Journal of The Korea Society of Computer and Information
Vol. 21 No. 4, pp. 31-38, April 2016

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.4.031

User Authentication Technology using Multiple SSO in the Cloud Computing Environment

Min-Hee Cho*, Eun-Gyeom Jang**, Yong-Rak Choi***

Abstract

The interface between servers and clients and system management in the cloud computing environment is different from the existing computing environment. The technology for information protection. Management and user authentication has become an important issue. For providing a more convenient service to users, SSO technology is applied to this cloud computing service. In the SSO service environment, system access using a single key facilitates access to several servers at the same time. This SSO authentication service technology is vulnerable to security of several systems, once the key is exposed.

In this paper, we propose a technology to solve problems, which might be caused by single key authentication in SSO-based cloud computing access. This is a distributed agent authentication technology using a multiple SSO agent to reinforce user authentication using a single key in the SSO service environment. For user authentication reinforcement, phased access is applied and trackable log information is used when there is a security problem in system to provide a safe cloud computing service.

▸ Keyword : Cloud computing, SSO, User authentication, Security policy

I. Introduction

클라우드 컴퓨팅은 '소프트웨어', '스토리지', '서버', '네트워크' 등의 자원을 가상화한 컴퓨팅 환경 서비스이다. 이 서비스는 기존의 인터넷 기반 컴퓨팅 환경인 그리드 컴퓨팅이나 유틸리티 컴퓨팅과는 또 다른 개념으로 분리된다. 이러한 클라우드 컴퓨팅은 기존의 인터넷 기반의 컴퓨팅에 비하여 사용자들에게 편리성과 활용가능성을 제공한다. 클라우드 컴퓨팅 환경에서는 사용자들에게 신뢰되고 원활한 서비스가 제공되어야 한다. 클라우드 컴퓨팅의 활성화를 위해서는 사용자 인증을 통한 정보보호 서비스가 필수 요소이고, 관리되는 정보의 외부 유출의 위험으로부터 보호되어야 한다. 또한 사용자의 시스템 사용의 편

리성과 보안성도 필수요소로 제공되어야 한다[1].
이러한 클라우드 컴퓨팅은 현재 모바일 기기들의 급속한 발전과 더불어 유·무선 클라우드 컴퓨팅 서비스 환경으로 바뀌었다. 모바일 클라우드 컴퓨팅은 기존의 클라우드 컴퓨팅과 각종 무선기기의 결합으로 얻어지는 이점을 기반으로 한 서비스를 제공한다. 무선기기는 스마트폰 및 노트북, 넷북, PDA, 태블릿 PC 등을 대표 모델로 볼 수 있다.

이러한 유·무선 클라우드 컴퓨팅에서 요구되는 사항을 해결하기 위해 다방면으로 연구되고 있다. 첫 번째로 부각되고 있는 기술은 사용자의 접근 및 관리의 용이성과 편리성 서비스를 위해 단일키를 활용한 SSO 서비스(Single-Sign On Service)이다. SSO 서비스는 단일키를 활용하여 여러 컴퓨터 시스템을 추가 정보 없이 접근이 가능하도록 하여 사용자에게는 편리한 기능을 제공한다. 그러나 사용자 영역의 로컬 시스템의 접근 환경에 따른 보안적인 측면과 단일키의 유출에 의한 다중 시스템 노출의 위험성을 가지고 있어 문제가 되고 있다. 또한 SAML(Security Assertion Markup Language)을 활용한 SSO 서비스는 XML 기반의 사용자

• First Author: Min-Hee Cho, Corresponding Author: Eun-Gyeom Jang
*Min-Hee Cho(chominhee@lycos.co.kr), Dept. of Computer Science, Daejeon University
**Eun-Gyeom Jang(jangeg@jangan.ac.kr), Dept. of Internet Communication, Jangan University
***Yong-Rak Choi(yrchoi@dju.ac.kr), Dept. of Computer Science, Daejeon University
• Received: 2016. 02. 15, Revised: 2016. 03. 10, Accepted: 2016. 04. 06.

권한 제어 프레임 워크이다[2]. 이것은 플랫폼의 거래 파트너들이 인증 정보, 권한부여정보, 프로파일 정보를 안전하게 교환할 수 있도록 설계 된 것으로 기업 내부 또는 기업 간의 SSO를 제공하고 기반 보안 인프라에 종속되지 않는다. SAML SSO는 공격자의 컴퓨터로부터 표적 시스템과 그 시스템이 속한 네트워크에 과다한 데이터를 보냄으로써 시스템과 네트워크의 성능이 급격하게 저하하여 DOS 공격에 취약하다[3].

