멀티 에이전트를 이용한 Single Sign-On 인증 모델에 관한 연구

학생회원 서 대 희*, 정회원 이 임 영*

A Study on Single Sign-On Authentication Model using Multi Agent

Dae-Hee Seo* *Student Member*, Im-Yeong Lee* *Regular Member*

요 약

인터넷의 급속한 확산으로 인해 사용자들은 다양한 서비스를 제공받고 있다. 그러나 대부분의 사용자들은 수많은 서비스 사이트에 여러 개의 ID와 패스워드를 이용해 서비스를 이용한다. 따라서 이를 관리해야 하는 사용자와 관리자의 비효율적인 관리체계에 따른 보안적 취약점을 보완하기 위해 제안된 시스템이 SSO(Single Sign On)이다.

SSO는 사용자와 관리자에게 효율성과 보안성을 높여 줄 수 있는 시스템으로써 최근 대중화된 SSO 시스템의 경우 브로커 인증 모델에 단일 에이전트를 결합한 혼합형 인증 시스템이 상용화 되고 있다. 그러나 혼합 형태의 인증 시스템의 경우 사용자의 사전 등록 문제, 익명 사용자 문제뿐만 아니라 참여 개체들 간의 부인봉쇄 서비스를 제공하지 못하고 있어 보안상의 많은 취약점을 발생시키고 있다. 또한 에이전트 자체에 대한 보안 서비스를 제공해 주지 못해 사용자의 개인 정보뿐만 아니라 SSO 시스템에 많은 취약점을 노출시키고 있다.

따라서 본 논문에서는 SSO 시스템의 인증 모델 중 브로커 인증 모델에 멀티 에이전트 시스템을 결합한 인증 모델을 제시하였다. 제안 방식은 기존의 혼합형 인증 시스템에 적용된 단일 에이전트의 보안적 취약점을 보완한 안전한 멀티 에이전트 시스템을 적용하여, 기존 브로커 인증 모델과 단일 에이전트의 혼합형 인증 시스템에서 제공하지 못했던 여러 가지 보안 사항을 만족할 수 있는 SSO 인증 모델을 제안하였다.

ABSTRACT

The rapid expansion of the Internet has provided users with a diverse range of services. Most Internet users create many different IDs and passwords to subscribe to various Internet services. Thus, the SSO system has been proposed to supplement vulnerable security that may arise from inefficient management system where administrators and users manage a number of IDs.

The SSO system can provide heightened efficiency and security to users and administrators. Recently commercialized SSO systems integrate a single agent with the broker authentication model. However, this hybrid authentication system cannot resolve problems such as those involving user pre-registration and anonymous users. It likewise cannot provide non-repudiation service between joining objects. Consequently, the hybrid system causes considerable security vulnerability. Since it cannot provide security service for the agent itself, the user's private information and SSO system may have significant security vulnerability.

* 순천향대학교 정보기술공학부

※ 본 과제(결과물)는 산업자원부와 한국산업기술재단에서 시행한 지역혁신인력양성사업(지역전략산업 석박사 연구인력 양성사업)의 연구결과입니다.

논문번호 : 030394-0805, 접수일자 : 2003년 8월 5일

This paper proposed an authentication model that integrates a broker authentication model, out of various authentication models of the SSO system, with a multi-agent system. The proposed method adopts a secure multi-agent system that supplements the security vulnerability of an agent applied to the existing hybrid authentication system. The method proposes an SSO authentication model that satisfies various security requirements not provided by existing broker authentication models and hybrid authentication systems.

Keyword : Single Sign-On, Multi Agent, Authentication Model

1. 서 론

최근 인터넷의 급속한 확산과 더불어 대중화된 웹 서비스에서 사용자들은 수많은 웹 사이트에 가입하여 다양하고 편리한 서비스를 제공받고 있다.
이러한 현상은 인터넷의 급성장에 따른 웹 서비스 기업의 차별화에 따른 현상이며, 점차 세분화되고 품질 높은 서비스를 요구하는 사용자의 욕구에 의한 것으로 볼 수 있다.

그러나 서비스의 복잡성으로 인해서 사용자들은 시스템에 액세스 할 때마다 각각 다른 아이디와 비밀번호를 기억해야 하는 불편과 함께 관리자로서는 모든 아이디와 비밀번호를 관리해야 된다는 문제점이 발생하게 된다.

따라서 이를 해결할 수 있는 시스템이 SSO(Single Sign-On)이다. SSO는 기업에서 제공하는 다양한 웹 서비스 환경에서 한 사이트에서만 인증 받으면 모든 사이트에 ID와 패스워드 없이 서비스를 이용할 수 있는 시스템이다. 따라서 SSO는 사용자뿐만 아니라 관리자에게 비용 및 편리성을 제공할 수 있는 매우 강한 솔루션이라 할 수 있다.

특히, 기존의 SSO 시스템을 채택하지 않은 시스템의 경우 사용자가 많은 아이디와 패스워드를 기반으로 여러 인터넷 서비스를 이용하여 서비스 효율과 사용자의 프라이버시 정보의 노출에 따른 안전성을 보장받을 수 없었다. 따라서 SSO를 도입함으로써 기존 시스템에서 고려되지 않았던 사용의 편리성과 사용자의 통합 관리를 통해 최대한의 시너지 효과를 얻을 수 있다. 따라서 기존의 시스템은 각각의 서비스 서버가 독립적으로 관할하던 데이터들을 통합하여 관리할 수 있는 관리의 효율성과 더불어 다양한 보안 서비스를 제공할 수 있어 기존 서비스와는 차별화된 서비스이다. SSO 서비스에서 가장 중요한 요소는 사용자 인증모델에 있으며, 이는 전체적인 SSO의 효율성과 안전성에 매우 큰 영향을 미치게 된다[1~3].

