The Journal of Digital Policy & Management 2013 Jul; 11(7): 215-223

멀티 클라우드 환경을 위한 OpenID 기반의 사용자 인증 기법

위유경*, 곽진**
순천향대학교 정보보호학과 정보보호응용및보증연구실*, 순천향대학교 정보보호학과**

OpenID Based User Authentication Scheme for Multi-clouds Environment

Yukyeong Wi*, Jin Kwak**
ISAA Lab, Dept of Information Security Engineering, Soonchunhyang University*
Dept of Information Security Engineering, Soonchunhyang University**

요 약 클라우드 컴퓨팅이 활성화됨에 따라 다양한 클라우드 서비스가 보급되고 있다. 하지만 각각 서로 다른 클라우드 서비스를 사용하려면 각각의 서비스에 개별적인 사용자 인증과정을 수행해야만 한다. 따라서 절차가 번거로울 뿐만 아니라 거듭된 인증 과정 수행으로 인한 비밀번호 노출, 각 클라우드 서버마다 사용자의 인증 정보를 소유해야 하는 데이터베이스의 과부하, 서비스마다 각기 다른 인증방법과 개인정보 입력방법으로 인한 피싱공격 등의 보안 문제점이 발생할 가능성이 높다. 따라서 다양한 클라우드 서비스를 사용하고자 할 때 사용자의 자격증명을 신뢰할 수 있는 ID제공업체에 의해 티켓을 제공받아 안전하게 멀티 클라우드 환경에 적용이 가능한 OpenID 기반의 사용자 인증 기법을 제안한다.

주제어 : 멀티 클라우드, OpenID, 사용자 인증, 접근제어, 공개키 암호 방식

Abstract As cloud computing is activated, a variety of cloud services are being distributed. However, to use each different cloud service, you must perform a individual user authentication process to service. Therefore, not only the procedure is cumbersome but also due to repeated authentication process performance, it can cause password exposure or database overload that needs to have user's authentication information each cloud server. Moreover, there is high probability of security problem that being occurred by phishing attacks that result from different authentication schemes and input scheme for each service. Thus, when you want to use a variety of cloud service, we proposed OpenID based user authentication scheme that can be applied to a multi-cloud environment by the trusted user's verify ID provider.

Key Words : Multi-clouds, OpenID, User Authentication, Access Control, Public Key Cryptosystem

* 이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2012-010886).
Received 31 May 2013, Revised 2 July 2013
Accepted 20 July 2013
Corresponding Author: Jin Kwak(Soonchunhyang University)
Email: jkwak@sch.ac.kr

ISSN: 1738-1916

1. 서론

클라우드 컴퓨팅 기술의 발전과 함께 다양한 클라우드 서비스가 등장하고 있으며, 최근에는 하나의 클라우드 서비스만 사용하는 것이 아닌 여러 개의 클라우드 서비스를 함께 사용하는 멀티 클라우드 환경이 주목을 받고 있다. 이에 따라 사용자 중심으로 네트워크 생태계가 변화하면서 기존의 단일 클라우드 환경에서 ID와 패스워드 기반 사용자 인증 방법을 대체할 새로운 인증 수단이 필요하게 되었다. 또한 다양한 클라우드 서비스가 빠르게 보급화 되면서 각 서비스마다 각기 다른 인증방법과 개인정보 입력방법으로 사용자 인증과정을 수행하므로 피싱공격, 하이재킹 등의 보안 문제점이 발생할 가능성이 증가하였다. 따라서 통일된 사용자의 신원정보를 제공할 수 있고, 보다 편리하고 간편한 클라우드 서비스 접근 절차가 요구되었다[1,2].

OpenID는 이러한 요구사항들을 만족시켜주는 URL 기반의 단일 사용자 신원정보를 제공하고, 사용자 중심의 분산형 인증체계를 제공하는 규격이다. 본 논문에서는 OpenID를 다양한 클라우드가 집약된 환경에 적용하기 위한 사용자 인증 기법을 제안하였다. 본 논문에서 제안하는 기법은 사용자가 각기 다른 클라우드 서비스에 접속하고자할 때 클라우드 서비스 제공업체가 OpenID를 통해 사용자 인증을 처리하는 기법이다[3].