이러한 시스템 관리 및 접근의 위험성으로부터 클라우드 컴퓨팅의 시스템 보호 및 효율적인 관리를 위해 본 논문에서는 2장에서 클라우드 컴퓨팅 관련 기술을 분석하고 3장에서는 보안 강화된 접근 기술을 제안한다. 4장에서는 제안 기술을 시험하여 성능 결과를 논하고 5장에서 결론으로 논문을 구성하였다.

II. Literature Review

1. Cloud Computing

1.1 Main Characteristics and Models of Cloud Computing

1) Main Characteristics of Cloud Computing

클라우드 컴퓨팅 서비스의 전체 구성은 그림 1과 같이 유·무선 네트워크 기기들의 접근을 통해 다양한 서비스를 제공하고 있다.

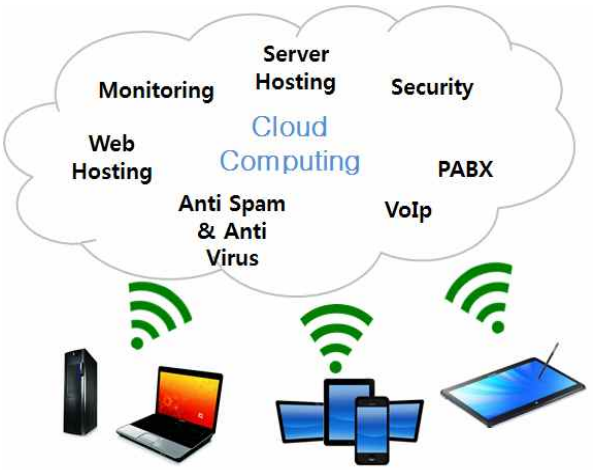

Fig.1. Cloud Computing Service

NIST에서 정의된 클라우드 컴퓨팅의 주요 특성은 다음과 같이 5가지로 정리할 수 있다[4].

① On-demand self-service : 서비스 제공자와 직접적인 연동 요구도 없이 필요시에는 언제나 자동으로 네트워크 스토리지와 같은 컴퓨팅 능력을 일방적으로 준비할 수 있다.

② Broad network access : 사용자 플랫폼(모바일, 노트북 및 PDA등)에 의한 네트워크에 가용성이 있거나 표준 메커니즘을 통한 접근 능력이 있다.

③ Resource Pooling : 서비스 제공자의 컴퓨팅 자원은 다중 모델을 사용하여 모든 사용자들에게 제공되도록 저장되어 있으며, 사용자의 요구에 따라 서로 다른 물리적이거나 가상적 자원들이 할당과 재 할당된다. 사용자는 일반적으로 제공된 자원의 정확한 위치에 대한 정보를 알수 없고 통제 할 수 없으나 높은 권한을 가진 사용자나 관리자는 보다 높은 정보를 규정가능하다.

④ Rapid elasticity : 서비스를 언제든지 아주 빠르게 확대 또는 축소 그리고 구매할 수 있는 탄력적 특성이다.

⑤ Measured service : 서비스 유형에 따른 적절한 측정 능력을 이용하여 자원 사용을 최적화 시키며, 자원 사용을 모니터 및 제어하고 기록하여 서비스 제공자와 서비스 사용자 사이에 투명성을 제공한다.

2) Cloud Computing Service Model

클라우드 컴퓨팅 서비스 모델은 Cloud Software as a Service(Saas), Cloud Platform as a Service(PaaS), Cloud Infrastructure as a Service(IaaS)로 구성된다[5].

- SaaS는 사용자가 클라우드 인프라 상에서 클라우드 서비스 제공자의 애플리케이션을 사용하는 서비스를 의미한다. SaaS 사용자에게는 서비스의 환경설정만 제한적으로 제공될 뿐 클라우드 인프라 등에 대한 관리 및 제어가 불가능하다.
- PaaS는 서비스 제공자에 의해 지원되는 프로그래밍 언어와 도구들을 이용하여 사용자가 클라우드 인프라에서 애플리케이션 개발이 가능하도록 제공해주는 서비스를 의미한다. 사용자는 클라우드 인프라, 네트워크, 서버, 운영체제, 스토리지에 대한 관리/제어 권한은 없지만 사용자가 구성한 애플리케이션과 호스팅한 애플리케이션의 시스템 환경 구성 관리 권한을 가질 수 있다.
- IaaS는 사용자가 소프트웨어(OS나 애플리케이션 포함)를 개발하거나 동작시킬 경우 프로세서, 스토리지, 네트워크 그리고 다른 기본적인 컴퓨팅 자원들을 제공하는 서비스를 의미한다.

