한 병목현상을 막기 위해서는 서버의 사용자 인증연산 과정에서 서버 비밀키가 분리된 보호모듈 밖으로 나올 수 밖에 없고 이러한 경우 *AMP*의 안전성은 결국 다른 비대칭 패스워드 기반 인증된 키교환 프로토콜과 동일해 진다.

또한 위에서 언급한 모든 패스워드 기반의 인증된 키교환 프로토콜은 그것이 대칭구조이든 비대칭구조이든 또는 서버 비밀키를 이용하든 패스워드에 대한 일방향 함수 값은 사전공격에 대해 취약하기 때문에 내부공격자에 대해 안전하지 않다. 즉 사용자는 서버를 신뢰하여야 한다.

### 2.2 *CDL*(Credential Down Load) 프로토콜

지금까지 패스워드 기반 프로토콜은 사용자 인증 후 세션키를 교환하는 프로토콜이 대부분이었다. 하지만 최근 또 다른 형태의 패스워드 기반 프로토콜인 *CDL*프로토콜이 제안되었다. 사용자는 패스워드를 이용해서 사전에 키를 위탁해 놓은 키위탁 서버로부터 엔트로피가 높은 비밀키를 내려 받은 후 이 키를 이용해서 응용서버와 사용자 인증을 수행한다[2].

이러한 *CDL*프로토콜 중 주목할 만한 것은 Ford-Kaliski방식[10]과 Jablon방식[11]이다. 이 두 가지 방식은 패스워드의 약점(낮은 엔트로피문제)을 보완하기 위한 방법으로 제안되었다. 즉, 위에서 지적한대로 서버의 사용자 인증정보가 노출되면 항상 사용자를 위장할 수 있다. 이러한 서버의 사용자 인증정보 노출문제를 해결하기 위해 위의 두 가지 방법에서는 사용자 인증정보를 다중서버에 분산하여 저장한다. 따라서 인증에 참여하는 $n$개의 서버 중 $n-1$개의 사용자 인증정보가 공격자에게 노출되어도 공격자는 사용자의 패스워드를 알 수 없다.

## Ⅲ. 인증정보의 안전성

기존 대부분의 패스워드 기반 인증프로토콜의 경우, 안전성을 위해 사용자가 인증서버에게 위탁하는 인증정보를 관리하는 서버를 사용자가 신뢰하여야 한다는 조건이 요구된다. 즉 내부공격자에 대해 사용자 인증정보는 매우 취약하다는 약점을 가지고 있다. 이처럼 내부 공격자가 존재하는 경우에도 안전한 인증프로토콜을 설계하기 위한 방법으로 "계산 불가능 안전성"과 "분산 안전성"이라는 두 가지 안전성에 기반하는 인증된 키교환 프로토콜을 설계할 수 있다. 이 두 가지 안전성에 대해 설명하면 다음과 같다.

먼저 "계산 불가능 안전성"의 경우를 살펴보면, 이는 서버의 계산 능력에 기반하는 안전성이라 볼 수 있다. 어떤 패스워드 기반 인증프로토콜에서 서버에 저장된 사용자 인증정보로부터 다항식 시간 내부공격자가 사용자를 위장할 수 있는 정보(패스워드)를 찾아낼 수 있다면, 이 프로토콜에서는 사용자 위장공격에 대한 안전성을 위해 사용자는 서버를 신뢰하여야 한다. 하지만 다항식 시간 내부공격자가 서버에 저장된 인증정보로부터 사용자 위장정보를 계산할 수 없을 경우 서버에 대한 신뢰성이 요구되지 않으며 이 프로토콜은 내부 공격자에 대해 안전한 프로토콜이라 할 수 있다. 이와 같은 계산 불가능 안전성에 대해 정의하면 다음과 같다.

**[정의1] (계산 불가능 안전성)**

인증프로토콜 $T$가 주어졌을 때, $T$의 임의의 내부공격자 $W$의 계산 능력을 $O(\cdot)$라 하고, $T$상에서 사용자 $U_i$의 비밀정보($\delta$)를 인증하기 위해 저장하고 있는 인증정보를 $I_i(\delta)$라 하자. 이때 모든 $i$에 대해 다음을 만족하면 $T$는 내부 공격자에 대해 "계산 불가능 안전성"에 기반하여 서버를 신뢰하지 않아도 되는 인증프로토콜이라 한다.

$$O(\cdot) < O(I_i)$$

여기서 $O(I_i)$는 $I_i(\delta)$로부터 $\delta$를 계산하기 위해 요구되는 계산 복잡도이다.

즉, 임의의 내부공격자의 계산능력에 비해 사용자의

인증정보로부터 사용자 위장공격을 위해 필요한 정보를 구하는 계산량이 더 크도록 프로토콜을 구현한다면 이는 신뢰성이 없는 서버에 대해서도 안전한 프로토콜이라 할 수 있다. 본 논문에서는 다항식 시간 내 부공격자에 대해 안전하면 사용자가 서버를 신뢰하지 않아도 되는 인증프로토콜이라 정의한다.

다음으로 "분산 안전성"에 대해 살펴보도록 한다. 이것은 내부 공격자로부터 사용자 인증정보에 대한 안전성을 보장하기 위해 사용자 인증정보를 다중서버에 분산 저장하는 방식에 대한 안전성 요구사항이다. 모든 분산된 인증정보가 모일 경우에만 사용자 인증정보가 복원될 수 있도록 함으로써 사용자는 각 서버를 신뢰할 필요가 없게 된다. 분산 안전성을 만족하는 사용자 인증정보 분산은 다음과 같이 정의할 수 있다.