이에 본 논문에서는 SSO의 인증 모델 중에서 브로커 모델을 기반으로 멀티 에이전트를 결합하여 효율성과 보안성을 향상시킨 SSO 인증 모델을 제안하였다. 본 논문의 2장에서는 기존 SSO 인증 모델이 가지는 보안적 취약점을 분석하고, 3장에서는 SSO 인증 모델의 개요와 SSO 인증 모델이 가져야 하는 보안 요구사항을 설명한다. 4장에서는 기존 방식의 취약점을 보완하면서 2장에서 제시한 보안 요구사항을 만족하는 안전하고 효율적인 SSO 인증 모델을 제안하였다. 5장에서는 4장에서 제안된 방식을 2장에서 제시한 보안 요구사항으로 분석한 뒤 마지막으로 6장에서 결론을 맺고자 한다.

2. 기존 방식 분석

다음은 SSO 인증 모델을 4가지로 구분하여 각 모델에 따른 상용화 제품의 분석을 통해 보안적 취약점을 분석하고자 한다.

2.1 브로커 인증 모델

Entrust get Access 제품의 경우 브로커 인증 모델을 응용하여 개발되었다. Entrust get Access는 kerberized 프로그램을 이용하여 구현된다. 또한 중앙 집중식 관리로 사용자의 인증 정보를 관리하고 모든 통신 구간의 암호화를 통해 안전성을 유지하고 있다[4,7,8]. 그러나 다음과 같은 보안적 취약점을 노출하고 있다.

- 사용자의 사전 등록 문제 : 브로커 인증 모델의 경우 사용자의 사전 등록에서 사용자의 비밀스러운 패스워드의 노출에 대한 공격에 매우 취약하여 사용자의 도용이나 위장 공격에 노출되어 있다.
- 익명 사용자 문제 : 브로커 인증 모델에서는 반드시 등록되어 있는 ID와 패스워드의 비교를 통해 사용자를 인증한다. 그러나 인터넷과 전자상거래의 발전에 따른 익명 사용자의 확산은 브로커 인증 모델을 사용하는 서비스의 경우 이를 제공 할 수 없는 취약점이 있다.

- 부인봉쇄 서비스 문제 : 브로커 인증 모델의 경우 기존 프로그램을 수정하여 모든 통신에 대한 암호화 서비스를 제공하고 있다. 그러나 각 참여 개체간의 부인봉쇄 서비스를 제공하지 못하고 있어 개체의 송/수신 부인에 취약점이 있다.
- 인증 모델의 안전성 : kerverized 프로그램의 수정을 통한 인증 모델의 경우 호스트는 안전하고 동기화된 클럭 정보를 제공해야 한다. 실제 환경에서는 초기에 분배된 키가 패스워드로 유도된 것이므로 안전성은 패스워드에 의존하게 된다. 따라서 키의 공유시 패스워드의 안전성을 유지하면서 상호 인증을 제공해야 하지만 브로커 인증 모델의 경우 패스워드의 사전공격과 일방향 식별로 보안적 취약점을 내포하고 있다.

2.2 에이전트를 이용한 브로커 인증 모델

에이전트를 이용한 브로커 인증 모델은 기존 브로커 모델에 에이전트를 결합한 형태로서 KsignPassOne 제품이 대표적이다.

KsignPassOne 제품의 경우 에이전트를 이용해 자동 로그인, 프로파일 암호화 기능을 제공하며 이를 관리하는 PassOne Admin은 사용자의 프로파일 등록 기능, 프로파일 등록, 시스템 환경 설정 및 모니터링 기능을 제공하고 있다[4,7,8]. 그러나 KsignPassOne의 경우 다음과 같은 보안적 취약점을 지적할 수 있다.

- 익명 사용자 문제 : KsignPassOne의 경우 인증서 기반의 사용자 인증 과정을 수행한다. 따라서 익명 사용자에 대한 서비스는 한계가 있다.
- 기밀 연산 : KsignPassOne 서비스에서 사용되는 에이전트는 원격지에서 초기 비밀값을 노출시키지 않고 명령을 수행함으로써 에이전트의 불법 복사나 접근을 수행하지 못하고 있는 보안적 취약점이 존재한다.
- 사용자 정보 보호 기능 문제 : 에이전트를 이용하고 있는 KsignPassOne는 사용자의 프로파일을 에이전트에 전송한 후 보안 채널이 형성되어 사용자의 프로파일을 재전송한다. 따라서 보안 채널 이전에 전송되는 사용자 정보에 대한 보안 취약성이 발생한다.

2.3 게이트웨이 인증 방식

게이트웨이 인증 방식을 응용한 제품은 KyberPASS가 있다. KyberPASS는 프로그램의 수정이 쉽고 적용이 쉬운 게이트웨이 방식을 따르고 있다. 그러나 방화벽과 같은 위치에 존재하여 SYN Flood와 같은 공격에 취약할 수 있으며 게이트가 여러개일 경우 이에 대한 동기화가 어렵다는 단점이 있다. 이상의 일반적인 구성상에 취약점을 가질 뿐만 아니라 브로커 인증모델과 같은 보안적 취약점을 그대로 내포하고 있다[4,7,8]. 다음은 일반적인 보안 취약점과 브로커 인증 모델에서 언급한 보안적 취약점 이외의 내용에 대해 설명하고자 한다.