1.2 Cloud Computing Security Threats

클라우드 컴퓨팅 환경에서는 모든 데이터 및 소프트웨어가 중앙 집중화되어 관리되기 때문에 해커들의 주요 공격대상이 될 수 있다. 클라우드 컴퓨팅 보안 요구 사항을 도출하기 위해 CSA에서 클라우드 컴퓨팅 구성요소별 보안 위협을 정의하였다[6]. (위협 1) 클라우드 컴퓨팅의 오용과 비도덕적인 사용, (위협 2) 불안전한 인터페이스와 응용 프로그래밍 인터페이스, (위협 3) 악의적인 내부자, (위협 4) 기술 공유 문제, (위협 5) 데이터 유실 또는 유출, (위협 6) 계정 또는 서비스 하이재킹, (위

협 7) 알려지지 않은 위험 프로파일.

클라우드 서비스 구성요소별 보안 위협 정의를 보면 PaaS의 접근제어가 가장 큰 위협요소이다. 클라우드 서비스 제공자는 안정적인 PaaS 기반 서비스를 제공하기 위해서는 허가된 이용자만이 IT 자원에 접근하고, 서비스를 이용할 수 있도록 보장해야한다[7].

- 인증 : 아이디/패스워드 인증, 전자서명 인증 등 인증 수단을 이용하여 사용자에 대한 신원 확인을 수행한다. 또한 인증 빈도 및 횟수에 의한 문제점은 다수의 사이트와 다수의 서비스를 통합인증하는 SSO(Single Sign-On) 형태의 인증 기술을 통해 해결할 수 있다.
- 접근제어 : 사용자 계정별로 접근을 허용하는 영역 및 권한을 분리한다.

2. SSO Authentication

SSO 시스템은 사용자가 어느 시스템에 로그인이 되어 있으면 다른 시스템에서는 사용자의 정보를 공유하여 별도의 로그인 과정이 없는 인증방법이다. SSO 개념은 개방형인증 기술인 Open ID의 한 예이며 이러한 사용자 인증 방식은 다수의 사이트 및 서비스를 통합하여 서비스하는 클라우드 컴퓨팅 환경에 적합하다. SSO의 장점을 정리해 보면 다음과 같다[8].

① 사용자 계정 및 정보를 통합적으로 관리.
② 사용자는 단한번의 로그인으로 여러 사이트를 사용할 수 있는 편리성.
③ SSO 설계 자체에는 암호화와 인증으로 보안관리.
④ 외부 네트워크에서도 SSO를 통해 서비스 사용 및 제공

에이전트 기반의 SSO는 클라이언트에 새로운 모듈을 설치하여 중앙에서 통합 관리하여 SSO를 지원하는 방법이 있다. 그리고 쿠키를 사용하여 SSO를 지원하는 모델의 상용 제품으로는 Netegrity의 'SiteMinder', Entrust의 'getAccess', OpenNetwork의 'DirectorySmart'와 Oracle의 SSO 등이 있다[9].

그림 2는 중앙 통합관리 SSO 서비스[10], 그림 2는 쿠키 기반의 SSO 서비스 모델[11]이다.

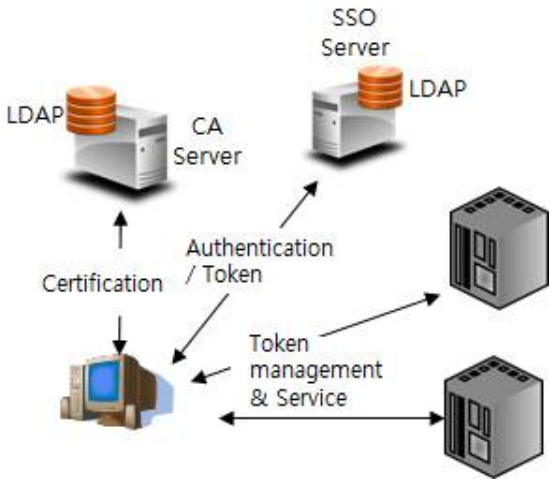

Fig. 2. SSO service of central integrated management

중앙 통합관리 SSO 메커니즘은 사용자가 인증서를 요청하여 발급 받아 보관한 후 SSO 서버를 통해 인증을 요구하고, 토큰을 사용하여 SSO 서버에 접속하는 서비스 일괄을 통합 화면을 통해 SSO 서버가 수행한다. SSO 서버를 중심으로 인증서 발급 서버, 각종 응용들의 웹 서버들을 사용자가 SSO를 통해 접근한다.