2. 관련연구

2.1 멀티 클라우드 컴퓨팅

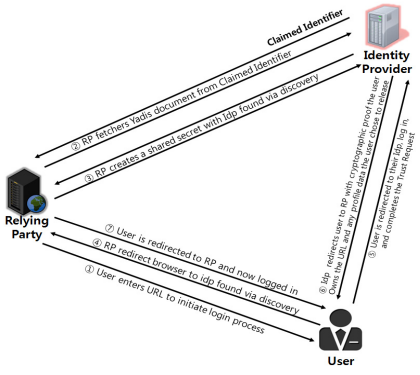
최근 태블릿 PC 및 스마트폰과 같은 고성능 휴대용 컴퓨팅 기기들의 보급률이 증가함에 따라 처리해야할 데이터양 또한 급속도로 증가하여 기존의 단일 클라우드 환경으로는 감당하기 어려운 경우가 발생하고 있다. 또한 모바일 기기의 특성상 무선네트워크, 배터리 소모 등의 성능 저하에 민감하기 때문에 다양한 클라우드 서비스 제공업체 중에서 가장 최적의 클라우드 자원을 가지고 있는 제공업체를 선택할 수 있도록 하는 기법이 요구되고 있다[4].

멀티 클라우드란 하나의 클라우드 서비스를 사용하는 것이 아닌 여러 개의 클라우드 서비스를 함께 사용하는 것을 말하며, 인터 클라우드 또는 클라우드의 클라우드(cloud of clouds)로도 혼용되고 있다. 이는 특정 클라우드 서비스의 사고로 인한 데이터 손실 혹은 전송 차단 등의 위험성을 줄이는 효과를 가지며, 벤더 락과 같은 문제도 해결함으로써 전반적인 성능 향상을 얻을 수 있다[5].

따라서 현재 멀티 클라우드 환경을 이용하여 하나의 클라우드 이용에 대한 종속성과 신뢰성을 극복하는 연구들이 진행되고 있다.

2.2 OpenID

OpenID는 사용자 중심의 새로운 ID 시스템으로 웹사이트처럼 URL 행태의 ID로 자신을 식별하게해주는 분산형 공개 표준 기술이다[6]. OpenID 시스템은 누구든지 추가로 소요되는 비용 없이 이용할 수 있으며, 인터넷 이용자들은 자신의 온라인 ID를 관리하기 위하여 하나의 사이트에 의존할 필요가 없다. 즉, 각 사이트에서 제공하는 서비스를 이용하기 위하여 사이트마다 생년월일, 이름, 주소등과 같은 개인정보를 입력하고 ID와 패스워드를 생성할 필요 없이 OpenID 협력 사이트에서 OpenID를 이용하여 서비스를 제공받을 수 있다. OpenID를 이용하면 이용자는 각 사이트마다 자신이 사용하던 ID와 패스워드를 따로 관리 하거나 분실할 위험성이 없다. 또한 서비스를 제공하는 업체의 입장에서는 ID와 패스워드 관리를 위한 비용들을 줄일 수 있으며, 사용자 인증 서비스와 SSO(Single Sign-On) 등의 아웃소싱 효과를 가져올 수 있다[7,8]. OpenID는 사용자와 OpenID를 제공하는 사이트(IDP : Identity Provider), OpenID 정보를 사용하는 협력 사이트(RP : Relying Party)로 구성되어 있으며, 다음의 [Fig. 1]에서 OpenID 인증 프로토콜의 기본적인 동작 과정을 나타낸다.



[Fig. 1] The basic OpenID protocol flow

3. 문제점 및 보안 요구사항 분석

본 절에서는 멀티 클라우드 환경에서의 사용자 인증 방식의 문제점을 분석한다. 또한 분석한 내용을 바탕으로 클라우드 환경에서 사용자 인증의 문제점을 해결하기 위한 보안 요구사항을 도출한다.

3.1 문제점

3.1.1 개인정보 관리 문제

멀티 클라우드 환경에서는 각각의 클라우드 서버마다 사용자의 개인정보 및 인증정보를 소유하게 된다. 이로 인해 사용자는 필요 이상의 개인정보를 각각의 클라우드 서비스마다 제공하게 된다. 따라서 여러 곳에 분산되어 있는 개인정보를 관리하기 어렵고, 개인정보가 전송되는 클라우드 서버가 많기 때문에 개인정보가 노출될 가능성이 높다. 그렇기 때문에 통신선로상의 안전이 보장되지 않는다면 사용자 개인정보의 기밀성을 보장할 수 없다.

3.1.2 로그인 정보 위/변조

클라우드 서버에는 다수의 사용자가 접근한다. 악의적인 공격자가 자신의 신분을 속이고 서버에 접근하여 사용자의 로그인 정보를 무단으로 위/변조할 가능성이 있다. 이에 따라 사용자 로그인 정보의 무결성을 보장할 수 없다[9].