- 인증에 대한 권한 설정 문제 : 스크립 방식의 경우 사용자에 따른 차별화된 권한 설정이 성립되어 있지 않다. 따라서 악의적인 내부 사용자의 공격에 노출되어 있다.
- 관리체계 : 게이트웨이 인증 방식의 경우 관리의 어려움으로 집중적인 사용자 인증 관리가 어려워 보안 정책 수립이나 적용이 부적절하다는 취약점을 가지고 있다.

2.4 스크립 인증 모델 방식

스크립을 이용한 인증 모델의 경우 프로그램의 수정이 쉽고 간단한 수정으로 다양한 시스템에 적용이 간편하다는 장점이 있다. 그러나 클라이언트에 대한 관리가 어려울 뿐만 아니라 클라이언트와 서버 연결 서비스를 제공하고 있지 않아 전송 데이터에 대한 보안 취약점을 내포하고 있다[4,8]. 이상의 일반적인 사항 이외에도 다음과 같은 보안 취약점이 발생한다.

그림 1. 스크립 모델 인증 방식

- 관리체계 : 스크립 인증 방식 역시 게이트웨이 인증 방식과 동일하게 클라이언트에 대한 관리가 어려워 중앙 집중식 관리가 필요하다.
- 무결성과 기밀성 : 스크립 방식의 경우 간단한 수정으로 다양한 시스템에 적용이 가능하므로 불법 사용자에 의해 스크립이 수정되었을 경우 전체 시스템에 영향을 미칠 수 있을 뿐만 아니라 초기 전송되는 데이터에 대한 보안 서비스를 제공하고 있지 않아 이에 대한 취약점이 문제시 되고 있다.

3. SSO의 브로커 인증 모델 개요와 보안 요구사항

현재의 정보통신 사회에서 널리 사용되는 인터넷 서비스는 기본적으로 ID/패스워드 기반의 인증을 사용한다. 이때 사용자는 여러 웹 서비스에서 대해서 각각 다양한 ID/패스워드를 기억해야 한다는 어려움을 가진다. 마찬가지로 웹 서비스 관리자 역시 여러 사용자들의 ID/패스워드를 관리하는데 많은 비용 및 노력을 소모해야 한다. 따라서 한 번의 안전한 인증 과정을 통해 사용자 및 관리자의 편리를 도모할 수 있는 SSO 시스템의 적용이 필수적으로 요구되고 있다.

SSO는 사용자를 한번 인증 한 뒤 모든 서비스를 이용할 수 있도록 하는 기술로서 사용자의 패스워드 분실 처리에 관련된 자원 및 비용을 감소 시킬 수 있다. 또한 중앙 집중식 인증/권한 관리를 통해 관리 생산성을 증대시켜 사용자 뿐만 아니라 관리자의 요구사항도 만족할 수 있는 방법이다.

SSO를 구성하기 위해서는 크게 3가지의 과정으로 구분할 수 있다. 첫 번째로는 사용자의 신원을 확인하는 인증(Authentication) 과정이며, 두 번째로는 인증 과정을 거친 후 해당 자원에 대한 접근이 가능하도록 하는 인가(Authorization) 과정, 마지막으로는 인증과 인가 과정 뿐만 아니라 사용자와 ID 및 패스워드를 관리하는 통합 관리(Administration Management)를 들 수 있다. 이상의 3가지 과정 중에서도 사용자를 인증하는 과정은 매우 중요하다 할 수 있으며, 이와 관련된 대표적인 인증 모델이 브로커 인증 모델이다[2,8].

브로커 인증 모델은 크게 순수 브로커 인증 모델과 단일 에이전트와 결합된 혼합 브로커 인증 모델로 구분된다. 순수 브로커 인증 모델의 경우 프로그램을 수정하여 중앙 집중형 관리를 하면서 모든 통신의 암호화를 통해 보안성을 유지한다. 따라서 사용자에게 클라이언트 사용법과 사용자 본인을 인증하는 방법을 숙지하도록 해야 한다. 에이전트와 결합된 형태의 모델의 경우 브로커 인증 모델을 기반으로 하여 프로그램 수정을 최소화하고 서비스 에이전트를 개발하여 중앙 집중식 관리를 실시하는 특징이 있다. 또한 에이전트를 통한 서버별 패스워드 관리 기능과 에이전트에 암호화를 추가함으로써 모든 구간의 암호화 서비스를 제공하며, 사용자 편리성 측면에서는 브로커 인증 모델과 동일한 편리성을 제공한다.

따라서 인터넷 및 전자상거래 환경에서 에이전트를 결합한 형태의 혼합형 인증 모델을 적용 하기 위해서는 다음과 같은 보안 요구사항을 만족해야 한다.

