Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.2, Apr. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.2.261

클라우드 환경에서 공모 저항을 지원하는 이중 키 기반의 사용자 인증 모델

최 정 희,† 이 상 호‡
충북대학교

A User Authentication Model Based on Double Key for Secure Collusion Resistance in the Cloud Environment

Jeong-hee Choi,† Sang-ho Lee‡
Chungbuk National University

요 약

최근 IT 기술이 발전하면서 휴대폰, 테블릿 등 다양한 이동 장치를 사용하는 사용자가 증가하면서 클라우드 서비스에 대한 관심이 증가하고 있다. 그러나, 사용자의 서비스 요구가 증가하면서 데이터에 접근하는 다양한 방법을 제어하거나 통제하는 기술들이 요구되고 있다. 본 논문에서는 클라우드 환경에서 제공하는 다양한 서비스에 접근하는 사용자의 접근 권한을 2개의 키(비밀키와 접근제어키)를 이용하여 사용자의 인증 효율성을 향상시킨 2중 키 기반 사용자 인증 모델을 제안한다. 제안 모델에서는 2개의 키를 이중으로 사용자의 접근 권한을 제어하기 위한 알고리즘(키 생성, 사용자 인증, 권한 등급 허용 등)을 시퀀스 다이어그램을 통해 동작과정 및 기능을 세분화하고 있다. 또한, 제안 모델에서는 2개의 키를 사용자 인증과 서비스 권한 등급에 사용하여 클라우드 서비스에서 문제되고 있는 다양한 보안 문제들을 해결하고 있다. 특히, 제안 모델에서는 사용자의 접근제어를 담당하는 알고리즘에서 권한에 따른 사용자의 서비스 등급을 결정하게 함으로써 클라우드 관리자가 사용자의 서비스 접근 허용 정보를 관리할 수 있도록 관리 프로세스를 단축시킨 것이 가장 큰 특징 중에 하나이다.

ABSTRACT

Recently, with the development of IT technology, there is an increasing interest in cloud services as the number of users using mobile devices such as mobile phones and tablets is increasing. However, there is a need for techniques to control or control various methods of accessing data as the user's service demands increase. In this paper, we propose a dual key based user authentication model that improves the user 's authentication efficiency by using two keys (secret key and access control key) to access the users accessing various services provided in the cloud environment. In the proposed model, the operation process and the function are divided through the sequence diagram of the algorithms (key generation, user authentication, permission class permission, etc.) for controlling the access right of the user with dual keys. In the proposed model, two keys are used for user authentication and service authorization class to solve various security problems in the cloud service. In particular, the proposed model is one of the most important features in that the algorithm responsible for access control of the user determines the service class of the user according to the authority, thereby shortening the management process so that the cloud administrator can manage the service access permission information of the user.

Keywords: Intra-Cloud, Security, Authentication, Single-Authority, Multi-Authority

Received(08. 14. 2018), Modified(11. 13, 2018), Accepted(01. 28. 2019)

† 주저자, heebest75@gmail.com
‡ 교신저자, shlee@chungbuk.ac.kr(Corresponding author)

I. 서 론

최근 사용자들이 사용하는 통신장비들이 유선에서 무선으로 변화하면서 한 장소에서 서비스를 이용하기 보다는 여러 장소를 이동하면서 클라우드 서비스를 이용하는 사용자 수가 몇 년 사이 급속하게 증가하고 있다.

클라우드 환경에서는 서비스 범위에 따라 Intra 클라우드와 Inter 클라우드로 구분되며 서비스를 이용하는 사용자의 권한에 따라 인증의 허용범위가 다른 인증모델이 적용이 된다. 서로 다른 인증 모델에서 서비스 사용을 위해 사용자 스스로가 사용자 인증을 제어, 설정, 변경하기는 어려운 이유로는 일반사용자가 사용하기에는 설치 절차가 복잡하고[4], 기본 인프라의 안전한 배치와 서비스의 안전한 사용을 보장하는 정교하고 완벽한 보안 솔루션이 없기 때문이다[5]. 따라서 클라우드 환경에서 사용자의 권한에 따른 인증의 복잡함은 여전히 존재한다. 따라서 사용자 인증을 위한 인증의 효율성과 인증과정에서 발생하는 위험성을 제거하는 정교한 개인정보보호와 데이터 보호의 효율적이고 안정적인 새로운 인증모델이 필요하다.

그러나, 최근 사용자에게 제공되고 있는 클라우드 서비스는 다양한 기업에서 다양한 인증 방법으로 사용자에게 제공되고 있어 사용자의 인증 효율성을 향상시키기 위한 관리 인증 모델이 필요하다. 사용자의 클라우드 서비스 인증 효율을 향상시키기 위해서 최근 연구에서는 다중 속성을 이용한 기법과 암호화 기법을 이용한 기법으로 나뉘어 연구가 진행되고 있다. 다중 속성을 이용한 기법에서는 사용자의 서명과 암호화 체계를 결합하여 사용자의 접근을 제어하는 연구가 대부분이며, 암호화 기법을 이용한 기법에서는 사용자의 키 생성과 복호화의 효율성을 개선시키는 내용이 중심이되는 연구가 진행되고 있다.

본 논문에서는 클라우드 환경에서 사용자의 인증 효율성을 개선시키기 위해서 인증서버와 클라우드 관리자 사이에서 사용자의 접근 권한을 제공하기 위해서 비밀키와 접근제어키를 이용한 2중 키 기반의 사용자 인증 모델을 제안한다. 제안 모델은 2중 키를 인증서버와 클라우드 관리자 사이에서 사용자의 키 생성, 사용자의 인증 권한, 권한 등급 허용 알고리즘 등을 각각 처리하여 사용자의 접근 권한의 효율성을 향상시키고 있다. 제안 모델은 각 알고리즘의 동작과정을 객관적으로 명시화하기 위해서 시퀀스 다이어그램을 통해 동작과정과 기능을 세분화하고 있다. 제안 모델은 기존 모델에서 제공되지 않은 프라이버시 예방과 익명 인증을 중심으로 사용자의 접근 인증 방법을 2중키를 이용하여 제공하고 있는 것이 기존 연구와의 가장 큰 차이점 중 하나이다. 특히, 제안 모델은 클라우드 서비스에서 사용자에게 제공하려고 하는 서비스의 등급과 사용자의 접근 권한에 따라 서비스를 유기적으로 제공함으로써 인증서버와 클라우드 관리자의 오버헤드를 낮추고 있다. 특히, 제안 모델에서는 사용자의 접근제어를 담당하는 알고리즘에서 권한에 따른 사용자의 서비스 등급을 결정하게 함으로써 클라우드 관리자가 사용자의 서비스 접근 허용 정보를 관리할 수 있도록 관리 프로세스를 단축시키고 있다.