3.1.3 서로 다른 사용자 인증 절차

각각의 서로 다른 클라우드 서비스를 이용하려면 해당되는 클라우드 서비스의 정책에 따라 별도의 사용자 인증과정을 수행해야만 한다. 따라서 수행되는 절차가 번거로울 뿐만 아니라 반복적으로 수행되는 패스워드 입력으로 인해 정보 노출에 대한 문제점이 발생할 가능성이 존재한다.

3.1.4 접근제어 관리 문제

멀티 클라우드 환경은 사용자가 다수의 클라우드 서버에 접근이 가능하다. 따라서 인가되지 않은 사용자의 특정 클라우드 서버 무단 접근은 다른 제 3의 클라우드 서버에도 무단으로 접근이 가능해진다. 이는 악의적인 사용자가 하나의 클라우드 서버뿐만 아니라 여러 클라우드 서버에 악성 데이터를 무단으로 업로드하는 문제로 이어질 가능성이 있다. 또한 클라우드 서버의 가용성을 침해하는 등 심각한 보안 문제점이 발생할 가능성이 존재한다[10].

3.2 보안 요구사항

3.2.1 기밀성

클라우드 서버의 티켓 정보는 기밀성이 보장되어야 한다. 클라우드 서버의 티켓 정보는 사용자의 OpenID생성에 대한 정보를 가지고 있기 때문에, 만약 공격자에 해당 정보가 유출될 경우에 다른 클라우드 서비스까지 접근 가능한 OpenID가 공개될 가능성이 있다. 이를 위해 클라우드 환경의 통신에 사용되는 파라미터는 정당한 사용자만이 확인할 수 있어야 하며, 파라미터의 출처 및 수신지, 횟수, 길이 또는 통신선로 상의 트래픽 특성에 대하여 공격자가 알지 못하게 해야 한다[11].

3.2.2 무결성

클라우드 사업자가 ID제공업체(IDP : Identity Provider)로부터 사용자 인증 서비스를 제공받기 위해 사용되는 서비스 인증값은 무결성이 보장되어야 하며, 클라우드 환경에서 데이터베이스에 저장 또는 네트워크를 통해 전송되는 정보가 위변조 및 파괴되지 않도록 해야 한다. 만약 서비스 인증값의 정보가 유출된다면 위조, 삭제 및 변조를 통해 다른 제 3의 클라우드 서비스 가용성까지 침해할 수 있는 추가적인 보안문제를 야기할 수 있다. 따라서 전송받은 파라미터의 위조 및 변조를 감지하기 위해 해쉬함수 연산 및 전자서명 기법 등을 이용하여 무결성을 보장해야 한다[12].

3.2.3 사용자 인증

멀티 클라우드 환경에서의 사용자 인증기능은 각각의 클라우드 서비스마다 인증절차가 서로 다르기 때문에 그에 따른 각각의 서비스를 이용하기 위해서 서로 다른 인증절차를 하나로 통일할 필요가 있다. 또한 서비스를 이용하기 위해 접근하는 사용자가 전송한 메시지 또는 파라미터의 출처가 명확하고, 해당 값의 신분이 정당한 사용자라는 것을 검증할 수 있어야 한다[13,14].

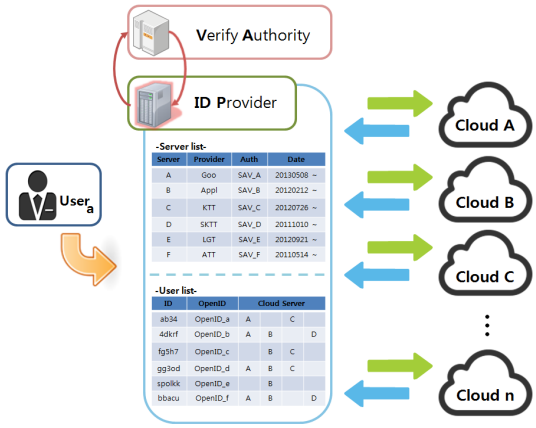
3.2.4 접근제어

클라우드 서버내의 정보 자원에 대한 읽기 및 변경 등의 모든 접근 행위에 대해 그 권한을 명확하게 구분하여 허가되지 않은 접근 시도를 사전에 차단할 수 있도록 하는 접근제어 기능이 필요하다. 운영체제에서는 접근통제 기능을 사용해야 하며, 네트워크에서는 침입차단 시스템을 사용하여 접근통제의 수준을 향상 시킬 수 있다. 또한 정당하지 않은 사용자는 클라우드 서비스를 이용할 수 없도록 해야 하며, 서버 자체에 접근을 할 수 없도록 해야 한다[15].