- 기밀성 : 전송되는 데이터는 암호화를 통해 제3자로의 공격으로부터 안전성을 유지할 수 있어야 한다.
- 무결성 : 신뢰받지 않은 네트워크를 통해 사용자의 인증 정보가 전송될 경우 공격자가 이를 위조하거나 변조할 수 있으므로 이에 대한 무결성이 보장되어야 한다.
- 부인봉쇄 : 인증 모델에 참여하는 모든 객체들에 대해 데이터의 전송 및 수신에 대한 부인을 막을 수 있는 보안 서비스가 제공되어야 한다.
- 사용자의 사전등록 : 사용자의 사전 등록 과정에서 사용자가 생성한 비밀스러운 정보가 공개되지 않으면서 이를 검증할 수 있어야 한다.
- 익명 사용자 : 익명 사용자의 경우 특별한 등록 과정 없이도 자신의 공개된 정보로 서비스 이용이 가능해야 하며, 이를 검증할 수 있어야 한다.
- 기밀연산 : 에이전트는 원격지에서 초기 비밀값을 노출시키지 않고 명령을 수행하도록 해야 한다.
- 관리 체계 : SSO의 인증 모델에서는 반드시 중앙 집중식 관리 체계가 필요하다. 이는 많은 사용자들의 인증 정보를 중앙에서 관리하여 효율성을 극대화 시키기 위해서이다. 또한 중앙 집중식 관리에서 통신량 감소를 위한 분산화가 필수적으로 요구된다.

4. SSO에 적용 가능한 브로커 인증 모델 제안

본 논문에서는 멀티 에이전트 시스템을 이용해 클라이언트와 인증 서버의 상호인증 후 클라이언트와 인증 서버 그리고 Kerberized server를 중심으로 이

루어지는 브로커 인증 모델을 제시하고자 한다. 멀티 에이전트 시스템의 경우 정보 전송을 위한 모바일 에이전트를 관리하는 AMS(Agent Management System)와 권한 정보 에이전트를 생성 관리하는 ANS(Agent Name Server)로 구분된다. 기존 시스템에서 단일 에이전트를 기반으로 인증이 수행되는 모델과 에서 모바일 에이전트만으로 정보의 전송과 권한 설정에 대한 정보를 관리에 대한 보안상 취약점을 보완하고자 모바일 에이전트와 기지 에이전트를 구분하여 관리할 수 있는 시스템이다. 이상의 시스템을 기반으로 제안방식은 다음과 같은 구성 개체로 이루어진다.

- AMS : 멀티 에이전트의 구성요소중의 하나로써 모바일 에이전트의 고유값을 생성하고 이를 ANS에 전송하는 개체로써 모바일 에이전트를 생성 및 관리하는 개체이다.
- ANS : AMS에서 생성된 모바일 에이전트에 해당되는 권한 정보 에이전트를 생성 관리하는 개체이다. 권한 정보 에이전트는 ANS에 저장되는 기지 에이전트로써 사용자의 권한 설정에 대한 정보를 포함하고 있다.
- 인증 서버 : 브로커 인증 모델에 참여하는 모든 객체들의 공개 고유값을 공개 디렉토리에 공개하고, 고유 ID와 Client/Server ticket을 생성하여 클라이언트와 Kerberized server에게 이를 전송하는 개체이다.
- 클라이언트 : 인증 서버로부터 Client ticket과 고유 ID를 발급받고 이를 기반으로 Kerberized server의 어플리케이션 서비스를 제공받고자 하는 개체이다.
- Kerberized server : 여러 어플리케이션 서비스를 관할하는 통합 객체로서 인증 서버로부터 Server ticket과 고유 ID를 발급 받고 이를 기반으로 클라이언트에 어플리케이션 서비스를 제공하는 개체이다.

- 제안 방식은 SSO에 요구되는 다양한 프라이버시 정보와 권한 정보를 자동화 소프트웨어인 모바일 에이전트와 기지 에이전트로 구분하여 수행함으로써 기존의 에이전트 인증 모델과 브로커 인증 모델의 결합된 혼합 모델에서 보안 취약점을 보완하기 위해 제안한다. 제안 방식은 클라이언트가 어플이케이션 서버로 접속하기 위해 인증 서버와의 상호 인증 과정을 수행하고 kerberized 서버와의 티켓 교환을 통해 안전한 어플이케이션 서비스 연결까지의 과정을 제안하였다.

4.1 시스템 계수

다음은 SSO에 적용 가능한 브로커 인증 모델을 위한 시스템 계수를 기술한다.

* (클라이언트: c, 인증 서버 : a, Kerberized server : k, AMS : AMS, ANS : ANS)

pw : 클라이언트가 생성하는 고유 비밀번호 ($g^{pw} = g^{pw_1} + g^{pw_2}$)

$*_p$, $*_q$: *의 공개키, 개인키

r_*, a_* : *의 객체가 생성한 의사난수

n : 공개키 암호 알고리즘의 모듈러

g : Z_n상에서 최대 위수를 갖는 원소

ID_* : *의 information data

E, EC : 공개키 암호 알고리즘, 대칭키 암호 알고리즘

Sig : 공개키 서명 알고리즘

$Userlevel$: 사용자 권한 정보 (사용자 레벨에 따른 권한 정보)

T_* : *가 생성한 타임 스탬프

H : 안전한 해쉬 알고리즘

M_A : 모바일 에이전트의 ID, 모바일 에이전트 생성 시간의 타임스탬프, 기능이 포함된 모바일 에이전트의 정보 메시지

$V_{*@}$: *에서 @으로 전송되는 암호화된 값

m_a : 인증서버의 인증 메시지

Server Ticket : 인증 서버가 생성한 서비스 ticket으로써 다음과 같은 내용을 포함하고 있다. (Ticket 요청자, ticket 응답자, 생성시간, 유효기간)

Client ticket : 인증서버에서 생성한 클라이언트 ticket으로써 다음과 같은 내용을 포함하고 있다. (Server Ticket + (ticket 요청자, ticket 응답자명, 생성시간, 유효기간))

4.2 제안방식의 구성

본 논문의 제안 방식은 다음과 같은 전제사항을 기반으로 사전단계를 제외하고 총 7 단계로 이루어

져있다.