이 논문의 구성은 다음과 같다. 2장에서는 클라우드 환경에서 인증 모델들의 다중 속성 및 단일 속성에 따른 기존 인증 모델들에 대한 연구들을 분석한다. 3장에서는 클라우드 보안 위협에 안전하고 효율적인 사용자 권한 기반의 인증모델을 제안하고, 제안한 인증 모델의 과정에 따른 시퀀스다이어그램의 기술과 함께 각 단계에서 이루어지는 사용자 인증 알고리즘을 상세히 설명한다. 4장에서는 제안 모델의 보안에서의 기능과 안전성을 기존의 기법들과 비교분석하고 마지막으로 결론을 맺는다.

II. 관련연구

2.1 클라우드 서비스

클라우드 컴퓨팅은 구성 가능한 컴퓨팅 자원(네트워크, 서버, 스토리지, 응용 프로그램 및 서비스 등)의 공유 풀에 편재하고 편리한 On-Demand 네트워크 액세스를 가능하게 하는 모델로 최소한의 관리 노력으로 신속하게 공급하고 배포 할 수 있다[1].

클라우드 모델은 다섯 가지 중요 특성으로 On-Demand Self-Service, Broad Network Access, Resource Pooling, Rapid Elasticity, Measured Service과 세 가지 서비스 모델 SaaS, PaaS, IaaS로 구성되며, 네 가지 배치 모델 Private Cloud, Community, Public Cloud, Hybrid Cloud로 구성된다[2].

클라우드 환경에서의 인증은 크게 세 가지로 볼 수 있다. 첫째, 호스트 인증은 클라우드 호스팅 구성요소가 기본 클라우드 인프라 스트럭처에서 제공하는

서비스와 통신 할 때 이루어지는 인증으로, 운영 체제와 프로세스 간의 인증의 경우와 같은 인증 메커니즘에 의해 활용 될 수 있다. 둘째, Intra 클라우드 인증은 구성 요소 두 개의 클라우드 호스트 구성 요소가 동일한 클라우드 플랫폼에서 통신 및 실행되는 인증이며, 마지막으로 Inter 클라우드 인증은 구성 요소가 다른 클라우드 플랫폼 및 개별 관리 도메인에서 실행될 때 클라우드 간 인증이다[3].

클라우드 환경에서는 서비스를 이용하는 사용자의 권한에 따라 인증의 허용범위가 다른 인증모델이 적용이 된다. 서로 다른 인증 모델에서 서비스 사용을 위해 사용자 스스로가 사용자 인증을 제어, 설정, 변경하기는 어려운 이유로는 일반사용자가 사용하기에는 설치 절차가 복잡하고[4], 기본 인프라의 안전한 배치와 서비스의 안전한 사용을 보장하는 정교하고 완벽한 보안 솔루션이 없기 때문이다[5]. 따라서 클라우드 환경에서 사용자의 권한에 따른 인증의 복잡함은 여전히 존재한다. 따라서 사용자 인증을 위한 인증의 효율성과 인증과정에서 발생하는 위험성을 제거하는 정교한 개인정보보호와 데이터 보호의 효율적이고 안정적인 새로운 인증모델이 필요하다. 본 논문에서는 사용자의 개인정보보호와 저장되어진 데이터의 보호를 위한 권한에 따라 인증 범위가 달라지는 권한 기반의 사용자 인증 모델을 제안한다.

2.2 기존연구

최근 많은 논문에서 클라우드 환경에서 보다 안전한 사용자 인증을 위한 방법으로 다중 속성을 이용한 키 생성 방식의 클라우드 모델이 연구되고 있다. 또한 키 생성과 암호화 시에 보다 효율적인 연산이 이루어질 수 있는 클라우드 모델 역시 연구되고 있다.

Chen et al.에서는 서명과 암호화 체계의 결합 보안에 중점을 두고 결합 보안 설정에서 대표적으로 CP-ABSC(Ciphertext Policy Attribute - based Signcryption) 체계를 제시했다[6]. 그러나 서명 확인을 위해 일반 텍스트 메시지가 필요하기 때문에 공용 검증을 지원할 수 없다.

Liu et al.에서는 CP-ABE와 ABS 기반의 PHR 시스템에 대해 보안 공유 체계를 제안하였다[7]. 하지만 이 방식은 임의의 오라클 모델에서 안정적인 보안 보장된다.

Sreenivasa 에서는 공용검증과 함께 접근 제어 기반으로 하는 CP-ABSC를 제안하였다[8]. 보안이 표준 모델에서 현실적으로 이루어질 수 있지만, 그 기법은 개인 정보 보호 속성을 고려하지 않았다.

Yu et al.에서는 키 정책 서명과 암호문 정책 암호화를 지원하는 하이브리드 접근정책 ABSC를 제안했다[9]. 암호문의 크기는 일정하고. 표준모델에서 안전하다는 것이 입증되었다. 하지만, 암호화 단계에서 오직 임계 엑세스 구조에만 지원하고, ABSC 체계상에서 다중 권한 시스템을 지원할 수 없고 오직 단일 권한만을 갖는다.

Han et al. 에서는 분산 환경에서 기밀성 및 세분화 된 제어를 제공하기 위해 다중 권한 체계를 구축했다[10]. 이 체계는 중앙 권한을 필요로 하지 않고 사용자의 개인 정보 보호가 가능하지만 공모에 대한 내성이 없으며 사용자 인증을 제공할 수 가 없다.

Lewko et al.에서는 분산 된 완전 보안 MA-ABE 체계를 구축했다[11]. 공모 공격에 저항하기 위해 다른 권한에 의해 발행 된 사용자의 비밀키는 모두 자신의 포괄적인 식별자로 묶여있다. 그러나 보안은 임의 오라클 모델에서 입증되었고, 복합 주문 그룹 요소의 더 긴 크기로 인해 비효율적이다.

Ruj et al. 에서는 다중속성 기관이 사용자에게 비밀키를 발급하는 분산형 ABE체계를 제안하였다[12]. 이 기법에서는 또한 인증을 제공하기 위해 ABS 기법을 채택하여 사용하고 있다.

Yang et al. 에서는 다중 권한 클라우드 시스템에 대한 데이터 액세스 제어 체계를 제안했다[13]. 이 기법에서는 해독 알고리즘에서 가장 비용이 많이 드는 작업은 클라우드에 아웃소싱으로 처리하고 데이터 사용자는 평문을 복구하기 위해 하나의 지수 연산만 수행한다. 그러나 이 기법에서 클라우드 서버는 사용자의 부분 암호 해독을 실행하기 위해 사용자의 속성을 알아야하기 때문에 속성 프라이버시를 보장할 수 없다.