4. 제안방식

본 논문에서 제안하는 방식은 사용자가 이용을 원하는 클라우드 서버로부터 티켓을 발급받아 ID제공업체에 전송하여 저장한다. 또한 ID제공업체가 신뢰할 수 있는 자격검증기관으로부터 해당 사용자의 자격을 검증받아 최종적으로 해당 사용자의 OpenID를 발급한다. 클라우드 서버에서는 해당 사용자의 인증 요청 시에 사용자의 OpenID를 ID제공업체에게 요청하여 신뢰할 수 있는 사용자의 신원정보를 받을 수 있고, 클라우드 서버로부터 발급받은 티켓이 사용자의 OpenID에 존재하지 않는다면, 해당 사용자의 서버접속을 사전에 차단하여 사용자 정보를 제공받을 수가 없어 서비스 자체가 불가능하게 된다. 또한 추가적인 클라우드 서비스를 사용하고자 할 때 해당 클라우드 서버로부터 티켓을 발급 받아 ID제공업체에 등록을 하여 기존의 OpenID에 추가만하여 보다 안전하고 편리하게 멀티 클라우드 환경에 접근하도록 구성하였다.

다음의 [Fig. 2]는 본 논문에서 제안하는 멀티 클라우드 환경을 위한 OpenID 인증 프로토콜의 개념도를 나타낸다.



[Fig. 2] Overview of the proposed scheme

4.1 용어정리

제안하는 인증 프로토콜에서 사용되는 주요 시스템 파라미터는 다음과 같다.

- SID_* : 클라우드 서버 *의 ID
- SPW_* : 클라우드 서버 *의 패스워드
- CS_* : 클라우드 서버 *
- IDP : OpenID 서비스 제공업체
- VA : 사용자 검증기관
- SAV_* : 클라우드 서버 *의 openID 서비스 인증값
- ID_* : 사용자 *의 ID
- PW_* : 사용자 *의 패스워드
- $ticket_*$: 클라우드 서버 *가 생성한 티켓 정보
- $OpenID_*$: 사용자 *의 OpenID 정보
- N_* : *이 생성한 난수값
- T : 타임스탬프값
- E_{PK_*} : *의 공개키 암호화
- $H(\bullet)$: 해쉬함수 연산

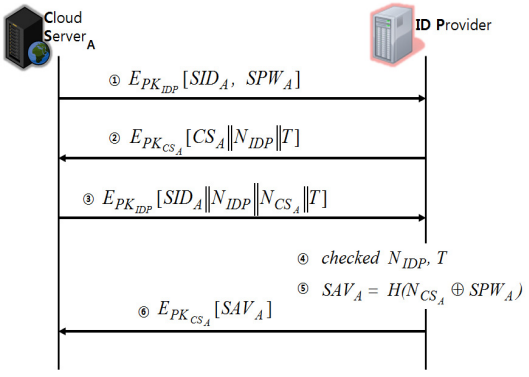
4.2 제안 프로토콜

본 절에서는 제안하는 OpenID 인증 프로토콜의 동작을 세 가지 단계로 구분하여 설명한다. 먼저, 클라우드 사업자가 해당 ID제공업체로부터 서비스를 제공받기위해 클라우드 사업자 정보를 ID제공업체에 등록하는 OpenID

서비스 등록 과정, 클라우드 서비스 사용자가 클라우드 서버로부터 티켓을 발급받아 ID제공업체에게 등록하고, 해당 사용자의 OpenID를 발급받는 OpenID 발급 과정, 발급받은 OpenID를 기반으로 클라우드 서버에 접속하여 인증하는 사용자 인증 과정으로 나뉜다. 이를 위해 본 제안방식의 서로 다른 클라우드 서비스 사업자는 공통된 ID제공업체로부터 사용자의 신원정보를 증명 받을 수 있는 통합 프레임워크임을 가정한다.

4.2.1 OpenID 서비스 등록 과정

다음의 [Fig. 3]는 클라우드 사업자가 해당 ID제공업체로부터 서비스를 받기위해 등록하고, 서비스 인증값을 발급받는 과정이다. ID제공업체 서버는 클라우드 서버로부터 난수값과 패스워드 정보를 연산하여 해당 클라우드 사업자 고유의 서비스 인증값을 생성한 후 발급받는 과정을 보여준다.