**[정의2] (분산 안전성)**

인증 프로토콜 $T$가 주어졌을 때, 모든 자연수 $n$에 대해 $T$의 임의의 분산된 내부공격자의 집합을 $W=\{w_1,\cdots,w_n\}$라 하고 $T$상에서 각 사용자 $U_i$의 비밀정보 $\delta$에 대한 인증정보를 $I_i(\delta)$라 하자. 모든 내부공격자 $w_j$는 각 사용자 $U_i$의 인증정보 $I_i(\delta)$에 대해 사용자의 분산 인증정보 $I_i^j(\delta)$ 만을 저장하고 있다. 이때 다음 (1), (2) 조건이 성립하면 $T$는 내부 공격자에 대해 "분산 안전성"에 기반하여 신뢰성이 존재하지 않는 인증프로토콜이라 한다.

(1) 각 사용자 $U_i$에 대해 내부공격자가 저장하고 있는 사용자의 모든 분산 인증정보 $I_i^j(\delta)$를 입력값으로 하면서 다음을 만족하는 다항식 시간 알고리즘 $\Omega$가 존재한다.

$$\Omega(\bigcup_{j=1}^{n} I_i^j(\delta)) = I_i(\delta)$$

(2) 내부 공격자중 일부가 공모한 공모집단을 $C=\{w_{j1}, w_{j2},\cdots, w_{jc}\}$, $(c\le n-1)$라 하자. 이때 공모집단이 저장하고 있는 사용자 부분 인증정보를 입력값으로 하여 사용자 인증정보를 찾는 알고리즘 $A$가 존재할 때 임의의 난수를 입력값으로 하면서 다음을 만족하는 알고리즘 $B$가 존재한다.

$$\Pr[I_i'(\delta)=I_i(\delta) \mid I_i'(\delta)=A(\bigcup_{w_i\in C} I_i^j(\delta))]$$
$$=\Pr[I_i'(\delta)=I_i(\delta) \mid I_i'(\delta)=B(\cdot)]$$

위의 정의에서 (1)번이 의미하는 것은 항상 모든 부분 인증정보가 모였을 때만, 쉽게 사용자 인증정보 $I_i(\delta)$를 찾아낼 수 있음을 의미한다. (2)번의 경우는 $c(\le n-1)$명의 각 인증서버의 내부 공격자들이 공모를 통해 얻은 $c$개의 사용자 부분 인증정보를 이용하여 사용자 인증정보 $I_i(\delta)$를 계산할 확률은 사용자의 부분 인증정보를 전혀 알지 못하는 공격자가 사용자 인증정보를 계산해내는 확률과 동일해야 함을 의미한다.

## Ⅳ. 계산불가능 안전성에 기반한 인증된 키교환 프로토콜.

*CDL*프로토콜과 기존의 패스워드 기반 키교환 프로토콜을 이용하여 다음과 같은 2개의 인증된 키교환 프로토콜을 설계할 수 있다.

- *CDL* 대칭구조 인증된 키교환 프로토콜.
- *CDL* 비대칭구조 인증된 키교환 프로토콜.

본절에서는 Jablon 방식의 *CDL*프로토콜을 이용한 인증된 키교환 프로토콜([그림 1])을 설명한다.

다음은 Jablon방식 프로토콜이다.

**• Jablon 방식 CDL프로토콜**

**1) 사용자 등록**

- $p=2rq+1$을 만족하는 소수 $p$, $q$, $r$
- $h$ : 일방향 해쉬함수.

① 사용자 $U$는 패스워드 $p/w$를 선택하고, $p/w$에 대응하는 위수가 $q$인 생성자 $g_p$를 계산한다.

- $g_p = h(p/w)^{2r} \bmod p$

② 난수값 $y_i$(부분키)를 생성하여 $B_i$를 생성한 뒤, $j$ 비트의 비밀 마스터 키 $K$을 계산한다. 또한, 확인자 *proofPK*을 생성한다.

- $B_i = g_p^{y_i} \bmod p$
- $K = h(B_1 \| B_2 \| \ldots \| B_n) \bmod 2^j$
- $proofPK = h(K \| g_p)$

③ 인증서버가 정당한 부분키들을 보냈는지 확인할 수 있도록 *proofPK*를 계산하고, $U$가 올바르게 $K$을 계산했는지를 확인할 수 있도록 키쌍$(V, D)$을 선택한다.

- $proofPK = h(K \| g_p)$

- $D$ : 서명키.
- $D_K$ : 마스터키로 $D$를 암호화한 값.
- $V$ : 서명확인 키.

④ $CDL$서버 $\{S_i\}_{1\le i\le n}$에 각각의 부분 인증정보를 등록한다.

- $U \rightarrow S_i$ :( $y_i$, $D_k$, $V$, $proofPK$)

**2) $K$ 다운로드**

① $U$는 $p/w$를 통해 $g_p$를 계산한 뒤, 난수 $x$를 선택하여 블라인드 챌린지 $Q= g_p^x \bmod p$를 생성하여 사용자 $ID$와 함께 $\{S_i\}_{1\le i\le n}$에게 전달한다.

- $U \rightarrow \{S_i\}_{1\le i\le n}$ : ( $ID$, $Q$)

② $\{S_i\}_{1\le i\le n}$는 $R_i= Q^{y_i} \bmod p$를 계산한 뒤 $R_i$, $proofPK$, $D_K$를 사용자에게 전달한다.

- $\{S_i\}_{1\le i\le n} \rightarrow$ 사용자 : $R_i$, $proofPK$, $D_K$

③ $U$는 $B_i= R_i^{1/x} \bmod p$를 계산한 뒤 $K$을 복원한다. 또한 $K$이 올바른지 검증한다.