Li et al. 에서는 효율성 향상을 위해 ABE 기법을 공식화 하고 키 생성과 복호화의 여러 구조를 제안하였고[14], 이 기법에서 제시하는 복호화의 정확성을 검증하기 위해서, Lai et al.에서는 클라우드 서버에 의해 계산 된 복호화 결과를 사용자가 효율적으로 확인할 수 있는 검증 가능한 메커니즘을 제안했다[15]. [14,15] 두 논문 모두 ABE 방법의 효율성 향상에 관한 검증방법을 제시했지만, 정작에 사용자의 보안에 대한 부분은 논의하지 않았다. 따라서 ABE기법의 효율성은 검증이 되었지만 보안 측면에서의 안전성 개선이나 검증은 없어 여전히 ABE기법

의 개인정보 보안 문제가 존재한다.

Zhang et al. 기법에서는 암호문에 접근 정책을 숨기고 복호화 전에 시험 연산을 수행하여 비밀 키가 숨겨진 암호 속성과 일치하는지 검사하는 익명 ABE 기법을 제안했다[16]. 그러나 이 방법은 AND 게이트 액세스 정책만 지원하고 임의의 Oracle 모델의 보안만을 기술하였다. 또한, 키 생성 단계에서 특정 개인정보의 안전성을 보장할 수 없다.

Q. Huang et al. 에서는 데이터 접근 제어를 위한 계산을 fog 컴퓨팅 시스템에서 하는 ABE 기법을 제안하였다[17]. 사용자측에서 암호화 또는 복호화를 하면 많은 계산 오버 헤드가 발생하지만, fog node들에서 암·복호 계산을 진행하게 되면, 최종 사용자 자원 제약 장치의 효율성은 개선되지만, 사용자의 접근제어키의 암·복호화를 계산을 fog node에서 함으로 인한 사용자 정보 노출의 가능성이 높아 보안 위협이 커진다.

J. Wang et al. 기법은 세분화 된 액세스 제어, 전체 권한 위임 및 확장 가능한 속성 해지를 제공하기 위해 계층 적 ABE를 제안했다[18]. 또한 제안 방법은 프록시의 재 암호화 (PRE) 및 지연 재 암호화 (LRE)를 HABE 체계에 적용하였다. 이 기법은 속도와 계산에 효율성은 좋으나, 프록시를 이용하여 재 암호화를 하면 사용자의 개인정보에 취약하며, 액세스 제어에 필요한 계산을 권한 위임하게 되면 키 위탁문제에 노출이 된다.

III. 클라우드를 위한 권한 기반 속성 키 인증 모델 설계

이 절에서는 클라우드 환경에서 클라우드 사용자가 인증서버로부터 부여받은 권한에 따라 안전하게 서비스를 제공 받을 수 있는 권한 기반의 인증 모델을 제안한다.

3.1 개요

최근 클라우드 서비스가 대중화되면서 많은 사용자들이 개인 PC를 사용하지 않고 언제 어디서나 간편하게 사용할 수 있는 클라우드 서비스를 사용하고 있다. 그러나 클라우드 서비스는 다양한 기업에서 다양한 인증 방법이 제공되고 있어 사용자의 인증 방식이 여러 형태로 요구되고 있어 사용자 인증을 보다 효율적으로 관리할 수 있는 모델이 필요하다. 제안 인증 모델에서는 기존 클라우드 서비스의 인증 모델 방식의 문제점을 분석하고 분석된 문제점을 기반으로 사용자의 인증을 효율적으로 관리할 수 있는 권한 기반의 인증 모델을 제안한다. 제안된 권한 기반 인증 모델은 사용자와 인증서버 사이에서 인증을 수행할 때, 인증서버와 클라우드 관리자간 서로 공모는 할 수 없다고 가정한다.

제안 인증 모델은 클라우드 환경에서 사용자 권한에 따라 클라우드 서비스 접근 허용 등급이 달라진다. 사용자에는 일반 사용자와 그룹사용자로 구분되며, 인증서버는 사용자의 정보를 이용하여 일반사용자는 *PID*, 그룹사용자는 *GID*로 사용자 등록을 한다. 이 때, 인증서버는 일반사용자와 그룹사용자의 권한을 확인 한 후, 비밀키와 접근제어키를 생성한다. 비밀키는 사용자의 서비스 인증을 위해 사용되며, 접근제어키는 사용자의 권한에 따른 접근제어 허가 등급을 확인하기 위해 사용된다. 비밀키와 접근제어키는 키 쌍으로 이루어져있으며, 사용자와 인증서버 그리고 클라우드 관리자가 필요한 정보만을 저장·관리한다. Fig.1은 클라우드 환경에서 사용자의 권한에 따라 서비스 접근 제어 허가 등급이 달라지는 권한기반 인증모델을 보여준다.

제안 모델의 구성요소는 클라우드 서비스를 이용하는 사용자(Personal User or Group User : User), 인증서버(Authentication Server : AS), 클라우드 관리자(Cloud Manager : CM) 그리고 클라우드 서비스 공급자(Cloud Service Provider : CSP)로 구성된다. 사용자는 사용자 에이전트가 관리하는 그룹사용자와 일반사용자로 나뉘며, 사용자는 클라우드 공급자가 제공하는 서비스를 이용하기 위해 클라우드 관리자에 서비스 요청을 한