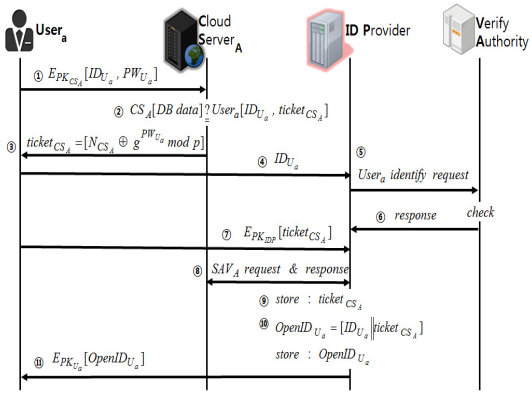


[Fig. 3] OpenID service registration phase

- step 1 : 클라우드 사업자는 ID제공업체로부터 서비스를 받기위해 IDP 서버에 접속한다.
  ($E_{PK_{IDP}} = [SID_A, SPW_A]$)
- step 2 : IDP 서버는 해당 클라우드 사업자의 서비스 인증값 생성을 위해 IDP 서버 자체의 난수값과 타임스탬프값을 연접 연산하여 클라우드 서버의 공개키로 암호화하여 전송한다.
  ($E_{PK_{CS_i}} = [CS_A \| N_{IDP} \| T]$)
- step 3 : 클라우드 서버는 난수값을 생성하여 IDP 서버로부터 전송받은 값을 복호화 하여 얻은 IDP의 난수값과 타임스탬프값과 함께 IDP의 공개키로 암호화하여 전송한다. 이는 해당 클라우드 서버가 정당한 사업자라는 것을 증명하게 된다.
  ($E_{PK_{IDP}} = [SID_A \| N_{IDP} \| N_{CS_A} \| T]$)
- step 4 : IDP 서버는 전송받은 정보를 복호화 하여 IDP 서버 자체의 난수값과 타임스탬프값을 통해 해당 클라우드 서버의 정당성을 확인한다.
  (N_{IDP}, T)
- step 5 : IDP 서버는 클라우드 서버로부터 전송받은 난수값과 패스워드를 연산한 결과에 해쉬연산을 한 서비스 인증값을 생성하여 저장한다.
  ($SAV_A = H(N_{CS_A} \oplus SPW_A)$)
- step 6 : IDP 서버는 생성한 서비스 인증값을 클라우드 서버에 전송하여 해당 클라우드 서버의 서비스 이용을 위한 등록 단계를 마무리 한다.
  ($E_{PK_{CS_i}}[SAV_A]$)

4.2.2 OpenID 발급 과정

다음의 [Fig. 4]는 클라우드 서버에 접속하기 위한 OpenID를 발급받기 위한 과정이다. 사용자는 클라우드 서버로부터 티켓을 발급받고, 발급받은 티켓을 ID제공업체에게 등록한다. ID제공업체는 신뢰할 수 있는 자격검증기관으로부터 해당 사용자를 검증받아 사용자 고유의 OpenID를 발급하는 과정을 보여준다.



[Fig. 4] OpenID publication phase

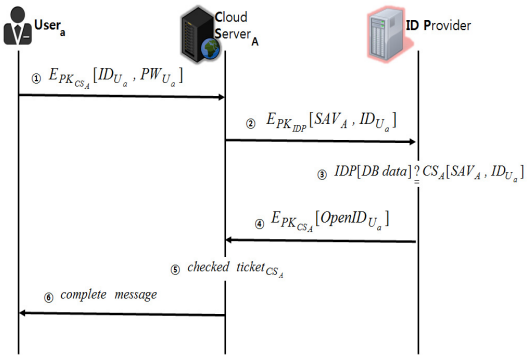
- step 1 : 사용자는 사용을 원하는 클라우드 서비스

에 접속한다.($E_{PK_{CS_i}} = [ID_{U_a}, PW_{U_a}]$)