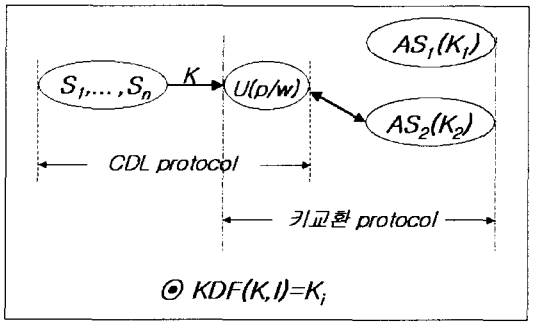
- $proofPK$=? $h(K \| g_p)$

④ $U$는 $D_{K_m}$에서 $D$를 계산한뒤 $Q$를 서명해 $\{S_i\}_{1\le i\le n}$에게 정당한 사용자임을 증명한다.

⑤ $\{S_i\}_{1\le i\le n}$는 각기 $V$를 통해 전달된 서명값이 올바른지 확인한다.

사용자 $U$는 자신이 기억하고 있는 패스워드( $p/w$) 만을 이용하여 키위탁 서버집합 $\{S_i\}_{1\le i\le n}$로부터 엔트로피가 높은 비밀키 $K$를 내려 받고 $K_i$를 생성하여 각각의 응용서버 $AS_i$에 자신을 등록한다. 그리고 $K_i$를 이용하여 각각의 $AS_i$와 인증된 키교환 프로토콜을 수행한다. 이러한 환경에서 $CDL$대칭구조 인증된 키교환 프로토콜은 사용자가 자신이 가지고 있는 키( $K_i$)가 응용서버가 인증을 위해 저장하는 키 ( $K_i$)와 같다는 것을 증명하는 프로토콜이고, $CDL$비대칭구조 인증된 키교환 프로토콜은 사용자가 응용서버가 사용자 인증을 위해 저장하는 확인 정보( $VK_i$) 에 대한 비밀키( $K_i$)를 알고 있다는 것을 증명하는 프로토콜이다.

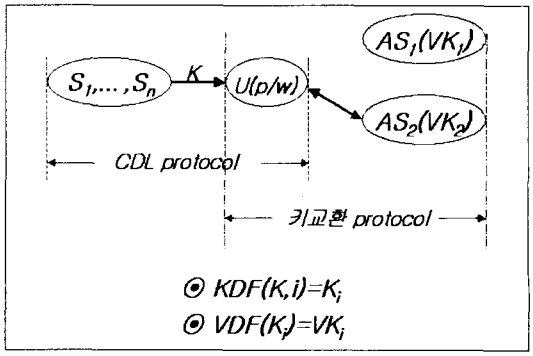
이러한 $CDL$프로토콜을 이용한 인증된 키교환 프로토콜은 $p/w$대신에 $p/w$에서 유도된 비밀키 $K$를 이용하여 키교환 프로토콜을 수행할 수 있다는 점에서 패스워드 기반 인증된 키교환 프로토콜에서 가능한 사전공격을 막을 수 있다.




〔그림 1〕 CDL 프로토콜을 이용한 대칭구조 인증된 키교환 프로토콜

[그림 1]인 $CDL$대칭구조 인증된 키교환 프로토콜을 설명한다. $CDL$대칭구조 인증된 키교환 프로토콜은 [그림 1]의 키교환 프로토콜이 대칭구조를 가지고 있는 인증된 키교환 프로토콜이다. 대칭구조를 이용하여 키교환 프로토콜을 구성한다면 공격자는 패스워드를 알아낼 필요 없이 목적한 응용서버를 공격해서 사용자를 위장할 수 있다. 즉 내부공격자에 대해 안전하지 않다. 예로 [그림 1]에서 $K_2$를 공격자가 알아낸다면 패스워드를 알 필요 없이 $AS_2$에 대해 사용자를 위장할 수 있다. 따라서 $CDL$대칭구조 인증된 키교환 프로토콜은 사용자가 서버를 신뢰하여야 하는 인증된 키교환 프로토콜이다.

$CDL$프로토콜과 기존에 존재하는 패스워드 기반 비대칭구조 인증된 키교환 프로토콜을 이용하여 "계산 불가능 안전성"에 기반 하여 서버에 대해 신뢰성이 없는 인증된 키교환 프로토콜을 구성할 수 있다. [그림 2]는 $CDL$ 비대칭구조 인증된 키교환 프로토콜을 설명한다. [그림 2]의 키교환 프로토콜로서 기존의 패스워드 기반 비대칭구조 인증된 키교환 프로토콜중 가장 효율성이 높은(서버의 분리된 모듈안에 저장되어




〔그림 2〕 CDL 프로토콜을 이용한 비대칭구조 인증된 키교환 프로토콜

있는 서버 비밀키가 인증서버의 메모리에 저장된다는 가정하에) *AMP*를 이용한다고 가정한다. 이렇게 구성된 *CDL*비대칭구조 인증된 키교환 프로토콜인 경우 공격자가 응용서버 $AS_i$에 위임된 인증정보( $VK_i$)를 얻을 지라도는 $K_i$가 $p/w$와는 달리 높은 엔트로피의 정보이므로 사전공격을 수행할 수 없다. 즉 어떤 다항식 시간 계산능력을 가진 공격자가 $AS_i$에 위임된 정보에 대해 사용자 인증정보를 계산할 수 없다. 따라서 *CDL*대칭구조 인증된 키교환 프로토콜과는 다르게 $AS_i$에 대하여 사용자를 위장할 수 없다. 따라서 *CDL*비대칭구조 인증된 키교환 프로토콜은 "계산 불가능 안전성"에 기반하여 사용자가 서버를 신뢰하지 않아도 된다. *CDL*비대칭구조 인증된 키교환 프로토콜에 대해 사용자를 위장할 수 있는 유일한 방법은 $\{S_i\}_{1\le i\le n}$ 전체에 대한 인증정보를 얻어 사전공격을 통하여 패스워드를 알아내어야 한다.