SSO 모델의 전체 흐름 중 인증서 발급, LDAP(Lightweight Directory Access Protocol) 저장 부분은 기존의 방식과 유사하다. 포탈 서비스 화면에서 같은 도메인 내의 서버들에 분산된 서비스로 이동하기 위해 다른 서비스 메뉴를 선택하면 인증 통합 메뉴에서 재인증 절차를 거치지 않고 SSO 서버에서 승인 과정을 거치도록 되어 있다[12].

그러나 SSO 시스템은 인증서 정보를 유지하기 위한 수단으로 쿠키가 사용되므로 쿠키의 도청이나 변조시 문제가 발생될 수 있으며 정해진 시간 내에 동일한 도메인 내에서만 사용 가능하다. 또한 클라이언트에 별도의 소프트웨어를 설치해야 하고 별도의 수정 및 부가 작업을 필요로 한다. 그리고 미리 정해진 간격으로 토큰이 유일하므로 지속적인 연결을 맺어야 되는 어려움이 있다.

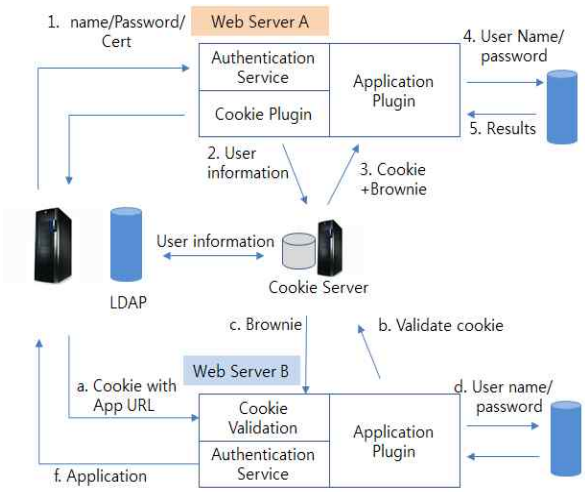

Fig. 3. Cookie-based SSO(Oracle's SSO)

그림 3은 쿠키 기반의 SSO 서비스 모델인 Oracle SSO의 상위 레벨 구조이다. 쿠키 서버(Cookie Server)가 중심이 되어 중앙에서 쿠키를 발행하고 검증한다. 세부적인 쿠키의 구조나 기능은 시스템들마다 다르나, 전체적인 구조는 대부분의 시스템이 비슷한 모습을 하고 있다.

3. Safe Mobile Cloud Computing Authentication

모바일 클라우드 컴퓨팅의 특징은 사용 주체가 모바일이므로 각종 모바일 디바이스를 통해 클라우드 컴퓨팅 서비스를 이용할 수 있다는 점과 근본적인 컴퓨팅 서비스의 개념은 기존의 클라우드 컴퓨팅과 같지만 모바일 기반이 추가됨으로써 클라우드 컴퓨팅 기술의 상용화가 급속도로 확산되고 있다는 점이다.

모바일 클라우드 컴퓨팅은 클라우드 컴퓨팅과 같이 IaaS, PaaS, SaaS 등의 서비스 종류로 구분하고 개방여부에 따라 Private, Public, Hybrid가 있다. 모바일 클라우드 컴퓨팅의 또 다른 분류로는 기업용 시스템 서비스 개념의 Private 클라우드와 개인 사용자를 위한 Public 클라우드가 있다. 표 1은 모바일 클라우드 컴퓨팅 주요 비즈니스 모델이다.

개인을 대상(B2C)으로 한 비즈니스는 모바일 기기를 이용하여 데이터를 자유롭게 저장하고 관리하는 서비스를 선보이고 있고, 기업을 대상(B2B)으로 한 비즈니스는 오피스 인프라를 제공하여 고객 기업의 생산성을 향상 시킬 수 있는 서비스를 w 지원한다. 실제 모바일 클라우드 컴퓨팅 모델은 다음과 같다.

▪ Apple의 'Mobile Me': 모바일 클라우드 컴퓨팅을 통하여 사용자의 메일, 주소록, 캘린더 정보의 유지 관리 기능을 제공하고 있다. 사용자의 모바일 기기인 아이폰, 맥, PC 등과 웹 사이트 간에 자동적으로 사용 동기화 기능이 가능하며, 모든 정보들의 동기화가 유지되로독 푸시 기술을 적용하고 있다.

▪ MS의 'My Phone': 개인의 Windows moblie phone 에 있는 컨텐츠에 대한 온라인 액세스를 제공하고 있다. 이 서비스는 웹 사이트에 암호화를 통한 보호 기능과 함께 Windows moblie phone에 있는 중요한 정보(연락처, 일정, 약속, 작업정보, 메시지, 비디오 등)에 대한 동기화 기능을 제공하고 있다.