- step 2 : 요청을 받은 클라우드 서버는 해당 사용자에게 OpenID 발급이 가능한 자사의 클라우드 서비스 티켓 유무를 확인한다. 티켓이 있다면 서비스 응답으로 인증을 마무리한다.
  ($CS_A[DBdata] \stackrel{?}{=} User_a[ID_{U_a}, ticket_{CS_A}]$)
- step 3 : 클라우드 서버는 해당 사용자에게 티켓을 생성하여 발급한다. 클라우드 서버의 티켓 정보($ticket_{CS_A}$)는 클라우드 서버가 생성하는 난수값과 사용자의 패스워드 정보를 지수승한 모듈러 값의 연산을 통해 생성한다.
  ($ticket_{CS_A} = [N_{CS_A} \oplus g^{PW_{U_a}} \bmod p]$)
- step 4 : 사용자는 클라우드 서비스 사용자 인증을 위한 OpenID 발급을 위해 ID제공업체 서버에 접속한다.(ID_{U_a})
- step 5 : IDP는 신뢰할 수 있는 자격검증기관(VA)에 해당 사용자의 자격증명을 요청한다.
  ($User_a\ identify\ request$)
- step 6 : 자격검증기관은 해당 사용자의 신원정보를 확인하여 ID제공업체에 검증결과를 전송한다.
- step 7 : ID제공업체는 사용자에게 사용을 원하는 클라우드 서비스에서 발급받은 티켓을 요청하여 전송받는다.
  ($E_{PK_{IDP}} = [ticket_{CS_A}]$)
- step 8 : ID제공업체는 전송받은 티켓 정보를 통해 자사의 ID 서비스에 가입되어 있는 클라우드 서버인지 확인한다.
  ($SAV_A\ request\ \ response$)
- step 9 : ID제공업체는 전송받은 사용자의 클라우드 티켓을 데이터베이스에 등록한다.
  ($store: ticket_{CS_A}$)
- step 10 : ID제공업체는 사용자의 해당 클라우드 서비스 티켓을 첨부하여 OpenID를 생성한 뒤 저장한다.($OpenID_{U_a} = [ID_{U_a} \| ticket_{CS_A}]$)
- step 11 : ID제공업체는 사용자에게 생성된 OpenID를 발급함으로써 단계를 마무리한다.
  ($E_{PK_{U_a}}[OpenID_{U_a}]$)

4.2.3 사용자 인증 과정

다음의 [Fig. 5]는 클라우드 서버에 접속하여 사용자 인증을 받는 과정이다. 클라우드 서버는 사용자 접속 시 ID제공업체로부터 해당 사용자의 사전에 발급받은 OpenID를 전송받아 OpenID에 첨부된 서비스 허가 티켓을 비교하여 사용자에게 서비스 제공여부를 결정하게 되는 과정을 보여준다.



[Fig. 5] User authentication phase

- step 1 : 사용자는 사전에 등록과정을 수행하여 티켓을 발급받은 클라우드 서비스 중 사용을 원하는 클라우드 서비스에 접속한다.
  ($E_{PK_{CS_i}} = [ID_{U_a}, PW_{U_a}]$)
- step 2 : 서비스 요청을 받은 클라우드 서버는 ID제공업체에 해당 사용자의 OpenID를 요구한다.
  ($E_{PK_{IDP}} = [SAV_A, ID_{U_a}]$)
- step 3 : ID제공업체는 클라우드 서버로부터 전송받은 사용자 정보를 자사의 데이터베이스에 저장된 사용자 정보와 비교한다.
  ($IDP[DBdata] \stackrel{?}{=} CS_A[SAV_A, ID_{U_a}]$)
- step 4 : ID제공업체는 해당 클라우드 서버에 등록된 사용자가 아닐 경우에 요청거부메시지를 전송하고, 등록되어있는 사용자일 경우에는 해당 OpenID를 전송한다.
  ($E_{PK_{CS_i}}[OpenID_{U_a}]$)
- step 5 : 클라우드 서버는 전송받은 해당 사용자의 OpenID에 자사의 클라우드 티켓 정보가 포함되어 있는지 검증한다.

- step 6 : 클라우드 서버는 전송받은 OpenID에 자사의 티켓 정보가 포함되어 있다면 서비스 응답을 하여 사용자 인증을 마무리한다.

5. 안전성 분석

본 장에서 제안 방식의 프로토콜의 안전성을 앞선 3장에서 제시한 보안 요구사항에 따라 분석한다.

5.1 기밀성

클라우드 서버의 티켓 정보($ticket_{CS_A} = [N_{CS_A} \oplus g^{PW_{U_a}} \bmod p]$)는 클라우드 서버가 생성하는 난수값과 사용자의 패스워드 정보를 지수승한 모듈러 값의 연산을 통해 생성된다. 해당되는 지수의 값은 이산대수 계산의 어려움에 기반을 두어 그 값을 알기 어렵기 때문에 해당 클라우드 서버의 티켓 정보를 탈취하더라도 해당 정보(N_{CS_A}, PW_{U_a})를 분석하는 것이 어렵다. 따라서 인증정보에 대한 기밀성을 보장한다.