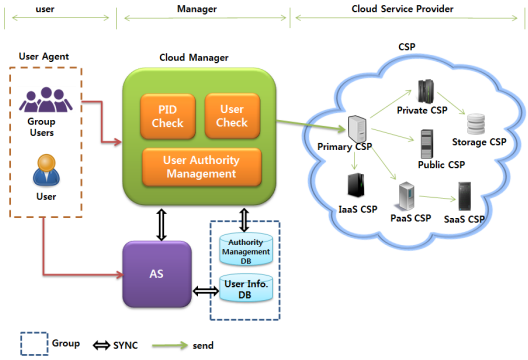

Fig. 1. An authority-based authentication model in Cloud

다. 사용자가 받는 서비스의 등급은 클라우드 공급자가 제공하는 public, authentication authority로 구분되며, 접근제어 권한 등급(pi 등급)에 따라 부여된 권한으로 이용한다. 클라우드 관리자는 사용자로부터 서비스 인증 요청이 오면 데이터베이스에 사용자의 등록 여부를 확인하고 사용자등록이 되지 않은 사용자라면 사용자에 사용자 등록을 요청 하고, 데이터베이스에 사용자 등록이 되어 있는 사용자라면 정상적인 인증 요청인지를 확인한다. 정상적인 사용자 인증 확인 후 사용자 등급에 따라 해당 클라우드 서비스 공급자의 API를 사용자에게 리다이렉트한다. 인증서버는 일반사용자와 그룹사용자를 구분하여 사용자 등록을 하고, PID를 생성한다. 또한, 사용자 등록과 동시에 사용자의 권한에 따른 접근제어키 k_{ac}와 인증에 필요한 비밀키 k_{se}를 생성하여 사용자 관리 테이블에 저장한다. 이때, 사용자 정보 저장 데이터는 인증서버와 클라우드 관리자가 공유하게 된다. 클라우드 관리자로부터 사용자의 인증 요청이 오면 인증서버는 사용자의 인증요청이 정당한지를 확인하여 그 결과를 클라우드 관리자에 응답한다. 마지막으로 클라우드 서비스 제공자는 클라우드의 저장장치 서비스, 어플리케이션 서비스, 클라우드 인프라 구축 서비스 등을 사용자에게 제공하는 서비스 제공자이다.

3.2 권한 속성 기반의 인증 모델

이 절에서는 사용자의 권한 기반 인증 모델에 대한 인증모델, 시퀀스 다이어그램 그리고 알고리즘 등으로 구분하여 기술한다.

3.2.1 인증 모델

클라우드 서비스를 이용하고자하는 사용자는 클라우드 접속 앱을 통해 사용자 등록 과정을 사전에 진행한다. 사용자 등록과정에서 SSL/TLS와 같은 안전한 채널을 통하여 사용자와 인증서버 간 공유키 k_1, 인증서버와 클라우드 관리자 간 공유키 k_2 그리고 사용자와 클라우드 관리자 간 공유키 k_3를 공유하고, 사용자는 인증서버로부터 사용자 정보 PID를 발급받는다. 이후 인증키 생성과 인증과정은 사용자 정보 PID를 이용하여 이루어진다.

인증서버에서 생성한 키 전달은 제안 인증모델을

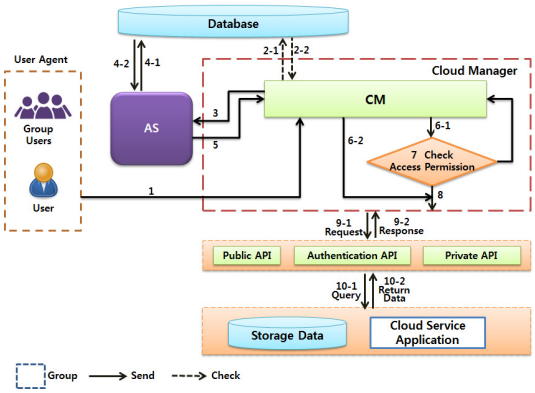

Fig. 2. Authentication Process

구성하는 사용자간 사용자 등록 과정에서 공유된 공유키를 이용한다. 제안 인증모델은 Fig. 2와 같이 10단계로 동작한다.

· Step 1 : 클라우드 서비스 요청

사용자가 클라우드 서비스를 제공받기 위해서 클라우드 관리자에게 서비스를 요청하는 단계이다. 이 단계에서 사용자정보 PID를 비밀키 k_{se}와 1증가시킨 시퀀스 넘버$sn+1$을 공유키로 암호화 하여 클라우드 관리자에게 식(1)과 같이 전달한다. 그러나 등록된 사용자이지만 인증을 위한 키 발급이 이루어지지 않은 사용자는 식(1)'와 같이 사용자 정보 PID를 공유키로 암호화 하여 전달한다.

User→CM :

$$E_{k_3}[PID, E_{k_1}[PID, h(k_{se}, sn+1)]] \quad (1)$$

$$E_{k_3}[PID] \quad (1)'$$

· Step 2 : 사용자 등록 확인

클라우드 관리자와 데이터베이스 간에 사용자의 등록정보가 데이터베이스에 저장되어 있는지 확인하는 단계이다. 식(2)와 같이 클라우드 관리자는 인증요청이 온 사용자 정보가 데이터베이스에 등록되어 있는지 확인한다. 등록된 사용자라면 인증 요청을 시작한다.

$$Compare\ PID \cong PID' \quad (2)$$

· Step 3 : 사용자 인증 확인 요청

클라우드 관리자는 사용자로부터 받은 사용자 정보 PID가 데이터베이스에 있음을 확인하고, 식(3)과 같이 클라우드 관리자와 인증서버간 공유된 공유키를 이용하여 암호화 한 후 인증서버에 사용자 인증 확인을 요청한다. 그러나 사용자 인증키가 없는 경우, 클라우드 관리자는 인증서버로 식(3)'과 같이 사용자 정보 PID를 전달한다.

CM→AS :

$$E_{k_2}[PID, E_{k_1}[PID, H(k_{se}, sn+1)]] \quad (3)$$

$$E_{k_2}[PID] \quad (3)'$$

· Step 4 : 사용자 비밀키와 접근제어키 생성

클라우드 관리자가 전달한 인증 요청 정보를 이용하여 인증서버는 사용자 정보 PID가 등록된 사용자의 인증요청이라 판단되면 저장된 사용자의 시퀀스 넘버를 1증가한 후 식(9)와 같이 확인한다. 그러나, 시퀀스 넘버의 만료 또는 사용자 인증키 발급이 필요한 경우에 다음과 같은 절차로 시퀀스 넘버, 접근제어키, 비밀키 그리고 권한등급을 생성한다. 인증서버는 사용자 정보 PID를 이용하여 식 (4)과 같이 사용자 권한등급 pi를 생성한다. 이때, 권한등급은 권한정보(AI : public 등급, authentication 등급, private 등급 등)의 서비스 제공 유무에 따라 0과 1로 나타내어 0부터 7사이의 등급을 부여한다. 시퀀스 넘버는 인증키 재사용 방지를 위해 식(5)와 같이 생성되며 인증 요청과 함께 1씩 증가하는 값을 갖는다. 또한 시퀀스 넘버가 일정 크기 이상의 값을 갖게 되면 시퀀스 넘버가 만료로 판별되어 새로운 인증 키 발급 과정을 수행한다. 접근제어키 k_{ac}를 생성하는 식(6)의 Z_p는 $\{x \in Q_p : ||x||_p \leq 1\}$와 같으며 산발적인 정수의 집합이다. 이때 난수(r_1, r_2)들을 뽑아 지수승을 선택함으로써 이산대수의 어려움에 기반 한 사용자의 인증 키 발급 과정을 안전하게 수행할 수 있다. x와 a는 사용자의 등록과 사용자 권한에 따른 접근제어 허가 등급을 확인하기 위해서 인증서버가 사용자별로 생성한다. 이 접근제어키 k_{ac}에 따라 비밀키 k_{se}의 해시값이 달라지기 때문에 제 3자로부터 비밀키 k_{se}의 악용을 방지 할 수 있다. 비밀키k_{se} 생성은 권한등급 pi, 접근제어키 k_{ac} 그리고 사용자 정보 PID를 일방향 해시함수 H의 해시값으로 식(7)과 같이 생성한 후, 생성된 비밀키와 접근제어키, 권한정보 그리고 시퀀스 넘버 등을 식 (8)과 같이 데이터베이스에 저장한다. 이때 시퀀스 넘버는 랜덤한 정수 값을 이용한다.