이러한 안전성에도 불구하고 *CDL*프로토콜을 이용한 인증된 키교환 방식은 *CDL*프로토콜을 먼저 실행한 후 별도로 키교환 프로토콜을 진행해야 되기 때문이 기존의 패스워드 기반 인증된 키교환 프로토콜에 비해 매우 비효율적이라는 문제점이 있다.

## V. MAP(Muti-Server Authenticated Key Exchange Protocol based on Password)

본 절에서는 "분산 안전성"에 기반하여 사용자가 서버를 신뢰하지 않아도 되는 인증된 키교환 프로토콜인 *MAP* 프로토콜을 설계한다. 설명에 앞서 본 논문에서의 지수 계산중 mod가 언급되지 않은 계산은 mod $P$계산을 전제로 한다.

### 5.1 사용자 등록

먼저 $U$는 *MAP*의 인증서버 집합 $\{S_i\}_{1\le i\le n}$에 자신을 등록한다. 초기 등록과정은 안전하고 인증된 채널을 통해 이루어진다고 가정하고 등록 과정은 아래와 같다. 본 논문에서 가정하는 안전하고 인증된 채널은 *PKI*등 다양한 방법으로 획득할 수 있다.

**1) 등록 요청.**

먼저 사용자 $U$는 인증서버 집합 $\{S_i\}_{1\le i\le n}$에게 등록요청 메시지를 보내 사용자 등록 과정을 시작한다.

- $U\rightarrow \{S_i\}_{1\le i\le n}$ : *RRM*
  (Registration Request message)

**2) 등록응답**

$U$로부터 *RRM*을 받은 $\{S_i\}_{1\le i\le n}$ 중 어떤 $S_i$는 $U$에게 현재 시스템에서 사용되고 있는 공용 파라미터를 전달한다.

- $\{S_i\}_{1\le i\le n}\rightarrow U$:{ $n(=|\{S_i\}_{1\le i\le n}|)$, $P(=2q+1)$}

**3) 인증정보 생성 및 전달.**

$\{S_i\}_{1\le i\le n}$로부터 공용 파라미터를 전달받은 $U$는 $\{S_i\}_{1\le i\le n}$에 등록할 인증정보를 아래와 같이 생성한다.

① 패스워드( $p/w$)및 비밀키 선택( $K$)

- 먼저 $U$는 자신이 기억할 $p/w$와 난수 $X(\in_R Z_q^*)$를 선택한다. 그 뒤 $p/w$를 $Z_P^*$상의 곱 연산에 대한 위수가 $q$인 원소로 대응시키는 함수 $BDF(\cdot)(=h(p/w)^2$: $h$는 해쉬함수)에 $p/w$를 입력해 $g_U(=BDF(p/w))$를 계산하고, $K=g_U^X(\bmod P)$를 계산한다.
  - $X\in_R Z_q^*$, $BDF(p/w)=g_U$, $K=g_U^X$.

② 비밀키 확인자( $h_K$) 생성.

- $K$에 대한 일방향 해쉬 함수 값 $h_K$을 계산한다.
  - $H(K)=h_K$ : $H(\cdot)$는 일방향 해쉬함수.

③ $\{S_i\}_{1\le i\le n}$의 부분키 생성.

- $U$는 각 인증서버가 저장할 서버 부분키를 다음과 같이 선택한다.
  - $k_i\in_R Z_q^*$ $(i=1,\cdots,n-1)$
  - $k_n = X-\sum_{i=1}^{n-1} k_i \pmod q$

④ $U$는 서버 집합에서 사용자를 인증할 때 필요한 서명키 쌍 $y$(서명 확인키), $x$(서명키)을 생성한 뒤, $x$를 $K$로 암호화해 $D(=E_K(x))$를 생성한다.

⑤ $p/w$에서 유도된 사용자 공개키 $\theta=g_U^{K''}(\bmod P$, $K''\equiv K \bmod q)$를 생성한다.

⑥ 각각의 인증서버에 위에서 생성한 부분 인증정보를 전달한다.

- $U\rightarrow S_i(i=1,\cdots,n)$ : ( $ID$, $k_i$, $\theta$, $x$, $D$, $h_K$)

**4) 사용자 등록**

각각의 인증서버 $S_i$는 사용자의 부분 인증정보 ( $ID$, $k_i$, $\theta$, $x$, $D$ )를 각각 사용자 $ID$별로 저장한다.

### 5.2 사용자와 서버간의 키교환 과정

$U$가 $\{S_i\}_{1\le i\le n}$내의 특정서버 $S_j(\in_R\{S_i\})$와 인증된 키 교환을 수행하는 과정을 살펴보자.

① $U$는 $a', a\in_R Z_q^*$를 선택하여, $Q_{U-S}$를 $\{S_i\}_{1\le i\le n}$에게 브로드캐스팅 한다.

- $Q_{U-S} = ID\|g_U^a\|g_U^{a'}\|S_j$

② $G(\{S_i\}_{1\le i\le n}: i\neq j)$의 각각의 인증서버 $S_i$는 $R_{G-U}$를 $U$에게 전달하고, $S_j$는 $b\in_R Z_q^*$를 선택한 뒤 $R_{S_j-U}$를 생성하여 $U$에게 전달한다.