5.2 무결성

클라우드 서버의 서비스 인증값($SAV_A = H(N_{CS_A} \oplus SPW_A)$)은 해당 클라우드 서버가 OpenID 서비스에 등록되어 있는지 유무를 판단하는 정보로 사용되기 때문에 무결성이 보장되어야 한다. 서비스 인증값은 해쉬함수 연산을 통해 생성되므로 생성된 서비스 인증값은 임의로 변경이 불가능하다. 만약 악의적인 공격자가 서비스 인증값을 획득하더라도 N_{CS_A}, SPW_A) 분석하는 것이 어렵다. 또한, 등록과정에서 ID제공업체의 데이터베이스와의 비교를 통해 등록이 가능한 클라우드 서버인지 여부를 판별하기 때문에 무결성을 제공한다.

5.3 사용자 인증

클라우드 서버의 티켓 정보($ticket_{CS_A} = [N_{CS_A} \oplus g^{PW_{U_a}} \bmod p]$)는 사용자의 OpenID 생성에 대한 정보를 가지고 있다. 악의적인 공격자는 해당 티켓 정보를 가로채더라도 서버의 난수값(N_{CS_A})과 사용자의 패스워드 정보(PW_{U_a})를 획득하기 어렵기 때문에 해당 클라우드 서비스의 올바른 티켓 정보를 발급받을 수 없다. 그렇기 때문에 클라우드 서비스를 사용하기 위한 OpenID 또한 발급받을 수 없기 때문에 정상적인 사용자임을 증명할 수 없다. 따라서 안전한 사용자 인증 기능을 제공한다.

5.4 접근제어

클라우드 서버의 티켓 정보($ticket_{CS_A} = [N_{CS_A} \oplus g^{PW_{U_a}} \bmod p]$)는 서버의 난수값과 사용자의 패스워드 정보가 필요하기 때문에 공격자가 발급받는 것 자체가 어렵다. 따라서 공격자가 클라우드 서비스에 접근할 수 없으며, 정상적인 서비스를 제공받을 수가 없다. 또한 난수값과 패스워드 정보를 획득하더라도 서비스 인증값($SAV_A = H(N_{CS_A} \oplus SPW_A)$)을 생성할 수 없기 때문에 클라우드 서버는 ID제공업체로부터 사용자 정보를 제공받을 수가 없다. 따라서 서비스 자체가 불가능하기 때문에 안전한 접근제어 기능을 제공한다.

〈Table 1〉 Security Analysis

	Security requirements	Proposed scheme
Confidentiality	Parameters must also be able to verify the legitimacy of that user.	$ticket_{CS_A}$ is created by discrete logarithm function. Because it is difficult to know the value of the N_{CS_A}, PW_{U_a}, even if the cloud ticket information is captured, the cloud cannot be used.
Integrity	Cloud service authentication value should not be data forgery.	It is difficult to know the value of the N_{CS_A}, SPW_A, even if the $ticket_{CS_A}$ information is captured, the cloud cannot be used. Therefore, our proposed scheme provides integrity
User authentication	There needs to be unified into authentication procedure different cloud service each.	Can only legitimate users to issue a $ticket_{CS}$. Multi-authentication using this cloud ticket is possible.
Access control	Access control should interrupt illegal access before it occurs.	It is possible for $ticket_{CS_A}, SPW_A$, to prevent an attacker to access thoughtlessly the cloud service.

6. 결론

다양한 클라우드 서비스가 등장함에 따라 하나의 클라우드 서비스만 사용하는 것이 아닌 여러 개의 클라우드 서비스를 함께 사용하는 멀티 클라우드 환경이 주목을 받고 있다. 하지만 각각 서로 다른 클라우드 서비스를 사용하려면 각각의 서비스에 개별적인 사용자 인증과정을 수행하기 때문에 절차가 번거로울 뿐만 아니라 거듭된 인증과정 수행으로 인한 피싱공격, 하이재킹 등의 보안 문제점이 발생할 가능성이 있다. 따라서 본 논문에서는 사용자가 각각의 다양한 클라우드 서비스를 사용하는 멀티 클라우드 환경에 접근하고자 할 때 사용자의 자격증명을 신뢰할 수 있는 ID제공업체에 의해 제공받아 안전하고 편리하게 사용자 인증 기법을 제안하였다.

