Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.5, Oct. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.5.1035

클라우드 환경을 위한 효율적인 권한 기반 키 설립 프로토콜

최 정 희,† 이 상 호‡
충북대학교

An Authority-Based Efficient Key Management Protocol for Cloud Environment

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

요 약

최근 IT 기술이 발전함에 따라 클라우드 서비스를 이용하는 사용자의 인증 방법이 다양화 되고 있다. 그러나, 클라우드 서비스를 이용하는 사용자의 인증 정보를 권한에 따라 안전하게 제공하는 연구는 현재까지 미비하다. 본 논문은 Intra 클라우드 환경에서 사용자의 역할 권한에 따라 비밀키와 접근 제어키를 사용하는 분할 인증이 가능한 키 설립 프로토콜을 제안한다. 제안 프로토콜은 사용자의 속성과 생성된 랜덤수(t_1, t_2)를 함께 이용하여 사용자의 접근제어 키와 비밀키를 생성하고, 키 생성 후 사용자의 권한에 따라 사용자의 역할을 분류하기 때문에 사용자의 인증 절차에 사용되는 불필요한 동작과정을 줄일 수 있다. 성능평가 결과 제안 프로토콜은 사용자의 접근제어 키와 비밀키를 분할하여 사용자를 인증하기 때문에 클라우드 환경에서 발생할 수 있는 다양한 공격 유형에 대한 안전성이 보장이 되었고, 키 설립에 이용된 암호문의 크기는 기존 프로토콜보다 $\sum+1$ 만큼 줄였다.

ABSTRACT

Recently, with the development of IT technology, authentication methods of users using cloud services have been diversified. However, research on providing authentication information of a user using a cloud service securely according to authority has not been make until now. In this paper, we propose a key establishment protocol which can perform split authentication using secret key and access control key according to the role authority of user in Intra cloud environment. The proposed protocol generates the access control key and secret key of the user by using the attributes of the user and the generated random number(t_1, t_2), and classifies the roles according to the user's authority after generating the key. Unnecessary operation processes can be reduced. As a result of the performance evaluation, the proposed protocol guarantees the security against various type of attacks that may occur in the cloud environment because the user is authenticated by dividing the access control key and secret key. The size of the ciphertext used to establish the key could be reduced by $\sum+1$ more than the existing protocol.

Keywords: Intra Cloud, Authentication, private key, public key

Received(05. 23. 2018), Modified(07. 19. 2018), Accepted(08. 31. 2018)

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

I. 서 론

최근 클라우드 환경에서 처리되고 있는 서비스의 종류가 다양화되면서 사용자가 클라우드 서비스를 손쉽게 제공받을 수 있는 인증 서비스가 각광을 받고 있다. 특히, Intra 클라우드 환경에서는 클라우드 서비스 종류에 상관없이 개인 맞춤형 인증 정보 관리 및 분석 기술이 필요하다.

현재 Intra 클라우드 환경에서는 데이터 손실 (Data loss), 개인정보보안 이슈 (Privacy issues), Infected Application, 보안 이슈 (Security issues)등의 보안에 대한 이슈가 대두되고 있다. 그러나, Intra 클라우드의 보안 이슈는 기존 인터넷상의 보안문제와 함께 고려되어야 한다. Intra 클라우드 컴퓨팅 환경에서의 비정상적인 인증의 위험은 일반 인터넷 환경보다 더 큰 보안 위협을 초래할 수 있다. Intra 클라우드 환경에서는 저장된 데이터의 접근 권한을 가진 사용자가 아닌 다른 사용자가 인증 허가를 받는 경우에, 소유자(피해자) 개인정보 및 클라우드 저장소에 저장되어져 있는 데이터들에 위협이 될 수 있다[1,2,3].

클라우드를 사용할 때에는 사용자 권한의 범위에 따라 인증의 단계가 여러 단계로 나뉘고 그 단계에 따라 적절한 수준의 인증이 이루어져야 한다. 그렇지 않다면, 암호화된 사용자 데이터를 복호화 할 수 있는 권한을 갖은 사용자 혹은 관리자의 인증정보가 악용되어, Intra 클라우드 자원을 공유하는 다른 사용자들에게 보안의 위협이 된다[4].

Intra 클라우드와 관련된 연구는 현재까지 키 관리와 관련된 다양한 연구(CP-ABE 기법, N-ABE 기법, 속성 기반 기법, 협력기반 기법 등)가 진행되고 있다[5,6,7,8]. 특히, Intra 클라우드 환경에서 사용되는 키 관리 분야에서는 액세스 정책 정의, 암호 해독, 속성 공유, 키 위탁 및 노출 예방 등이 연구되고 있다. CP-ABE 기법에서는 데이터 소유자가 자신의 액세스 정책을 정의하기 위해서 키 소유자의 개인키를 사용한다[5]. 그러나, 이 기법은 사용자의 허가없이 제3자가 개인키를 사용하기 때문에 암호문 해독과 키 위탁 문제가 발생할 수 있는 문제점이 있다. ABE 기법은 CP-ABE와 KP-ABE에 암호 해독 (OD-ABE)을 사용하고 있으며, 해독 연산에 프록시 서버를 만들어 사용하는 특징이 있다[6]. 그러나, 이 기법은 암호문 크기와 암호 해독 비용이 높은 단점이 있다. 속성 기반 기법은 인프라를 추가하지 않고 키 생성 센터와 데이터 저장 센터간에 two-party(2PC) 컴퓨팅을 도입하여 단일 권한 시스템에서 핵심적인 키 위탁 문제에 대한 새로운 솔루션을 제공한다[7]. 그러나, 이 기법은 키 노출로 인한 공격에 취약하기 때문에 클라우드 환경에서는 부적절하다. 협력기반 기법에서는 키 위탁 문제와 키 노출 위협을 해결하고 있다. 이 기법은 비밀키 CPK_1, CPK_2, CPK_3 를 Key Authority(KA), Cloud Server(CS), Client(CL) 등으로 구분하여 각 사용자별 인증이 이루어진다[8]. 이 기법은 키 위탁 문제와 키 노출 위협에는 효과적인 장점은 있지만 키 생성과정에서 발생되는 키 길이가 다른 기법보다 커지는 문제점이 있다.