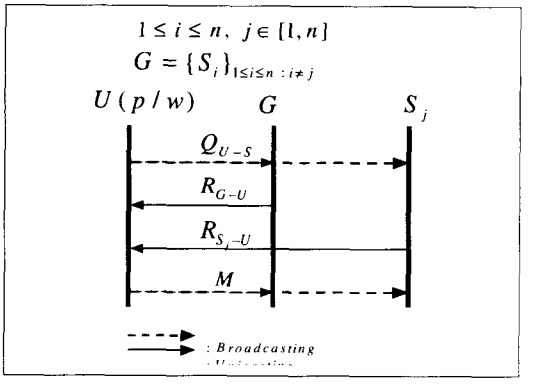
- $R_{G-U} = \{ S_i\|g_U^{a\cdot k_i}\|h_K \}_{1\le i\le n:\, i\neq j}$
- $R_{S_j-U} = S_j\|g_U^{a\cdot(k_j+b)}\|g_U^{a'\cdot b}\|D\|h_K$

③ $U$는 $R_{S_j-U}$와 $a', a$를 가지고 $g_U^{a\cdot k_i}, g_U^b$를 계산하고, $g_U^{a\cdot k_i}$와 $R_{G-U}$를 가지고 $K'$를 계산한다. 또한 $H(K')=h_K$를 통해 올바른 $K$를 생성했는지 확인한 뒤 $S_j$와의 세션키 $K_S$를 계산한다.

- $K'=(\prod_{i=1}^{n} g_U^{a\cdot k_i})^{1/a}$
- $K_S=(g_U^b)^{a+K'(\mathrm{mod}\,q)}$

④ $S_j$는 $U$가 $K_S$를 계산하는 동안 $U$의 $\theta$를 이용하여 세션키 $K_S$를 계산한다.

- $K_S=(\theta\cdot g_U^a)^b$




〔그림 3〕 MAP프로토콜

⑤ $U$는 자신이 정당한 사용자라는 것과 키확인을 위해 $K$를 이용해 $D$를 복호화해 $x$를 얻은뒤 $x$를 가지고 $Q_{U-S}$에 서명해 $M$을 생성하여 브로드캐스팅을 한다.

- $Sig = Sig_x(Q_{U-S})$
- $M = ID\|S_j\|Sig$

⑥ $M$을 전달받은 $\{S_i\}_{1\le i\le n}$ 각기 $x$를 가지고 $M$이 정당한지 확인한다. $S_j$는 $M$이 정당하면 $U$와 같은 세션키를 공유하고 있음을 확인할 수 있다.

- $Verification_x(Sig, Q_{U-S}) =^{?} true$

$MAP$을 통해 계산하는 $K$는 $p/w$와는 달리 엔트로피가 높은 값이다. 따라서 $\theta$는 각 인증서버 집합의 사용자 부분키($k_i$)가 모두 드러나기 전까지는 사전 공격에 대해 안전하다.

### 5.3 MAP과 CDL비대칭 구조 인증된 키교환 프로토콜의 효율성 비교.

"분산 안전성"에 기반하여 서버에 대한 신뢰성이 없는 패스워드 기반 인증된 키교환 프로토콜 $MAP$과 "계산 불가능 안전성"에 기반하여 서버에 대한 신뢰성이 없는 $CDL$비대칭 구조 인증된 키교환 프로토콜의 효율성을 분석하겠다. $CDL$비대칭구조 인증된 키교환 프로토콜에서 사용되는 키교환 프로토콜은 기존의 비대칭구조 패스워드 기반 인증된 키교환 프로토콜중 가장 효율성이 뛰어난 $AMP$를 사용하여 비교한다. 본 논문에서 지수 계산량은 사용자가 난수를 선택한 시점으로부터 키교환을 완료하기까지 소요되는 시간에서 메시지 전송에 소요되는 시간을 제외한 시간을 의미하고, 사전계산(pre-computation)은 고려하지 않겠다. 따라서 $MAP$의 지수 계산량은 다음과 같이 8번으로 고정된다.

〔표 1〕 MAP과 CDL비대칭구조 인증된 키교환 프로토콜의 효율성 비교

| 구분 | $CDL$ 비대칭구조 | MAP |
|---|---|---|
| 지수 계산량 | $7.6\sim n+5.6$ | 8 |
| 메시지 전송횟수 | 6 | 3 |

위의 비교에서 서버의 개수 $i$는 $2\le i\le n$이다. 즉 서버의 개수가 2개 일 경우 $CDL$비대칭 구조의 경우 7.6번의 지수 계산을 해야 하고 서버의 개수가 증가함에 따라 지수 계산량이 증가한다. 그에 비해 $MAP$은 서버의 개수의 크기와 관계없이 총 8번의 지수계산을 요구한다. 또한 $CDL$비대칭구조의 메시지 전송횟수는 $CDL$프로토콜 2번(마지막 $Q$에 대한 서명전달은 전송횟수에서 고려하지 않는다. Ⅳ 절의 Jablon 방식의 $CDL$ 프로토콜 참고)과 $AMP$ 4번으로 총 6회이고, 이에 비해 $MAP$은 키확인 까지 총 3번으로 메시지

전송횟수에서도 *MAP*은 효율성이 높다.

## Ⅵ. MAP 안전성 분석

정당한 사용자가 정당한 서버와 MAP 프로토콜을 실행하였다면 서버와 사용자 사이에 같은 세션키가 생성된다는 것은 위에 서술된 프로토콜을 통해 알 수 있다. 다음은 여러 가지 측면에서 안전성을 검토한다.

**1) *MAP*이 성공적으로 수행되는 동안 도청자에 의해 세션키에 대한 어떤 정보도 드러나지 않는다.**

증명을 위해 다음과 같은 기호를 정의한다.