가정사항

1) 사용자의 패스워드g^{pw}는 $g^{pw_1}+g^{pw_2}$로 이루어진다.
2) 인증서버의 공개키 a_p는 모든 개체에게 공개되어 있다.
3) 사용자의 권한정보 Level은 다음과 같다.
   - Level 1 : 해당 서비스의 임시 권한으로써 각 서비스의 읽기 권한만을 부여하고 가입 설정에 따른 서비스만을 접근할 수 있는 권한
   - Level 2 : 해당 서비스의 일반 사용자 권한으로써 무료 컨텐츠의 서비스(읽기 및 쓰기)를 제공 받을 수 있으며, 유료 컨텐츠의 사용시 별도의 권한 설정을 요구할 수 있는 권한
   - Level 3 : 해당 서버의 모든 서비스를 사용할 수 있으며 관리자만이 접속할 수 있는 서비스를 제외하고 모든 서비스에 대한 접근 권한
   - Level 4 : 사용자에게 전자상거래 서비스를 제공하는 서버를 관리하는 관리자 권한

[사전 단계] 클라이언트의 정보 전송 단계 : 클라이언트는 어플리케이션 서비스 연결을 위해 자신의 정보를 인증 서버에 전송하는 단계이다.

[단계 1] 멀티 에이전트의 생성 및 권한 설정단계 : 클라이언트의 접속 요구가 있을 경우 AMS는 모바일 에이전트를, ANS는 기지 에이전트를 각각 생성하고 권한을 설정하는 단계

[단계 2] 클라이언트의 인증 정보 및 권한 설정 정보 전송단계 : 단계 1에서 생성된 정보를 기반으로 하여 클라이언트의 인증 정보와 권한 설정 정보를 인증 서버에 전송하는 단계

[단계 3] 초기 등록 단계 : Kerberized Server가 인증서버에 자신의 고유 공개값을 등록하는 단계

[단계 4] **Kerberized Server**와 클라이언트 **ticket** 전송 단계 : Kerberized Server와 클라이언트에 전송할 서비스 ticket을 인증서버가 생성하고 이를 전송하는 단계

[단계 5] 클라이언의 서비스 요청 단계 : 클라이언트는 Client ticket을 확인하고 세션키를 이용해 암호화된 메시지로 서비스를 요청하는 단계

[단계 6] **Kerberized Server**의 **ticket** 및 세션키 획득 단계 : Kerberized Server는 인증 서버에서 전송된 정보를 통해 세션키를 생성하여 Server ticket을 획득하는 단계

[단계 7] 클라이언트의 서비스 연결 및 승인 단계 : Kerberized Server는 클라이언트로부터 전송된 정보를 기반으로 암호화된 메시지를 복호화 한 뒤 클라이언트의 해당 어플리케이션 서비스 연결을 승인하는 단계

4.3 프로토콜

[사전 단계] 클라이언트의 정보전송 단계

클라이언트는 인터넷 환경에서 서비스 받고자 하는 어플리케이션 서비스의 연결을 위해 패스워드 정보와 ID_c를 인증 서버에 전송하는 단계이다.

[단계 1] 멀티 에이전트의 생성 및 권한 설정단계

① 클라이언트의 정보를 전송받은 인증 서버는 전송된 정보에서 클라이언트의 g^{pw_1}, 클라이언트의 ID_c를 저장하고 모바일 에이전트 생성 요구 메시지인 M_a, ID_a, g^{pw_1}을 AMS에 전송한다.

② AMS는 인증 서버로부터 전송받은 ID_a, M_a, g^{pw_1}에서 서버가 요구하는 모바일 에이전트를 생성하기 위한 메시지 M_a를 확인하고 다음과 같이 모바일 에이전트의 초기 코드를 생성하기 위한 값들을 계산한다.

$$M_A = H(M_a \| ID_a) \oplus H(M_{info_{AMS}}),$$

$$K = g^{r_{AMS_1}} \bmod n$$

$$S_A = \frac{g^{pw_1}}{(K + H(M_A))} \bmod n,$$

$$R_A = K \| M_A$$

이상의 값들을 계산한 AMS는 해당 에이전트의 초기값을 다음과 같이 생성한다.

$$M_{code} = E_{ANS_p}(R_A, S_A, M_{info_{AMS}})$$

- $M_{info_{AMS}}$는 AMS가 모바일 에이전트를 생성한 타임 스탬프, 생성한 AMS 이름, 모바일 에이전트의 ID가 포함되어 있다.

에이전트의 초기 코드를 생성한 AMS는 M_{code} (에이전트 초기코드), M_a, $M_{info_{AMS}}$, T_{AMS}를 ANS에 전송한다.

③ ANS는 전송받은 모바일 에이전트의 초기 코드 값을 개인키로 복호화하여 R_A와 S_A를 획득한다. 획득한 R_A에서 K를 추출하여 임시저장 한 뒤 전송된 M_A 정보를 확인하고 이에 대한 권한 레벨 정보를 포함하는 서명정보를 생성한다.

$$\text{Userlevel} = Sig_{ANS_s}(M_{code} || T_{ANS} || (Level-information))$$

[단계 2] 클라이언트의 인증 정보 및 권한 설정 정보 전송단계

① AMS는 [단계 1]의 ②에서 생성된 정보를 기반으로 M_{code}, $E_{c_p}(r_{AMS_1}, H(M_A))$, T_{AMS}, S_A를 클라이언트에 전송한다.