$$pi \in AI (0 \leq k \leq 7) \quad (4)$$

$$Generate \quad sn' \leftarrow sn + 1 \in N^* \quad (5)$$

$$\begin{aligned} &Generate \quad (r_1, r_2) \in Z_p \\ &k_{ac} = (p^{x+ar_1}, p^{r_1}, p^{r_2}) \in Z_p \end{aligned} \quad (6)$$

$$Generate \ k_{se} = H(PID || pi || k_{ac}) \quad (7)$$

$$AS \Leftrightarrow Database : k_{se}, k_{ac}, pi, sn \quad (8)$$

$$Compare \ H(k_{se}, sn+1) \cong H(k_{se}, sn+1)' \quad (9)$$

· Step 5 : 사용자 인증 확인 응답

인증서버는 사용자 인증 후, 사용자의 접근권한 등급을 확인할 수 있도록 사용자 정보와 권한등급을 사용자 접근제어키로 암호화 하여 식(10)과 같이 전달한다. 이때 접근제어키는 사용자 키 생성 후 인증서버가 클라우드 관리자에 전달한 사용자 접근제어키로 인증서버와 클라우드 관리자가 분산 관리하는 접근제어키 정보이다.

$$AS \rightarrow CM : k_{ac}[PID, pi] \quad (10)$$

· Step 6 : 사용자 권한등급 확인

클라우드 관리자는 인증서버로부터 전송 받은 pi와 사용자 정보 PID를 이용하여 식 (11)과 같이 데이터베이스에 저장·관리되고 있는 사용자의 서비스 권한 등급(pi)을 확인한다. 확인된 권한 등급(pi)에 따라 public API 서비스와 check permission API로 나누어 처리한다. 이때, 데이터베이스의 사용자 정보는 인증서버와 클라우드 관리자가 동기화되어 관리된다.

$$CM \Leftrightarrow Database : pi \quad (11)$$

· Step 7 : 사용자의 권한 등급 정보에 따른 서비스 분기

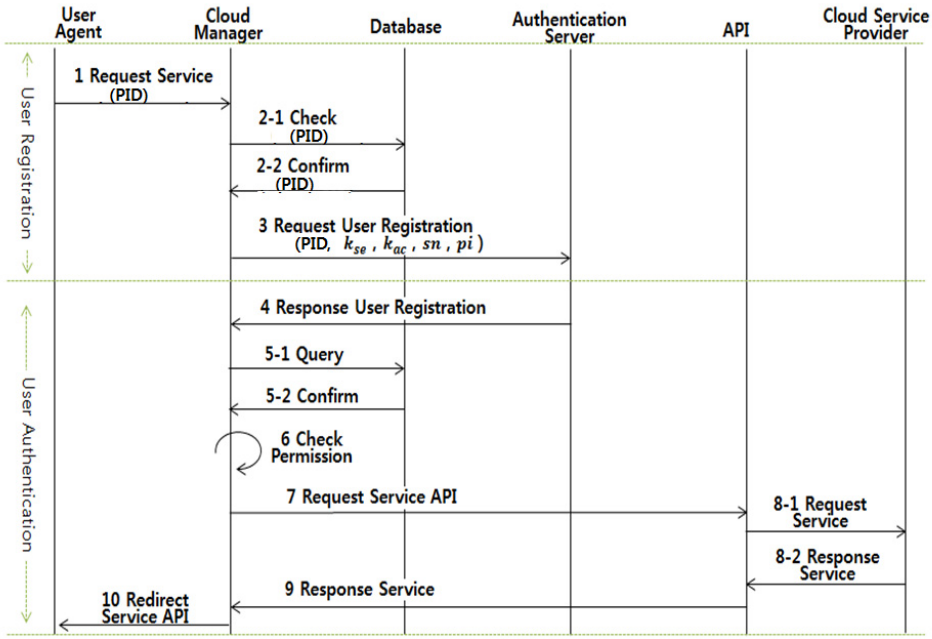

Fig. 3. User Registration Sequence Diagram

클라우드 관리자는 사용자의 권한 등급(pi)에 따라 권한 정보(AI)를 확인하여 사용자에 서비스할 인터페이스가 authentication API와 private API로 구분하여 처리한다.

· Step 8 : 사용자 권한 등급에 따른 서비스 API

이 단계에서는 사용자의 권한등급에 따라 구분되어져 사용자 권한 등급에 해당되는 API로 사용자의 서비스 요청이 분기되어 전송된다.

· Step 9 : CSP API 요청 및 응답

이 단계에서는 클라우드 관리자(CM)가 서비스 사용자가 이용하고자 하는 서비스 API에 대해 식 (12)과 같이 클라우드 서비스 공급자(CSP)에 요청하고, 클라우드 서비스 공급자(CSP)는 식 (13)과 같이 이에 응답한다.

$$\mathrm{CM} \rightarrow \mathrm{CSP\ API} \begin{bmatrix} public \\ authentication \\ private \end{bmatrix} \tag{12}$$

$$\mathrm{CM} \leftarrow \mathrm{CSP\ API} \begin{bmatrix} public \\ authentication \\ private \end{bmatrix} \tag{13}$$

· Step 10 : 사용자 서비스 요청 응답

이 단계에서 클라우드 관리자는 클라우드 사용자의 클라우드 서비스 요청에 따른 클라우드 서비스 공급자의 서비스 API를 사용자에 연결하여 서비스 응답한다.

3.2.2 시퀀스 다이어그램

이 절에서는 클라우드 환경에서 좀 더 안전하고 효율성 있게 처리할 수 있는 권한 기반 사용자 인증 모델을 Fig. 3처럼 시퀀스 다이어그램으로 나타낸다. 이 시퀀스 다이어그램은 사용자 정보 등록 과정과 사용자 인증과정으로 구분한다.