- $Adv_{MAP}^{C-S}$ : MAP 프로토콜의 공개값과 키교환 프로토콜중 전송되는 값들을 입력값으로 하여 *MAP*의 세션키를 계산하는 알고리즘.
- $Adv_{DDLP}$ : p (mod p)를 소수, g를 $Z_p^*$= <g>를 만족하는 생성자라 하자, 이때 임의의 a, a',b $\in_R Z_p^*$에 대해 입력값 $g^a$, $g^{a'}$, $g^{a\cdot b}$, $g^{a'\cdot b}$, g, p로부터 $g^b$를 계산하는 알고리즘.
- $Adv_{DLP}$ : p (mod p)를 소수, g를 $Z_p$= <g>를 만족하는 생성자라 하자, 이때 임의의 a, b $\in_R Z_p^*$에 대해 입력값 $g^a$, $g^{a\cdot b}$, g, p로부터 $g^b$를 계산하는 알고리즘. 즉 *DLP*문제를 푸는 알고리즘.

위에서 정의 한 알고리즘을 사용하여 도청자가 세션키 $K_S$값에 대한 어떤 정보를 알아내는 것은 적어도 *DLP*문제만큼 어렵다는 것을 보인다. 키 교환 과정에서 어떤 사용자에 대한 인증서버 집합의 모든 사용자 부분키 정보와 사용자의 패스워드가 노출 된 환경을 가정하자. 증명의 편의를 위해 공격자에게 알려진 사용자의 모든 부분키를 1로 가정한다. 그러면 $K$값은 $g_U^n$이 된다. 또한 증명의 편의를 위해 $K$값을 1로 가정한다. 이때 아래의 $Adv_{MAP}^{C-S}$의 파라미터는 $Adv_{DDLP}$와 비교할 수 있고, $Adv_{DDLP}$의 파라미터는 $Adv_{DLP}$의 파라미터와 다음과 같이 비교할 수 있다.

$$Adv_{MAP}^{C-S}(g_U^a, g_U^{a'}, g_U^{a\cdot(1+b)}, g_U^{a'\cdot b}, g_U, P)$$
$$= g_U^{a'\cdot b}\cdot g_U^b = K_S$$
$$Adv_{DDLP}(g_U^a, g_U^{a'}, g_U^{a\cdot b}, g_U^{a'\cdot b}, g_U, P)$$
$$= K_S / g_U^{a'\cdot b} = g_U^b \; Adv_{DLP}(g_U^{a+a'}, g_U^{(a'+a)\cdot b}, g_U, P)$$
$$= g_U^b$$

즉, $Adv_{MAP}^{C-S}$가 존재한다면, $Adv_{DDLP}$가 존재할 수 있고, $Adv_{DDLP}$가 존재한다면 $Adv_{DLP}$가 존재할 수 있다. 따라서 *MAP*프로토콜의 공개된 정보와 사용자의 패스워드 그리고 사용자 부분키를 이용하여 세션키를 알아내는 것은 적어도 *DLP*문제를 푸는 것만큼 어렵다.

**2) 전체 서버집합의 어떤 부분키를 통해 다른 부분키를 알아내거나 확인할 수 없다.**

*MAP*프로토콜의 공개 파라미터와 전체 부분키 $\{k_i\}_{1\le i\le n}$ 중 임의의 $k_i$를 제외한 모든 패스워드 $\lambda$가 다음과 같이 주어지고 다고 가정하자. 설명의 편의를 위해 $k_j$를 제외한 모든 $\{k_i\}_{1\le i\le n}$가 1이라고 가정한다.
$(g_U^a, g_U^{a'}, g_U^a, g_U^{a\cdot(k_i+b)}, g_U^{a'\cdot b}, \theta, Y, x, H(K), P, g_U)$
위와 같은 정보를 통해서 $k_i$를 계산하기 위해서는 *DLP*를 풀어야 한다.

**3) 성공적인 프로토콜의 진행동안 패스워드에 대한 사전공격이 불가능하다.**

패스워드에 대한 사전공격은 모든 서버집합의 부분키가 드러난 상태에서 $g_U^K$을 통해서만 가능하다. 또한 부분키를 알기 위해서는 2)번에서 보인 것처럼 *DLP*를 풀어야 하므로 패스워드에 대한 사전공격을 하기 위해서는 *DLP*를 풀어야 한다.

**4) *MAP*은 "분산 안전성"기반하여 서버에 대한 신뢰성이 없는 인증프로토콜이다.**

[정의 2]의 (1)에 대한 *MAP*은 당연하게 만족한다. $\Omega(\bigcup_{j=1}^{n} I_i^j(\delta)) = I_i(\delta)$를 만족하는 다항식 시간 알고리즘 $\Omega$는 $K=\left(\prod_{i=0}^{n-1} g_U^{a\cdot k_i}\right)^{1/a}$ 이고, $K$를 통해서 사용자 인증을 한다.

[정의 2]의 (2)에 대해 증명은 다음과 같다. 우선 전혀 사용자 인증정보에 대한 정보를 가지지 않은 공격자가 사용자 인증정보를 알아낼 수 있는 확률은 $Z_P^*$에서 어떤 수를 임의로 선택하는 확률이다. 따라서 $\Pr[I_i'(\delta)=I_i(\delta) \mid I_i'(\delta)=B(\cdot)]=\frac{1}{P-1}$이다.

또한 *MAP* 인증서버의 전체 부분 인증정보 $\{\{k_i\}_{1\le i\le n}, \theta, x, Y\}$중 한개의 임의의 부분키 $k_j$를 제외한 전체 사용자 부분 인증정보와 사용자가 프로토콜의 진행 동안 전달하는 파라미터 $g_U^a$, $g_U^{a'}$, $g_U^{a\cdot(k_i+b)}$, $g_U^{a'\cdot b}$를 공격자가 얻었다고 가정할 때 *MAP*프로토콜에서 공격자가 사용자의 인증정보 $K$를 알아내기 위

해서는 $\theta$를 통해 정당한 $k_j$와 $g_U$를 계산해 내어야 한다. 하지만 미지수가 2개이기 때문에 $\theta$를 통해서 정당한 $k_j$와 $g_U$를 확인할 수 없다. 따라서 정당한 인증정보 $K$를 선택하는 확률은 $Z_P^*$에서 어떤 수를 임의로 선택하는 확률과 같다.