▪ Soonr의 'Soonr': 스마트폰 상에서 클라우드 기반 모바일 오피스 서비스를 제공하는 모바일 클라우드 애플리케이션이다. Doc, PPT, XSL 등의 오피스 파일을 클라우드 서버 측에서 실행하고, 모바일 단말에서는 별도의 오피스 프로그램 없이 해당 파일의 실행 결과를 알 수 있다. Remote UI 기술로 서버가 실시간으로 오피스 파일을 실행하고, 결과 UI를 원격에 위치한 단말로 전송하는 원리다.

Table 1. Mobile cloud computing business model

Object	Field	Contents
B2C	Content storage and management	Service that stores and management the content in the cloud
	Synchronization Services	Service to share in real time and used in mobile devices(N-Screen)
	Office Services	Freely create and edit Web
B2B	Providing mobile office infrastructure	Using the cloud corporate intranet services
	Hardware Solutions	Provide new solutions that leverage the cloud

III. Safe Cloud Computer Authentication Reinforcement Technique

제안 시스템은 무선 네트워크 환경에서 클라우드 컴퓨팅 서비스의 가장 큰 보안 위협요소인 인증과 접근제어 방법을 강화한 기법으로, 인증 방법은 SSO Agent를 사용하고 공격자로부터 사용자의 정보 유출을 방지하기 위한 기술을 적용하였다. 그리고 핵심 정보 및 개체 접근은 'Security Reinforcement System'에서 제공하는 사용자 보안 등급에 따른 권한과 정책을 적용한 서비스를 제공한다. 또한 보안 강화 인증을 위해서 권한 접근에 대한 강화된 로그 정보 및 인증 정책을 적용한다.

1. Dispersion SSO Agent Authentication Technique

기존 SSO 인증은 단일 에이전트가 모든 인증 과정을 일괄적

으로 처리하는 방법을 사용한다. 그러나 단일 SSO 인증 에이전트는 공격대상이 되어 외부의 공격으로 노출될 경우 여러 응용 서버에 문제를 발생시킨다. 이러한 문제를 보완하기 위해 본 연구에서는 SSO 에이전트를 나누어 사용자 인증 정보를 분할하는 다중 에이전트 인증방법을 사용한다. 이 기법은 인증에 참여하는 n개의 서버 중 n-1개의 사용자 인증정보가 공격자에게 노출되어도 공격자는 사용자의 인증 정보를 획득할 수 없도록 한다. 그림 4는 사용자의 서버에 접근하기 위한 사용자의 접속 과정과 단계별 절차이다.

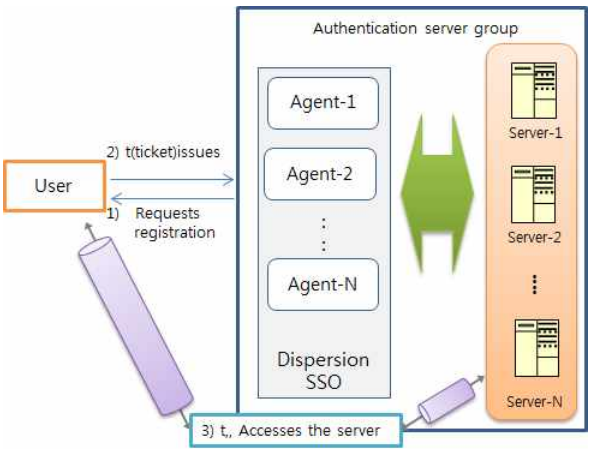

Fig. 4.Dispersion SSO agent authentication model

다음은 제안된 인증 기법에서 사용하는 주요 용어에 대한 설명이다.

용어 정의
U: 사용자, AS: 인증서버, ID_U: 사용자 U의 인식자, PW_U: 사용자 U의 패스워드, AS_PU: 인증서버의 공개키, AS_PR: 인증서버의 개인키, x, y: 사용자와 인증서버가 생성한 랜덤값, F_P: 유한체, P: 유한체 상의 소수, g: 원시 소수, SSK: 스마트카드의 비밀값, UI: 사용자 정보, SI: 보안 정보, S_K: 세션키, $E_*()$: *의 키를 이용하여 암호화, D_*: *의 키를 이용하여 복호화, $\oplus$: XOR 연산, $\|$: 연접, $h()$: 해쉬 함수

가. 사용자 등록

1) 등록 요청

U : 사용자, SI : 보안 정보,

$U \rightarrow \{S_i\}_{1 \le i \le n}$: 등록 요청 메시지

사용자(U)는 인증서버 집합 $\{S_i\}_{1 \le i \le n}$ 에게 등록 요청 메시지를 보낸다.