사용자 등록과정에서는 등록 시작과 동시에 사용자가 사용자 정보를 인증서버에 전송하여 사용자 등록을 한다. 사용자 등록이 완료되면, 사용자는 자신의 사용자 등록 정보 PID를 이용하여 1. Request Service와 같이 서비스 인증요청을 한다. 사용자로부터 서비스 인증 요청을 받은 클라우드 관리자는 사용자 등록을 2-1. Check과정에서 데이터베이스에 사용자 등록을 확인 요청하고, 2-2. Confirm과정에서 확인 응답을 받는다. 사용자 등록 확인 후, 클라우드 관리자는 사용자의 인증요청을 인증서버에 3. Request user Registration단계와 같이 요청한

다. 클라우드 관리자로부터 사용자 인증 요청을 받은 인증서버에서는 사용자의 비밀키 k_{se}를 데이터베이로 확인하고, 비밀키 k_{se}와 접근제어키k_{ac}를 이용하여 사용자의 인증 확인을 한다. 만약, 사용자의 비밀키 k_{se}와 접근제어키k_{ac}가 생성되지 않았거나, 시퀀스 넘버 sn의 만료로 인하여 새로이 생성해야 한다면, 인증서버에서는 사용자의 비밀키 k_{se}와 접근제어키 k_{ac}를 생성 후 저장·관리한다. 인증서버에서 사용자 인증의 확인 절차가 끝나면, 인증서버는 클라우드 관리자에 4. Response User Registration과 같이 인증 확인 응답을 전송한다. 인증확인 응답을 받은 클라우드 관리자는 사용자의 권한 등급 정보를 데이터베이스에서 5-1. Query로 확인 요청하고 5-2.와 같이 확인응답을 받는다. 데이터베이스로부터 받은 사용자의 권한 등급에 따라 클라우드 관리자는 6. Check Permission을 진행한다. 사용자 권한 등급에 따라 사용자가 접근 가능한 서비스 API를 7. Request API와 같이 요청한다. 사용자 요청 클라우드 서비스 API를 8-1. Request Service 요청과 같이 클라우드 서비스 공급자에 요청하고, 8-2. Response Service와 같이 클라우드 서비스 공급자에 서비스 응답을 받는다. 클라우드 관리자는 9. Response Service와 같이 클라우드 서비스 공급자로부터 받은 응답을 사용자에게 10. Redirect Service API와 같이 클라우드 서비스를 제공한다. 이때부터 사용자는 클라우드 서비스를 정상적으로 이용할 수 있다.

3.2.3 알고리즘

이 절에서는 클라우드 환경에서 사용자 권한 기반 인증 모델에서 사용되는 비밀키와 접근제어키 생성 알고리즘, 서비스 접근 권한 등급 생성 알고리즘, 사용자 인증 확인 절차 알고리즘 등을 기술한다.

Table 1.은 인증서버에서의 키 생성 알고리즘을 나타낸 것이다. 사용자 키 생성 요청과 인증과정에서 사용되는 사용자 정보 PID는 사용자 등록과정에서 생성한 사용자 식별 정보로 사용자 개인정보 노출에 안전한 식별자이다. 인증서버는 클라우드 관리자로부터 전달받은 사용자 정보 PID를 이용하여 사용자 등록정보의 등록 유무를 확인하고, 등록된 사용자의 키를 생성한다. 키 생성은 각 키 생성 알고리즘에서 수행한다. 생성된 키는 데이터베이스에 저장되고, 사

Table 1. Key Generation Algorithm on AS

```
Procedure : keyGeneration()
Input : PID
Output : k_se , k_ac , sn , pi
1.  begin
2.  while(true)
3.    CM requests checking PID to AS
4.    AS checks PID from database
5.    If((PID) ≠ null)
6.      /* AS generate k_se , k_ac , sn */
7.      sn = snGenerate(PID)
8.      k_ac = akGenerate(PID)
9.      pi = piGenerate(PID)
9.      k_se = skGenerate(
                  H(PID||k_ac||pi) )
10.     Save (PID, k_se, sn, k_ac, pi)
                  in Database
11.     send E_k1〔PID, k_se, sn〕 to User
12.     send E_k2〔PID, k_ac, pi〕 to CM
13.      break
14.   else
15.      AS response null to CM
16.  end while
17.  end
```

용자와 클라우드 관리자에 필요한 정보만을 각각 사전에 공유된 공유키로 전달한다. 사용자에게 전달되는 비밀키k_{se}와 인증 요청할 때 마다 1씩 증가 하여 사용되는 시퀀스넘버sn는 재전송 공격에 대한 안전성을 갖기 위해 사용자 측에서 해시값 $h(k_{se}, sn+1)$으로 인증 요청 메시지로 쓰인다. 그리고 클라우드 관리자에 전달되는 접근제어키k_{ac}는 사용자 인증 확인 후 접근권한을 확인하기 위해 인증서버가 사용자의 접근권한을 $k_{ak}(PID,pi)$와 같이 암호화 하여 전달 할 때 쓰인다. 따라서 사용자가 인증 요청할 때 전달하는 $h(k_{se}, sn+1)$의 비밀키는 인증서버와 사용자만이 알고 있는 키이며, 클라우드 관리자가 사용자와 인증서버 간 공유키인 k_1을 알지 못하기 때문에 클라우드 관리자는 인증서버와 공모하여 사용자의 인증을 위한 비밀키를 조작할 수 없다. 또한 사용자가 자신의 접근권한을 변경 및 조작하기 위해 시도할 때, 접근권한 pi를 $k_{ak}(PID,pi)$와 같이 암호화하는 접근제어키 k_{ac}를 알 수 없기 때문에 접근제어키를 이용한 사용자–인증서버 혹은 사용자–클라우드 관리자 간 공모는 이루어지지 않는다.

결국 사용자가 서비스를 받기 위해서 인증과정과 접근권한부여 과정이 모두 정상적으로 수행되야 하는데 인증과정에서 쓰이는 비밀키는 $E_{k_1}[h(k_{se}, sn+1)]$로 사용자와 인증서버 간 공유키와 비밀키 이고, 접근권한을 위해 쓰이는 접근제어키는 $E_{k_{ac}}[PID, pi]$로 인증서버와 클라우드 관리자만이 알고 있는 접근제어키k_{ac}로 암호화 하였기 때문에 사용자는 알 수 가 없다. 따라서 인증과정과 접근권한과정을 모두 거쳐야 서비스를 제공 받을 수 있는 제안 인증모델에서는 공모에 저항력을 가지고 있다.

Table 2.는 시퀀스 넘버 sn 생성 알고리즘이다. 입력받은 시퀀스 넘버가 $null$값 혹은 만기가 된 값이라면 생성하고, 시퀀스 넘버 sn가 정상 범주의 값이 전송 되었다면 1을 증가하여 $sn+1$을 만든다.