따라서

$$\Pr\left[I_i'(\delta)=I_i(\delta)\mid I_i'(\delta)=A\left(\bigcup_{w_i\in C} I_i'(\delta)\right)\right]=\frac{1}{P-1}$$

이고, 따라서

$$\Pr\left[I_i'(\delta)=I_i(\delta)\mid I_i'(\delta)=A\left(\bigcup_{w_i\in C} I_i'(\delta)\right)\right]$$
$$=\Pr[I_i'(\delta)=I_i(\delta)\mid I_i'(\delta)=B(\cdot)]$$

이다.

따라서 *MAP*은 "분산 안전성"에 기반하여 서버에 대한 신뢰성이 없는 인증프로토콜이다.

**5) 세션키가 드러나도 패스워드는 드러나지 않을 뿐만 아니라 사전공격도 불가능하다.[12](DS attack)**

*MAP*은 모든 부분 인증정보가 노출되었을 지라도 $g_U^K$에 대한 사전공격 이외에 사전공격 할 수 없다. 따라서 모든 부분 인증정보가 알려졌을 때 $g_U^K$에 대한 사전공격을 고려하지 않는다고 가정하면, 공격자에게 주어지는 공개 파라미터는 다음과 같다.

$(g_U^a, g_U^{a'}, g_U^{a'\cdot b}, g_U^K, Y, x, H(K), K_S, P)$

위의 정보에 대해 공격자가 모르는 미지수가 각각 2개 이상이므로 공격자는 사전공격을 수행할 수 없다. 따라서 세션키가 알려졌다 할지라도 공격자는 패스워드를 알 수 없다.

**6) 만일 사용자의 패스워드가 공격자에게 알려졌을 경우에도 공격자는 과거의 통신내용을 알 수 없어야 한다.(*Perfect forward secrecy*)**

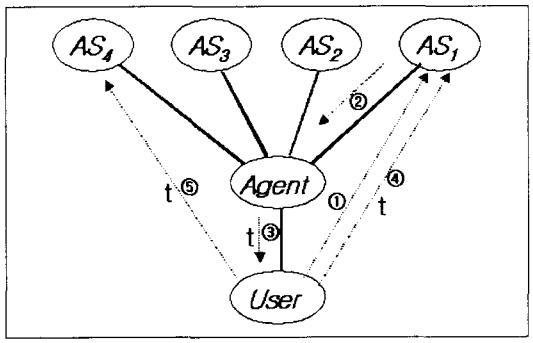
위의 1)번 증명에서 $g_U$가 알려진 상황에서도 공격자는 세션키를 알 수 없었다. 따라서 1)번 증명에 의해 패스워드를 알아낸 공격자도 과거의 통신내용을 알 수 없다.

또한 MAP 프로토콜은 로그인 실패 횟수를 체크하여 온라인 사전공격을 막을 수 있다.

## Ⅶ. MAP프로토콜을 이용한 SSSO(Secure Single Sign On)

SSO는 여러 개의 응용 서버를 제공하는 네트웍에서 각각의 응용서버에 대해 별도의 인증을 받지 않고 한번의 인증으로 추가적인 인증과정 없이 모든 응용서버에 접근 할 수 있도록 하는 편리한 네트웍 모델이다[13]. 이러한 SSO는 인증대행 서버를 네트웍에 두어 인증과정을 일괄적으로 처리함으로서 그러한 목적을 달성한다.

[그림 4]은 인증대행 서버(Agent)를 이용한 SSO의 구성도 이다. 사용자는 [그림 4]와 같이 자신이 접근하기 원하는 응용서버에($AS_1$) 접근을 요청하면 이러한 접근 요청을 받은 $AS_1$는 인증대행 서버에게 사용자의 접근에 대한 인증을 대행하여 줄 것을 요청하고 인증대행 서버는 $AS_1$대신해서 사용자를 인증하고 사용자의 접근 통제 정보가 들어있는 티켓 t를 사용자에게 전달한다. 이러한 티켓을 전달받은 사용자는 자신의 접근 통제 규칙에 따라 인증대행 서버가 인증을 대행하여 주는 응용서버에 추가적인 인증 없이 ④과 같이 접근할 수 있다. 기존의 *SSO*모델에서 인증대행 서버는 SSO에 참여하는 전체 응용서버의 사용자 인증정보를 관리하게 된다.

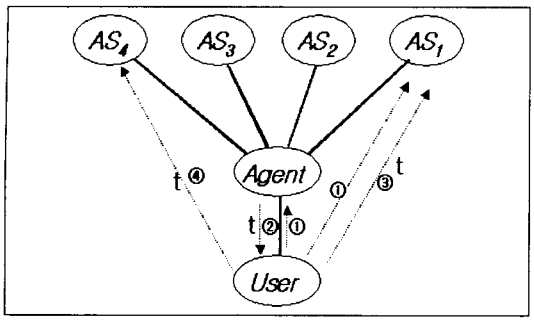



〔그림 4〕 SSO 구조

따라서 공격자는 SSO 인증대행 서버만을 공격하여서 전체 네트웍의 사용자 인증정보를 알아낼 수 있다. 특히 패스워드 기반의 인증을 제공하는 SSO 인증대행 서버가 공격을 당해 사용자 패스워드 파일이 노출된다면 그 인증대행 서버가 인증을 대행해 주는 모든 응용서버($AS_i$)의 모든 사용자 패스워드가 공격자에게 알려지게 된다. 즉, SSO의 인증대행 서버는 신뢰기관이 된다.