2) 등록 응답

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

사용자로부터 등록 요청 메시지를 받은 인증서버 집합 중 임의의 S_i는 사용자에게 현재 시스템에서 사용되고 있는 공용 파라미터를 전달한다.

3) 인증 정보 생성 및 전달

$\{S_i\}_{1 \le i \le n}$로부터 공용 파라미터를 전달받은 사용자는 $\{S_i\}_{1 \le i \le n}$에 등록할 인증 정보를 다음과 같이 생성한다.

3-① 패스워드(p/w) 및 비밀키(K) 선택

$X \in {}_R Z_q^*, BDF(p/w) = gu, K = g^X u$

사용자는 자신이 기억할 패스워드와 난수 $X(\in {}_R Z_q^*)$를 선택한다.

그리고 패스워드를 Z_P^* 상의 곱 연산에 대한 위수가 q인 원소로 대응시키는 함수 $BDF(\cdot\ )(= h(p/w)^2 : h$는 해쉬함수$)$에 패스워드를 입력하여 $g_U(= BDF(p/w))$를 계산한다. $K = g_U^X (\mathrm{mod}\, P)$도 계산한다.

3-② 비밀키 확인자(h_K) 생성

$H(K) = h_k : H(\cdot\ )$는 일방향 해쉬함수

K에 대한 단방향 해쉬 함수 값 h_K를 계산한다.

3-③ $\{S_i\}_{1 \le i \le n}$의 부분키 생성

$k_i \in {}_R Z_q^* (i = 1, \cdots, n-1), k_n = X - \sum_{i=1}^{n-1} k_i (\mathrm{mod}\, q)$

사용자는 각 인증 서버가 저장할 서버 부분키를 선택한다.

3-④ 사용자는 서버 집합에서 사용자를 인증할 때 필요한 서명키 쌍 x(서명키), y(서명 확인키)를 생성한 뒤, x를 K로 암호화하여 $D(= E_K(x))$를 생성한다.

3-⑤ 패스워드에서 유도된 사용자 공개키 $\Theta = g_U^k (\mathrm{mod} P, K' \equiv K \bmod q)$를 생성한다.

3-⑥ 부분 인증 정보 전달

$U \rightarrow S_i (i = 1, \ldots\ldots, n) : (ID, k_i, \Theta, x, D, h_k)$

각각의 인증 서버에 위에서 생성한 부분 인증 정보를 전달한다.

4) 사용자 등록

각각의 인증 서버 S_i는 사용자의 부분 인증 정보 (ID, k_i, Θ, x, D)를 각각 사용자 ID별로 저장한다.

나. 사용자와 서버간의 키 교환 과정

사용자(U)가 $\{S_i\}_{1 \le i \le n}$내의 특정 서버 $S_j(\in {}_R\{S_i\})$와 인

증된 키 교환을 수행하는 과정을 살펴보자.

① $U \rightarrow \{S_i\}_{1 \le i \le n} : Q_{U-S}$

사용자는 $a', a \in {}_R Z_q^*$를 선택하여, Q_{U-S}를 $\{S_i\}_{1 \le i \le n}$에게 브로드캐스팅한다.

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

② $\{S_i\}_{1 \le i \le n} : i \ne j \rightarrow U : R_{G-U}, S_j \rightarrow U : R_{S_j-U}$

$\{S_j\}_{1 \le i \le n} : i \ne j$의 각각의 인증서버 S_i는 R_{G-U}를 사용자에게 전달하고, S_j는 $b \in {}_R Z_q^*$를 선택한 뒤 R_{S_j-U}를 생성하여 사용자에게 전달한다.

$R_{G-U} = \{S_i \| g_U^{a \cdot k_i} \| h_k\}_{1 \le i \le n : i \ne j}$

$R_{S_j-U} = S_j \| g_U^{a \cdot (K_j + b)} \| g_U^{a' \cdot b} \| D \| h_K$

③ 사용자는 R_{S_j-U}와 a', a를 가지고 $g_U^{a \cdot k_j}, g_U^b$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 K를 생성했는지 확인한 뒤 S_j와의 세션키 K_S를 계산한다.

$K' = (\prod_{i=1}^{n} g_U^{a \cdot k_i})^{\frac{1}{a}}, K_S = (g_U^b)^{a + k' \pmod q}$

④ S_j는 사용자가 K_S를 계산하는 동안 사용자의 θ를 이용하여 세션키 K_S를 계산한다.

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

⑤ 사용자는 자신이 정당한 사용자라는 것과 키 확인을 위해 K를 이용해 D를 복호화해 x를 얻은 뒤 x를 기지고 Q_{U-S}에 서명하고 M을 생성하여 브로드캐스팅을 한다.