Table 3.은 사용자 정보 PID에 따른 접근제어키 k_{ac}의 생성 알고리즘이다. 접근제어키 k_{ac}를 생성하는 식(6)의 Z_p는 $\{x \in Q_p : ||x||_p \leq 1\}$와 같으며 산발적인 정수의 집합이며, 난수$(r_1, r_2)$들을 뽑아 지수승을 선택함으로써 이산대수의 어려움에 기반 한 사용자의 인증 키 발급 과정을 안전하게 수행할 수 있다. 또한 x와 a는 사용자의 등록과 사용자 권한에 따른 접근제어 허가 등급을 확인하기 위해서 인증서버가 사용자별로 생성한다.

Table 4.는 권한등급 pi 생성 알고리즘이다. 권한 등급은 권한 등급 정보 (AI : $public$, $authentication$, $private$)의 서비스 사용 유무에 따라 0, 1의 값을 갖게 되고 그 값에 따라 권한등급이 결정되는 알고리즘이다.

Table 5.는 비밀키 k_{se} 생성 알고리즘으로, 비밀키는 사용자 정보, 접근제어키, 사용자권한등급은 해쉬값이다. 접근제어키 값에 따라 사용자들의 비밀키가 달라지기 때문에 제3자의 추축이 불가능하다.

Table 6.은 사용자인증을 확인하는 알고리즘이

Table 2. Sequence Number(sn) Generation Algorithm

```
Procedure : snGenerate()
Input : PID
Output : sn
1.  begin
2.    If(sn==null)
3.        Generate sn ∈ N*
4.        return sn
5.    else
6.        Generate sn' ← sn+1 ∈ N*
7.        return sn'
8.  end
```

Table 3. Access Control key(k_{ac}) Generation Algorithm

```
Procedure : akGenerate()
Input : PID
Output : k_ac
1.  begin
2.    Generate (r1, r2) ∈ Z_p
3.    Generate k_ac = (p^(x+ar1), p^r1, p^r2) ∈ Z_p
4.  end
```

Table 4. Authority-grade pi Generation Algorithm

```
Procedure : piGenerate()
Input : PID
Output : pi
1.  begin
2.      Generate pi ∈ AI(0 ≤ k ≤ 7)
3.  end
```

Table 5. Secrete key k_{se} Generation Algorithm

```
Procedure : skGenerate()
Input : PID, k_ac , pi
Output : k_se
1.  begin
2.      k_se = H(PID || k_ac || pi)
3.  end
```

Table 6. User Authentication Algorithm on AS

```
Procedure : authentication()
Input : E_k1[PID, h(k_se, sn+1)]
Output : k_ac[PID, pi]
1.  begin
2.  while(true)
3.    AS checks PID ≅ PID'
4.    if(H(k_se, sn+1) ≠ H(k_se, sn+1)')
5.      break
6.    else send E_kac〔PID, pi〕 to CM
7.  end while
8.  end
```

다. 사용자와 인증서버 간 공유키로 암호화 된 인증 요청 메시지를 복호화하여 사용자 인증 확인 후, 인증서버는 접근제어키로 사용자 정보와 접근권한을 암호화 하여 클라우드 관리자로 전달한다.

Table 7.은 사용자 권한등급을 확인하는 알고리즘이다. 접근제어키로 암호화 된 접근권한을 복호화하여 사용자의 권한 등급에 따라 클라우드 서비스 공급자가 제공하는 해당 클라우드 서비스 API를 사용자에 리다이렉트한다.

Table 7. User Authority-grade_confirm Algorithm on CM

```
Procedure : authority_grade_confirm
Input : PID , k_ac{PID, pi}
Output : PID, AI
1.  begin
2.  while(true)
3.    CM checks PID ≅ PID'
5.    CM checks pi
6.    get AI
7.    return AI
8.  end while
9.  end
```

IV. 보안 평가

이 절에서는 클라우드 환경에서 사용자의 접근 권한에서 요구되는 다양한 요구사항(공모 저항, 적용 표준모델 유무, 개인정보보호, 암호화 속성, 연산 아웃소싱, 익명 인증, 다중 속성, 암호서명 개인정보보호, 공개 검증 가능성 등)에 대해서 Table 8.처럼 기존 기법(MACP-ABE기반 기법[10,11,12,13]과 ABSC 기법[6,7,8,9])과 비교 평가한다.

Table 8에서 프라이버시 보호 항목을 보면 기존 모델에서는 인증과정에서 키 노출이 발생하기 때문에 개인 정보보호를 보장하지 못하고 있지만, 제안 모델에서는 사용자 식별 정보(PID)로 등록이 이루어지고 인증 과정에는 키 생성 알고리즘을 통해 만들어진 사용자 비밀키 k_{se}와 접근제어키k_{ac}를 이용하여 인증 절차가 이루어지기 때문에 사용자의 프라이버시가 보호된다.

익명 인증 항목에서는 기존 모델이 사용자 인증과정 중에 사용자의 정보가 노출되어 사용자의 완벽한 익명 인증을 제공하지 못하고 있다. 그러나, 제안 모델에서는 사용자의 완벽한 익명 인증을 보장하기 위해서 사용자 등록 과정에서 인증서버가 사용자의 식별 정보를 PID로 생성하여 사용하기 때문에 사용자의 익명성이 보장된다.

Table 8. Security Evaluation

Schemes	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	Proposed Models
Collusion Resistance	√	√	√	√	×	√	√	√	√
Standard Model	√	×	√	×	√	×	×	×	√
Privacy Protection	×	×	×	×	√	×	×	×	√
Encryption Predicate	MBF	MBF	MBF	TG	MBF	MBF	MBF	MBF	MBF
Computation Outsourcing	×	×	×	×	×	×	×	√	×
Anonymous Authentications	×	×	√	√	×	×	√	×	√
Multi-Authority	×	×	√	√	√	√	√	√	√
Signcryptor Privacy	√	√	√	×	√	√	√	√	√
Public Verifiability	×	×	√	√	×	×	√	×	√

MBF : Monotone Boolean Function TG : Threshold Gate
√ : Satisfaction
× : Not Satisfaction

다중 권한 항목에서는 [6,7] 모델이 사용자 인증 과정에서 사용자의 다중 권한을 지원하지 않지만 제안 모델에서는 사용자의 서비스 접근 권한(pi)에 따라 시비스를 제공수준(AI)을 결정하기 때문에 다중 권한을 지원 한다.