가. 클라이언트는 AMS로부터 전송받은 S_A를 다음과 같은 과정을 통해 검증한다.

$$K' = g^{(r_{AMS_1})'} \bmod n,$$

$$S_A{}' = \frac{g^{pw_1}}{(K' + H(M_A))} \bmod n$$

$S_A{}'$ = S_A일 경우 클라이언트는 자신의 패스워드의 또 다른 정보인 g^{pw_2}를 생성하여 자신의 ID와 함께 ANS에 이를 전송한다.

② ANS는 모바일 에이전트의 권한정보인 Userlevel, g^{pw_2}를 인증 서버에 전송한다.

③ 인증 서버는 전송된 모바일 에이전트의 권한 정보를 획득하고 전송된 모바일 에이전트의 초기 값과 ANS의 권한 정보에서 추출한 코드 값으로 전송 메시지의 무결성을 검증한 후 사용자의 패스워드를 구성한 후 클라이언트의 패스워드를 이용한 공개정보 ($P_C(= g^{pw})$, T_C)을 인증서버의 공개 디렉토리에 등록한다.

[단계 3] 초기 등록 단계

Kerberized Server는 인증 서버에 고유 공개값을 등록하기 위하여 다음과 같은 계산을 수행하여 (P_k, T_k)를 인증 서버에 전송한다.

- Kerberized server : Kerberizded server는 r_k를 선택 후 공개된 시스템 계수를 이용하여 P_k, ID_k를 인증서버에 전송한다.

$$P_k = g^{r_k} \bmod n$$

[단계 4] Kerberized Server와 클라이언트 **ticket** 전송 단계

인증 서버는 전송된 (P_k, T_k), (P_c, T_c)를 공개 디렉토리에 저장하고 다음과 같은 과정을 수행한다.

① 클라이언트에서 전송된 (P_c, T_c)를 확인한 뒤 클라이언트의 Client ticket을 생성한 뒤 다음을 계산하여 클라이언트에게 V_{ac}, h_{a_1}, T_{a_1}를 전송한다.

$$K_{a_1} = H(P_c * g^{pw_2} \,||\, ID_c) \bmod p,$$

$$\alpha_{a_1} = g^{r_{a_1}} \bmod p$$

$$Z_{a_1} = KH_{\alpha_{a_1}}(Clientticket || P_a) \oplus H(m_a || P_c) \bmod p$$

$$S_{a_1} = \frac{\alpha_{a_1}}{(Z_{a_1} + \alpha_{a_1})} \bmod q,$$

$$h_{a_1} = H(\alpha_{a_1} || T_{a_1} || V_{ac})$$

$$V_{ac} = EC_{K_{a_1}}(m_a || Clientticket || \alpha_{a_1} || r_{a_3})$$

② 인증 서버는 Kerberized server의 Server ticket을 생성한 뒤 다음을 계산하여 Kerberized server에게 V_{ak}, h_{a_2}, T_{a_2}를 전송한다.

$$K_{a_k} = H(P_k * g^{r_3} \,||\, ID_k) \bmod p,$$

$$\alpha_{a_2} = g^{r_{a_3}} \bmod p$$

$$Z_{a_2} = KH_{\alpha_{a_2}}(Serverticket || P_a) \oplus H(m_a || P_k) \bmod p$$

$$S_{a_2} = \frac{\alpha_{a_2}}{(Z_{a_2} + \alpha_{a_2})} \bmod q,$$

$$V_{ak} = EC_{K_{ak}}(m_a || Serverticket || \alpha_{ak} || Userlevel || r_{a_2})$$

$h_{a_2} = H(\alpha_{a_2} || T_{a_2} || V_{ak})$

[단계 5] 클라이언트의 서비스 요청 단계

클라이언트는 단계 4에서 전송 받은 V_{ac}, h_{a_1}, T_{a_1}에서 다음과 같은 검증 과정을 거쳐 전송 데이터에 대한 검증을 수행한다.

$$K_{a_1}' = H(g^{pw} * g^{r_a} || ID_c) \mod p,$$

$K_{a_1}' \stackrel{?}{=} K_{a_1}$이면 전송된 V_{ac}를 복호화하여 m_a, T_{a_1}, $Clientticket$, α_{a_1}를 확인 한 뒤 이를 기반으로 α_{a_1}를 검증한다.

$\alpha_{a_1}' = \alpha_{a_1}$이면 $Z_{a_1}' = Z_{a_1}$

이상의 검증 과정을 거쳐 클라이언트는 인증 서버로부터 전송되어온 Client ticket을 획득한다. Client ticket을 획득한 클라이언트는 서버에 전송하고자 하는 메시지를 세션키 C_{ck}를 다음과 같이 생성한다.

$$C_{ck} = H(\alpha_{a_1} \oplus \alpha_{a_2}) \mod p$$

이후 클라이언트는 세션키 C_{ck}로 어플리케이션 서비스 요청 메시지를 암호화 한 뒤 V_{ck}, T_c를 Kerberized server에 전송하여 서비스를 요청한다.

$$V_{ck} = EC_{C_{ck}}(M_{request} || Clientticket)$$

[단계 6] Kerberized Server의 ticket 및 세션키 획득 단계

Kerberized server는 인증 서버로부터 전송 받은 V_{ak}, h_{a_2}, T_{a_2}를 이용하여 다음을 검증한다.