이를 통해 멀티 클라우드 환경을 구축하여 각각의 클라우드 서비스에 사용자 인증 과정을 수행할 때 정당한 사용자에게 클라우드 서버 고유의 티켓 정보를 발급하여 OpenID에 추가 하고, 추가적인 클라우드 서비스에 대해서도 티켓 정보만을 발급 받아 해당 사용자의 고유의 OpenID에 추가하여 단일인증을 수행한다. 따라서 인증되지 않은 사용자에 대해서는 서비스 이용이 원천적으로 차단되며, 이에 따른 중요한 데이터의 노출에 대한 차단 또한 가능할 것으로 기대할 수 있다. 또한 OpenID를 사용하여 편리하게 제 2, 3의 클라우드 서비스에 사용자 인증이 가능하다. 따라서 다양한 클라우드 서비스가 제공되고 있는 멀티 클라우드 환경에 대한 보안성과 편의성을 향상시킬 수 있을 것으로 기대된다.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2012-010886)

REFERENCES

[1] Korea Internet & Security Agency. Research on the Environment Analysis of Cloud Services and Policy Direction, Nov. 2010.

[2] A. Bessani, M. Correia, B. Quaresma, F. Andre,´ and P. Sousa. DEPSKY: Dependable and Secure Storage in a Cloud-of-Clouds. Proceedings of ACM EuroSys, pp. 31-46, 2011.

[3] Rasib Hassan Khan, Jukka Ylitalot, Abu Shohel Ahmed. OpenID Authentication As A Service in OpenStack. Information Assurance and Security (IAS), 2011 7th International Conference on, pp. 372-377, Dec. 2011.

[4] S. J. Park, H. Y. Kim. Design and Implementation of a Secure Data Storage System for Corporations using Multi-clouds. Journal of Korean Institute of Information Technology, Vol 11. No 3. pp. 151-157, 2013.

[5] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon. RACS: A Case for Cloud Storage Diversity. Proceedings of ACM SoCC'10, pp. 229-240, 2010.

[6] David Recordon, Drummond Reed. OpenID 2.0: A Platform for User-Centric Identity Management. DIM '06 Proceedings of the second ACM workshop on Digital identity management, pp. 11-16, 2006.

[7] J. H. You, C. K. Park.. Design and Implement of User Authentication using I-PIN in OpenID Service. Proceedings of the KAIS Fall Conference, 04. pp. 949-952, 2009.

[8] David Nu˜nez, Isaac Agudo, Javier Lopez. Integrating OpenID with proxy re-encryption to enhance privacy in cloud-based identity services. Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp.241-248, Dec. 2012.

[9] Zhifeng Xiao, Yang Xiao. Security and Privacy in Cloud Computing. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 2, SECOND QUARTER 2013, pp. 843-859, May. 2013.

[10] Kevin D. Bowers, Ari Juels, Alina Oprea. HAIL: a high-availability and integrity layer for cloud storage. CCS '09 Proceedings of the 16th ACM conference on Computer and communications security, pp. 187-198, 2009.

[11] C. Cachin, R. Haas, and M. Vukolic. Dependable storage in the Intercloud. Research Report RZ 3783, IBM Research, Aug. 2010.
[12] M. AlZain, E. Pardede, B. Soh, and J. Thom. Cloud computing security: from single to multi-clouds. the 45th Hawaii International Conference on System Sciences, pp. 5490-5499, Jan. 2012.
[13] Y. S. Cho, S. H. Jin. Overview and Comparison of Internet Identity Management System. Electronics and Telecommunications Trends, Vol. 22, No. 3, pp. 136-143, Jun. 2007.
[14] Y. S. Jeong, S. H. Lee. User Authentication Protocol through Distributed Process for Cloud Environment. Journal of the Korea Institute of Information Security and Cryptology, Vol. 22, No. 4, pp. 841-849, Aug. 2012.
[15] Mohammed A. AlZain, Ben Soh, Eric Pardede. MCDB: Using Multi-Clouds to Ensure Security in Cloud Computing. Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on, pp. 784-791, Dec. 2011.

위 유 경(Wi, Yukyeong)

・2012년 2월 : 순천향대학교 정보보호학과(공학사)
・2012년 2월 ~ 현재 : 순천향대학교 정보보호학과 석사과정
・관심분야 : 정보보호제품평가, 스마트워크 보안, 제어시스템 보안, 콘텐츠 보안
・E-Mail : ykwi@sch.ac.kr

곽 진(Kwak, Jin)

・2000년 8월 : 성균관대학교 생물기전공학과(공학사)
・2003년 2월 : 성균관대학교 컴퓨터공학과(공학석사)
・2006년 2월 : 성균관대학교 컴퓨터공학과(공학박사)
・2007년 3월 ~ 현재 : 순천향대학교 정보보호학과 교수
・관심분야 : 자동차 보안, 암호프로토콜, 응용시스템보안, 개인정보보호, 정보보호제품평가, 클라우드 컴퓨팅 보안
・E-Mail : jkwak@sch.ac.kr