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

⑥ M을 전달받은 $\{S_i\}_{1 \le i \le n}$는 각기 x를 가지고 M이 정당한지 확인한다.

S_j는 M이 정당하면 사용자와 같은 세션키를 공유하고 있음을 확인할 수 있다.

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

(SSO Agent-1, Server-1),… (SSO Agent-N Server-N)의 쌍을 각각 별도로 구성하여, 총 N개의 인증서버 그룹을 형성한다. 사용자는 N개의 인증 서버 그룹 중 하나에 등록하게 된다. 사용자 등록 후 기존의 SSO처럼 자신이 속한 인증 서버 그룹과 한번의 인증과정을 통해서 티켓 t를 전달 받으면 t에 포함된 자신의 접근 통제 규칙에 따라 SSO Agent에 포함된 응용서버에 접근이 가능해진다. SSO Agent의 경우 공격자가 Agent를 공격해서 사용자 인증정보 획득하여도 인증서버 그룹의 다른 사용자 인증 정보가 있어야 접근할 수 있다. 즉 공격자는 사용자를 위장하여 다른 시스템에 접근할 수 없다.

IV. Performance Test and Analysis

1. Performance Test

제안 기술의 실험은 SSO 에이전트의 처리 속도와 인증을 위한 패킷 및 인증 처리 시간을 중심으로 실험 하였다. 실험 환경은 네트워크는 100Mbps의 기반 시스템 환경에서 Sun Fire과 인텔 i7 모델을 SSO 에이전트와 사용자 시스템으로 구성하여 실험하였다.

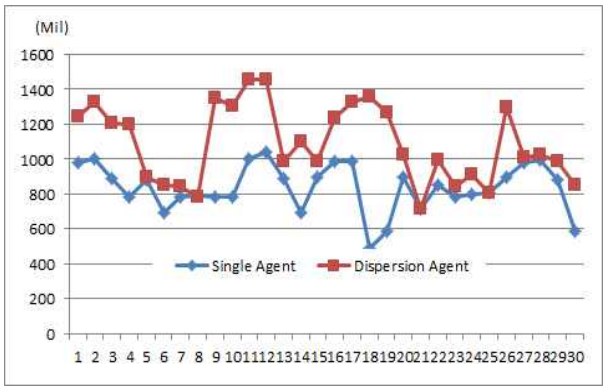

Fig. 5 Authentication processing time analysis of SSO agent

실험은 단일 SSO 에이전트 인증과 분산 SSO 인증을 실험하여 처리속도와 공격 복잡도를 추출하였다. 또한 적용한 분산 SSO 에이전트는 5개로 분산하여 인증을 처리하였다. 그리고 그림5와 같이 같은 네트워크 트래픽 환경에서 실험하기 위해 동일 네트워크와 시간대에 실험하였다.

2. Performance Analysis

에이전트 인증 분석을 위해 단일 SSO 인증과 분산 SSO 인증처리 시간을 비교하기 위해, 인증 처리시간은 네트워크 트래픽의 속도와 관련된 결과를 동일하게 적용하였다. 그러나 실험 적용시에 네트워크 트래픽에 차이가 발생하여 약간의 오차가 있는 것을 볼 수 있다.

- 단일 SSO 인증 처리 시간= NTTime+ STime
- 분산 SSO 인증 처리 시간 = NTTime + ASTime + CJTime

단일 SSO 인증은 네트워크 트래픽(NTTime)과 시스템처리

속도(STime)의 시간을 갖는다. 그러나 분산 SSO 인증 기법은 네트워크 트래픽(NTTime)과 N개의 에이전트 시스템처리속도(ASTime), 인증정보조합시간(CJTime)에 의해 처리시간이 결정된다. 인증 처리 시간의 차이는 실험시의 네트워크 트래픽과 인증정보조합시간(CJTime)에 의해 차이가 나는 결과를 얻었다. 실험 결과, 분산 에이전트의 처리 속도가 평균 0.65의 처리 결과의 차이를 얻을 수 있다. 시스템의 처리 속도도 영향이 있지만 많은 영향을 미치지는 못하였다. 에이전트를 5개 적용한 실험결과 평균 0.65의 결과를 얻었다면 N개의 에이전트를 적용하면 약 N*0.13의 시간이 더 지연되는 결과를 얻는다. N개의 SSO 에이전트를 적용한 모델에 대한 공격의 시간 복잡도는 다음과 같다.