공모 저항성 항목에서는 [10] 모델이 익명 인증을 제공하지 않고 있기 때문에 제3자가 악의적으로 접근할 수 있는 문제점이 있어 공모 저항성을 지원하지 못하지만 제안 모델은 사용자의 비밀키와 접근제어키(k_{se} , k_{ac})를 생성하고 비밀키k_{se}는 인증과정에서 사용자와 인증서버 간 인증확인을 위해 사용되는 키이고, 접근권한을 확인하기 위해 사용되는 접근제어키 k_{ac}는 인증서버와 클라우드 관리자 간에 사용되는 키이다. 따라서 비밀키는 클라우드 관리자가 알 수 없고, 접근제어키는 사용자가 알 수 없다. 따라서 두 개의 이중 키로 인증과정을 수행하는 제안 인증 모델에서는 공모에 대한 저항력을 가지고 있다.

V. 결 론

최근에는 휴대폰, 테블릿과 같은 소형 장치의 기술들이 발달하면서 대용량의 저장장치를 서비스로 제공받는 클라우드 서비스에 대한 관심이 증가하고 있다. 클라우드 컴퓨팅 발전과 함께 다양한 서비스의 제공이 이루어지면서 사용자들에 대한 인증 요구사항도 더 다양해지고 안전성의 요구도 증가하고 있다.

본 논문에서는 클라우드 환경에서 사용자의 인증 효율성을 개선시키기 위해서 인증서버와 클라우드 관리자 사이에서 사용자의 접근 권한을 제공하기 위한 2중 키 기반의 사용자 인증 모델을 제안하였다. 제안 모델에서는 이중 키를 키 생성, 사용자의 인증 권한, 권한 등급 허용 알고리즘 등에서 사용하고 있어 사용자의 접근 권한의 효율성을 향상시켰다. 또한, 제안 모델은 각 알고리즘의 동작과정을 객관적으로 명시화하기 위해서 시퀀스 다이어그램을 통해 동작과정과 기능을 세분화하여, 사용자의 권한정보(AI)에 따라 서비스 접근 허용 등급 값(pi)이 결정되고 그 접근 허용 등급 값을 이용하여 서비스를 이용하기 때문에 불필요한 동작과정을 줄였다. 특히, 제안 모델의 보안 및 기능 평가에 있어서 제안 모델은 공모 저항, 개인정보보호, 암호화 속성, 익명 인증, 다중 속성, 암호서명 개인정보보호, 공개 검증 가능성 등의 기능을 지원한다. 향후 연구에서는 본 연구의 결과를 기반으로 클라우드 환경에서의 현재 운용되고 있는 클라우드 서비스에 사용자 인증 모델을 적용할 계획이다.

References

[1] Sung-Jae Jung and Yu-Mi Bae, "Trend analysis of Threats and Technologies for Cloud Security," Journal of Security Engineering, Vol.10, no.2, pp.199~212, Apr. 2013.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145. Sep. 2011.

[3] Kevin Walsh, John Manferdelli, "Intra-Cloud and Inter-Cloud Authentication," IEEE 10th International Conference on Cloud Computing (CLOUD), pp. 1-8, Sep. 2017.

[4] Primoz Cigoj, Borka Jerman Blazie and Tomaz Klobucar. "an approach in the design of common authentication solution for a multi-platfotm cloud environment," 5th International Conference on Cloud Computing and Service Science. pp. 365-372. Jan. 2015.

[5] H.A. Dinesha and V.K. Agrawal, "Multi-level authentication technique for accessing cloud services," Computing, Communication and Applications(ICCCA), 2012 International Conference on, pp. 1~4, Feb, 2012.

[6] C. Chen, J. Chen, H. Lim, Z. Ahang, and D. Feng, "Combined public key schemes: The case of ABE and ABS," in Proc. Provable Secure, Chengdu, China, pp. 53-69, Sep. 2012.

[7] H.Lui, Y. Huang, and K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext policy attribute-based singncryption," Future Generation Computer System, vol.

52.pp. 67-76, Nov. 2015.

[8] Y. Sreenivasa, "A Secure and efficient ciphertext policy attribute-based signcryption for personal health records sharing cloud computing," Future Generation Computer System, Vol.67, pp.133-151, Feb. 2017.

[9] G. Yu, and F. Cao, "Attribute-based signcryption with hybrid access policy," Peer-to-Peer Networking and Applications, Vol.20, no.1 pp.1-9, Nov. 2015.

[10] G. Han, W. Susilo, Y. Mu, Y. Zhou, and A. Au, "Improving privacy and security in decentralized CP-ABE," IEEE Transactions on Information Forensics and Security, Vol. 10, No. 3, pp.665-678, Dec. 2014.

[11] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in proc. Advances in Cryptology-EUROCRYPT 2011, Tallinn, Estonia, pp.568-588, May.2011.

[12] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," IEEE Transaction on Parallel and Distributed Systems, Vol.20, No. 2, pp.384-394, Feb. 2013.

[13] K. Yang, H. Jia, and K. Ren, "DAC-MACS: Effective data access control for multi-authority cloud storage systems," IEEE Transactions on Information Forensics and Security, Vol.8, No.11, pp.1790-1801, Jul. 2013.

[14] J Li, F. Chen, W. Li, F. Jia, F.Ma, and J. Lou, "Fine-grained access control system based on outsourced attribute-based encryption," in proc. Coputer Security-ESORICS 2013, Egham, UK, pp 592-609, Sep. 2013.

[15] Z. Lai, H. Deng, W.Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," IEEE Transactions on information Forensics and Security, vol. 8, no. 8, pp. 1343~1354, Jul. 2013.

[16] H. Zhang, F. Chen, J. Li, S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," Information Sciences, vol. 379, pp. 42-61, Feb. 2017.

[17] Q. Huang, X. Yang, and C. Wang, "Secure Data Access Control with Ciphertext Update and Computation Outsourcing," IEEE Access, vol.5, pp.12941-12950, Jul. 2017.

[18] G. Wang, Q. Lui, J. Wu, and Y. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," Computer & Security, vol.30, no.5, pp. 320-331, Jul. 2011.

〈저자소개〉

최 정 희 (Jeong-hee Choi) 정회원
1999년 2월: 서원대학교 상업교육학과 학사
2002년 8월: 충북대학교 컴퓨터과학과 이학석사
2019년 2월: 충북대학교 컴퓨터과학과 공학박사
〈관심분야〉 정보보호, 인증, 클라우드

이 상 호 (Sang-ho Lee) 종신회원
1972년 2월: 숭실대학교 전자계산학과 공학사
1981년 2월: 숭실대학교 대학원 전자계산학과 공학석사
1989년 2월: 숭실대학교 대학원 전자계산학과 공학박사
1981년 3월~2018년 8월: 충북대학교 소프트웨어학과 교수
〈관심분야〉 컴퓨터네트워크, 통신보안, 스마트팩토리, IT융합