이러한 문제점을 서버에 대한 신뢰성이 없는 패스워드 기반 인증프로토콜인 *MAP*을 이용하여 SSO를 구성하면 신뢰기관이 존재하는 SSO가 가지는 문제점을 해결할 수 있다. *MAP*을 이용한 SSO구성은 [그림 5]와 같다.

[그림 5]에서 (Agent, $AS_1$), ... ,(Agent, $AS_4$)의 쌍을 각각 별도의 *MAP* 프로토콜로 구성하여, 총 4개의 인




〔그림 5〕 SSSO 구조

증서버 그룹을 형성된다. 사용자는 4개의 인증서버 그룹 중 하나에 등록하게 된다. 사용자 등록 후 기존의 SSO처럼 자신이 속한 인증서버 그룹과 한번의 인증과정을 통해서 티켓 t를 전달 받은 뒤 [그림 5]와 같이 t에 포함된 자신의 접근 통제 규칙에 따라 SSSO에 포함된 응용서버에 접근이 가능해진다. SSSO의 경우 공격자가 Agent를 공격해서 사용자 인증정보를 얻어낸다 할지라도 공격자는 사용자를 위장할 수 없다. 또한 만일 공격자가 한 쌍의 (Agent, $AS_1$) 공격해서 사용자의 인증정보를 얻어낸다 할지라도 공격자는 $AS_1$에 등록되어 있는 사용자에 대해서만 사용자 위장이 가능하다.

## Ⅵ. 결 론

본 논문에서는 사용자가 서버를 신뢰하지 않아도 되는 인증프로토콜을 구성하기 위해 인증서버에 위탁되는 인증정보에 대한 "계산 불가능 안전성"과 "분산 안전성"을 정의했다. 또한 기존의 Jablon 방식의 *CDL*프로토콜과 패스워드 기반 비대칭 인증된 키교환 프로토콜을 이용하여 구성한 "계산 불가능 안전성"을 기반으로 서버에 대한 신뢰성이 없는 *CDL*비대칭구조 인증된 키교환 프로토콜 보다 효율적인 패스워드 기반 인증된 키교환 프로토콜인 MAP을 제시하였다. 또한 이러한 서버에 대한 신뢰성이 없는MAP을 이용한 SSO가 신뢰기관이 존재하는 SSO의 문제점을 해결함을 보였다.

## 참 고 문 헌

[1] D. Jablon, Strong password-only authenticated key exchange, Computer Communication Review, 26(50) 5-26. October 1996.

[2] C. Kaufman, R. Perlman, "PDM : A New Strong Password-Based Protocol".

[3] T. Kwon, "Authentication and Key Agreement via Memorable Password".

[4] S. M. Bellovin and M. Merritt, Encrypted Key Exchange : Password-Based Protocols Secure Against Dictionary Attacks(or here), Proceedings of the I.E.E.E. Symposium on Research in Security and Privacy, Oakland, May 1992.

[5] S. M. Bellovin and M. Merritt, "Augmented encrypted key exchange. A password-based protocol secure dictionary attacks and password file compromise", Technical report. AT&T Bell Laboratories 1994.

[6] D. Jablon, "Extended password methods immune to dictionary attack", In WETIC'97 Enterprise Security Workshop, Cambrige, MA, June 1997.

[7] T. Wu, Secure remote password protocol, Internet Society Symposium on Network and Distributed System Security, 1998.

[8] M. Bellare and P. Rogaway, The AuthA Protocol for password-based authenticated key exchange.

[9] S. Lucks, Open Key exchange: how to defeat dictionary attacks without encrypting public keys. The Security Protocol Workshop'97, April 7-9, 1997.

[10] W. Ford and B. Kaliski, Server-Asisted Genenation of a Strong Secret from a Password, Proc, 9th Interantional Workshops on Enabling Technologies : Infrastructure for Collaborative Enterprises, IEEE, June 14-16,200.

[11] D. Jablon, Password Authentication Using Multiple Servers, LNCS 2020: Topics in Cryptology -- CT-RSA 2001, April 8-12, 2001 Proceedings, pp. 344 ~360, 2001, Springer - Verlag.

[12] D. Denning and G. Sacoo. Timestamps in Key distribution. communications if the ACM, August 1981.

[13] 손태식, 이상하, 유승화, 김동규, "단일 인증 시스템의 인증 기법과 인증 모델 분석" 정보보호학회지, 제11권, 제4호, 2001.8

## 〈著者紹介〉

**이 정 현 (Jung Hyun Lee) 학생회원**
2001년 2월 : 고려 대학교 전산학과 졸업.
2001년 3월~현재 : 고려 대학교 정보보호 대학원 석사 과정
<관심분야> 암호이론, 암호 프로토콜

**김 현 정 (Hyun-Jeong Kim) 학생회원**
1994년 2월 : 경희 대학교 수학과 졸업.
1984년 1월~1999년 12월: 삼성 SDS 근무
1999년 9월 2001년 월 : 고려대학교 수학과 석사
2001년 9월~현재 : 고려대학교 정보보호 대학원 박사과정
<관심분야> 암호이론, 암호 프로토콜, 정보은닉

**이 동 훈 (Dong Hoon Lee) 정회원**
1984년 : 고려대학교 경제학과 졸업
1987년 : Oklahoma Univ. 전산학과 석사
1992년 : Oklahoma Univ. 전산학과 박사
1993년~현재 : 고려대학교 전산학과 교수
2000년~현재 : 고려대학교 정보보호 대학원 교수
<관심분야> 암호이론, 암호 프로토콜, 정보이론