- 한 개의 SSO 에이전트 공격 시간 : AttackTime
- N 개의 SSO 에이전트 공격 시간 :

$$\sum_{1}^{N} Attack1_{Time} + Attack2_{Time} + \cdots + AttackN_{Time}$$

- N개의 분산 SSO 에이전트 공격 시간 :

$$(\sum_{1}^{N} Attack1_{Time} + Attack2_{Time} \cdots + AttackN_{Time}) + Attack_{Time}!$$

단일 SSO 에이전트 공격시간이 10이라할 때, N개의 SSO 공격시간은 10N이 된다. 그러나 제안한 분산 SSO 에이전트 공격은 N 개의 에이전트 공격 시간에 N개의 정보 조합이 필요함으로 '10N + 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2'의 처리 시간이 필요하다.

V. Conclusion

본 논문에서는 무선 클라우드 컴퓨팅 환경에서 사용자 인증 기법에 대하여 연구하였다. 본 논문의 제안 기술은 기존의 SSO와 VPN을 적용한 사용자 인증에 의한 클라우드 컴퓨팅 접근의 보안을 강화한 사용자 인증 기법을 제안한다. 제안한 기법은 단일 SSO 에이전트를 이용한 접근을 분산 SSO 인증 기법이다. 분산 SSO 에이전트를 이용한 인증은 단일 SSO 사용자 인증을 강화한 기법으로 하나의 SSO 인증 에이전트가 노출되었을 경우 접근 가능한 여러 응용 서버가 위험에 노출되기 때문에 분산 SSO 에이전트를 적용하였다. 분산 SSO 인증은 여러 개의 SSO 에이전트에서 사용자 인증 정보를 분산하여 관리하는 기법을 적용하여 공격자로부터 복잡하고 안전한 시스템을 유지할 수 있다.

제안 시스템을 실험한 결과, 처리 속도는 인증을 위한 사용자 정보 조합 처리에 지체되는 결과를 얻었다. 그러나 네트워크의 속도에 의해 차이는 있지만, 공격자로부터 시스템의 안전성을 높은 효율성을 얻을 수가 있었다. 사용자 인증 처리에 소요되는 시간은 단일 SSO 보다는 높지만 현재 및 향후 기가급의 네트워크 트래픽 환경이라면 문제가 되지 않을 것으로 본다.

REFERENCES

[1]Wikipedia,http://en.wikipedia.org/wiki/Cloud_computing
[2] Google, "SAML Single Sign On(SSO) Service for Google Apps" Google Inc. 2006.
[3] OASIS "Profile for the OASIS Security Assertion Language(SAML)V2.0" OASIS Standard, 15 March 2005.
[4] NIST, NIST Cloud Computing Standards Roadmap, Special Publication, p.291, 2011.
[5] P.Mell, T.Grance, "The NIST Definition of Cloud Computing, National Institute of Standards and Technology", National Institute of Standards and Technology, ver.15, July. 2010.
[6] Cloud Security Alliance, "Top Threats to Cloud Computing v1.0", Mar. 2010.
[7] Dongwon Jeong, “A Standard Reference Model for Semantic Interoperability in Cloud Computing”, Journal of The Korea Society of Computer and Information, Vol 17 No. 8, August 2012.
[8] X/Open Single Sign On Pluggable Authentication Modules, The Open Group, 1997.
[9] Google, “SAML Single Sign On(SSO) Service for Google Apps” Google Inc., 2006.
[10] B. Pfitzmann, B. Waidner, "Token-based web Single Sign-On with Enabled Clients", IBM Research Report RZ 3458(93844), Nonmember 2002.
[11] V. Semar, "Single Sign-On Using Cookies for Web application. Proceedings", IEEE 8th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise(WET ICE 99), 1999.
[12] Eun-Gyeom Jang, "A Study on Access Control Through SSL VPN-Based Behavioral and Sequential Patterns", Journal of The Korea Society of Computer and Information, Vol 18 No. 11, November 2013.

Authors

Min-Hee Cho received a Ph.D. Candidate in Daejeon University 2010. It has an interest in system security and Cloud Computing.

Eun-Gyeom Jang received a Ph.D in Daejeon University in 2007. Hi is currently a Professor in the Department of Internet Communication Jangan University.
It has an interest in mobile communications, system security and Computer Forensics.

Yong-Rak Choi received a Ph.D. degree from Chung-Ang University Korea. He worked as a senior researcher at Electronics and Telecommunications Research Institute (ETRI), Korea from 1982 to 1986.
In 1986, he joined Daejeon University and is now a professor at Department of Computer Engineering, Daejeon University. He was a vice-chair of Korea Institute of Information Security & Cryptology, His research interests are computer communication security, computer forensics, and DRM.