$K_{a_2}' \stackrel{?}{=} K_{a_2}$, $\alpha_{a_2}' \stackrel{?}{=} \alpha_{a_2}$, $Z_{a_2}' \stackrel{?}{=} Z_{a_2}$

이상의 내용이 올바른 경우 클라이언트와의 세션키 C_{ck}를 생성하고 Server ticket을 획득한다.

$$C_{ck} = H(\alpha_{a_1} \oplus \alpha_{a_2}) \mod p$$

[단계 7] 클라이언트의 서비스 인증 연결 및 승인 단계

Kerberized server는 단계 5에서 클라이언트로부터 전송된 V_{ck}, T_c에서 V_{ck}를 단계 6에서 생성한 세션키 C_{ck}로 복호화 하여, 클라이언트가 접속하고자 하는 어플리케이션 서비스에 연결한다. 연결이 이루어진 후 해당 클라이언트의 권한 정보 설정 후 해당 클라이언트에 어플리케이션 서비스를 제공한다.

이상의 프로토콜 진행을 통해 SSO에 적용 가능한 브로커 인증 모델을 제안하였다. 이상의 프로토콜 진행을 다음과 같이 그림 2로 표현할 수 있다.

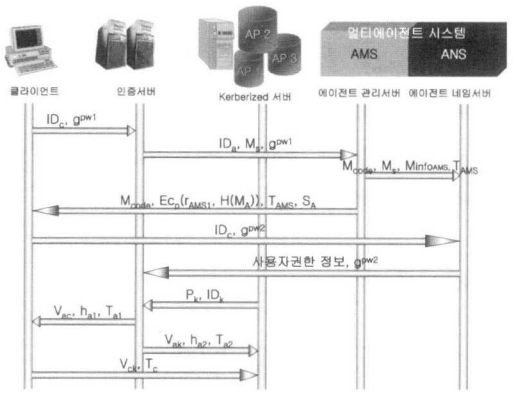

그림 2. SSO에 적용 가능한 브로커 인증 모델

5. 제안방식 분석

본 장에서는 4장에서 제시했던 제안 방식을 3장에서 제기된 보안 요구사항을 기반으로 분석하고자 한다.

제안 방식은 에이전트와 브로커 인증 모델이 결합한 혼합형 인증 모델과 비교해 볼때 단일 에이전트의 사용에 따른 보안상 취약점을 보완할 뿐만 아니라 다양한 에이전트의 적용이 가능한 멀티 에이전트를 사용하여 적용에 대한 효율성을 높이는 방법을 제시하였다. 또한 3장에서 제시한 보안 요구사항에 기반하여 다음과 같은 특징을 가지고 있다.

- 기밀성 : 브로커 인증 모델의 참여 객체들은 이산대수 문제에 근거한 공개 정보와 ID를 이용한 세션키 설정으로 제 3자의 공격으로부터 안전한 데이터 전송이 가능하다.
  - 사용자의 비밀번호 pw의 구성을 g^{pw_1}과 g^{pw_2}로 구성하여 각각의 패스워드 정보를 프로토콜의 내부 정보로 사용함으로써 사용자의 비밀정보에 대한 기밀성을 유지할 수 있다.
  - ID를 이용한 세션키 설정 : 각 단계에서는 참여 객체의 비밀 정보를 대칭키 및 공개키 암호 알고리즘으로 암호화하여 전송하며, 수신 개체는 사전 공유된 키를 이용해 복호화한 뒤 전송 내용을 비교함으로써 상호 기밀성을 유지한다.
- 무결성 : 신뢰받지 않은 네트워크를 통해 사용자의 인증 정보가 전송될 경우 안전한 해쉬 함수와 타임스탬프를 통해 무결성 서비스를 제공할 수 있다.
  - 티켓의 무결성 서비스 : 클라이언트와 Kerberized Server는 Keyed 해쉬된 Z를 이용해 티켓에 대한 무결성 보안 서비스를 제공 받을 수 있다.

클라이언트:

$$Z_{a_1} = KH_{a_{a_1}}(Clientticket \| P_a) \oplus H(m_a \| P_c) \bmod p$$

Kerberized Server :

$$Z_{a_2} = KH_{a_{a_2}}(Serverticket \| P_a) \oplus H(m_a \| P_k) \bmod p$$

- 부인봉쇄 : 인증 모델에 참여하는 모든 객체들은 자신만의 비밀 의사난수인 r을 선택하여 서비스 정보가 전송된다. 선택된 의사난수는 공개키를 생성할 때 사용된다. 따라서 각 클라이언트 마다 고유의 공개키가 존재하고 이를 기반으로 전송 데이터의 부인 봉쇄가 가능하다.
- 사용자의 사전등록 : 사용자의 사전 등록 과정에서 사용자가 생성한 비밀 정보인 pw값이 공개되지 않으면서도 이를 기반으로 생성한 P의 값을 이용해 사용자의 안전한 사전 등록이 가능하다.

사용자의 비밀정보 pw는 $g^{pw} = g^{pw_1} + g^{pw_2}$로 분할 구성된다. 분할된 정보는 서비스 요청에 따른 공개키 등록 과정에서 (P_c $(= g^{pw})$, T_c)로 공개 디렉토리에 공개되어 사용자의 비밀값이 공개되지 않으면서, 안전하게 사전 등록 과정이 수행된다.

- 익명 사용자 : 익명 사용자가 서비스를 제공받고자 한다면 인증 서버에서 할당해주는 고유 ID와 익명 사용자가 전송한 공개 정보에 의해 익명 서비스가 가능하다.
- 기밀연산 : 본 논문에서 제안된 멀티 에이전트를 이용한 방식은 멀티 에이전트 시스템 개체의 의사난수와 사용자의 패스워드 정보를 이용한 R, S는 기밀 연산 서비스를 제공한다.

모바일 에이전트의 초기 코드값에 포함되는 $S_A = \frac{g^{pw_1}}{(K + H(M_A))} \bmod n$, $R_A = K \| M_A$는 AMS에서 생성하는 난수 r을 기반으로한 k와 사용자의 비밀 정보 값 중의 하나인 g^{pw_1}을 기반으로 생성되므로 모바일 코드에 대한 기밀연산을 수행할 수 있다.

- 관리 체계 : 제안된 방식은 관리체계인 중앙집중식 관리 체계 방식을 이용하고 있다. 따라서 사용자의 관리에 대한 효율성 뿐만 아니라 멀티 에이전트 시스템(ANS, AMS)를 활용하여 인증 서버에 대한 오버헤드를 최소화 하고자 하였다.

이상의 내용을 2장에서 분석한 기존 방식과 비교해 표 1과 같이 정리하였다.

표 1. 제안방식과 기존방식 비교 분석

	브로커 인증 모델	에이전트 인증 모델	게이트웨이 인증 모델	스크립 인증 모델	제안된 인증 모델
기밀성	X	△	△	X	O
무결성	O	X	△	X	O
부인 봉쇄	X	X	X	△	O
사용자 사전 등록	△	△	△	O	O
익명 사용자	X	X	X	O	O
관리 체계	O	O	X	X	O

< X : 취약, △: 부분, O : 안전>

6. 결 론

최근 각 기업들의 인트라넷 시스템과 웹서비스가 대폭 확장됨에 따라 보안 솔루션 시장은 급속히 확대되고 있다. 특히, SSO를 도입하면 각각의 시스템마다의 인증 절차를 밟지 않고도 사용자에 부여된 1개의 계정만으로 다양한 시스템에 접근할 수 있어 사용자 편의가 대폭 높아지고, 관리자 입장에서도 인증정책의 변경이나 권한 설정이 수월해져 관리 비용 및 수고를 크게 덜 수 있다.

최근의 SSO 기술은 인증된 사용자에게 시스템 정보 및 자원에 접근할 수 있는 권한은 물론 중요 접근제어 권한까지 부여하는 공개키 기반구조로 발전되는 추세이다.

따라서 본 논문에서는 SSO에 적용 가능한 인증 모델을 제시하였다. 이는 SSO에서 가장 중요한 부분중의 하나인 인증 부분에서 필요한 여러 가지 보안 요구사항을 제시하고 이를 만족할 수 있는 사용자의 편리성과 관리자의 중앙집중식 관리를 통해 기존의 SSO 인증 모델중의 하나인 브로커 모델의 보안적인 취약점 뿐만 아니라 멀티에이전트를 결합한 혼합 형태의 인증 모델을 제시하였다. 본 논문에서는 단순히 모바일 에이전트와 기지 에이전트로 구분하여 보안 프로토콜을 제안하였으나, 클라이언트의 특성, 어플리케이션의 서비스에 따라 여러 형태의 에이전트를 결합할 수 있다는 특징이 있다.

제안 방식은 향후 발전해 가는 인터넷 환경을 고려하였을 경우 지속적인 연구가 필요한 분야라 할 수 있다.

참 고 문 헌

[1] B.C. Neuman, Theodore Ts'o. Kerberos, "An Authentication Service for computer Network," IEEE Communication, 32(9) : 33-38, September, 1994

[2] RFC 1510, Public key Cryptography for Initial Authentication in Kerberos, draft-ietf-cat-kerberos-pkinit-14.txt.

[3] Alfred J. Menezes, Paul C.van Oorschot, Scott A. Vanstone "HANDBOOK of APPLIED CRYPTOGRAPHY", CRC, 1996.11.

[4] 이만영, 김지홍, 류재철, 송유진, 염흥렬, 이임영 "전자상거래 보안 기술", 생능출판사 1999.8.

[5] 최용락. 소우영, 이재광, 이임영 "통신망정보보호", 그린출판사, pp343-393, 2001

[6] http://technet.oracle.co.kr/docs/oracle78/network32x/NWANO233/ch1.htm

[7] http://www.entrustkorea.co.kr/

[8] http://www.krnet.or.kr

서 대 희 (Dae-Hee Seo) 학생회원

2001년 2월 : 동신대학교 전기전자공학부(학사)

2003년 2월~현재 : 순천향대학교 전산학과(석사)

2004년 5월 : 순천향대학교 전산학과 박사과정

<관심분야> 암호 이론, 컴퓨터 보안

이 임 영 (Im-Yeong Lee) 정회원

1981년 8월 : 홍익대학교 전자공학과 졸업

1986년 3월 : 오사카대학 통신공학과 석사

1989년 3월 : 오사카대학 통신공학과 박사

1989년 1월~1994년 2월 : 한국전자통신연구원 선임연구원

1994년 3월~현재 : 순천향대학교 정보기술공학부 부교수

<관심분야> 암호이론, 정보이론, 컴퓨터 보안