본 논문에서는 여러 가지의 보안 이슈 중 사용자의 인증이 정상적으로 이루어지지 않을 때의 위협들을 살펴보고, 사용자의 안전하고 효율적인 인증 방법을 위해 사용자들의 권한에 따른 키 관리 프로토콜을 제안한다. 제안 프로토콜은 Intra 클라우드 환경에서 사용자가 손쉽게 클라우드 서비스를 제공받기 위해서 사용자의 권한에 따른 키를 인증 서버로부터 발급받는다. 이 때, 인증 서버는 사용자의 속성과 함께 생성된 랜덤 수(t_1, t_2)를 이용해서 접근제어 키와 비밀키를 생성하게 된다. 제안 프로토콜에서 인증서버는 일반 클라우드 사용자에게 서비스 인증 요청 시 사용할 비밀키를 전송하고, 클라우드 관리자에게는 등록한 사용자의 속성과 함께 접근 제어키를 전송한다. 클라우드 관리자는 사용자의 인증 요청 메시지가 왔을 때, 사용자가 보내온 비밀키와 클라우드 관리자가 저장·관리하고 있는 클라우드 사용자의 속성값과 접근관리키를 비교하여 등록된 사용자임을 확인한다. 등록된 사용자임이 확인이 되면 클라우드 관리자는 인증서버에 사용자 인증을 요청한다. 만약, 사용자가 보내온 비밀키와 클라우드 관리자가 보관하는 사용자의 속성값과 접근제어키가 일치하지 않는다면 인증과정을 더 이상 진행하지 않고 종료한다.

이 논문의 구성은 다음과 같다. 2장에서는 클라우드 환경의 키 관리 인증 프로토콜에 관한 기존 연구에 대해서 알아본다. 3장에서는 클라우드 보안 위협에 안전하고 효율적인 권한 기반의 키 관리 프로토콜을 제안하고, 4장에서는 제안 기법의 보안 평가 및 기존 기법과 비교평가하고 마지막으로 결론을 맺는다.

II. 관련연구

2.1 클라우드 컴퓨팅

클라우드 컴퓨팅은 언제 어디서든(ubiquitous), 간편하게, 요청에 의해서 최소한의 관리 노력과 최소한의 서비스 제공자와의 상호 작용에 의해 빨리 준비되고 배포되는 설정 가능한 공유 컴퓨팅 자원(네트워크, 서버, 스토리지, 어플리케이션, 서비스) 풀에 네트워크 접근을 가능하게 하는 모델이다[3]. 클라우드 서비스는 기업 및 개인을 대상으로 하드웨어·소프트웨어 등의 컴퓨팅 자원을 필요한 만큼만 빌려 쓰고(On-demand) 사용한 만큼 요금을 지불하는 방식(Pay-as-you-go)이다[1,9].

클라우드 모델은 다섯 가지의 핵심 특성(On-demand self-service, Broad network access, Resource pooling, Rapid elasticity, Measured service)와 세 가지의 서비스 모델(SaaS, PaaS, IaaS) 그리고 네 가지의 배치 모델(Private cloud, Community, Public cloud, Hybrid cloud)로 구성된다. 이러한 클라우드 서비스는 다양한 장점이 있지만 활성화가 지연되고 있는 가장 큰 이유는 보안 문제이다. 클라우드 서비스에는 사용자의 개인정보 유출, 사업자의 해킹과 같은 사이버테러로인한 기업의 정보 유출, 클라우드 스토레지에 저장되어있는 개인 및 기업들의 데이터 손실 등의 위험성이 존재하고 있다[1, 10]. Fig. 1.은 클라우드 컴퓨팅 아키텍처를 보여준다.

Intra 클라우드 환경에서는 서비스를 이용하는 사용자의 권한에 따라 인증의 허용 범위가 달라진다. Intra 클라우드 환경에서는 클라우드 사용자는 클라우드 서비스 사용을 위해 사용자 스스로가 사용자 인증을 제어, 설정, 변경하기 어렵다. 그 이유는 일반 사용자가 사용하기에는 설치 절차가 복잡하고, 기본 인프라의 안전한 배치와 서비스의 안전한 사용을 보장하는 정교하고 완벽한 보안 솔루션이 없기 때문이다. 특히 여러 클라우드 시스템에 액세스 할 수 있는 다중 플랫폼 클라우드 및 인터 클라우드에서 사용자와 관리자는 서로 다른 인증 및 권한 부여 시스템과 다양한 로그인 대화 상자를 사용한다[12, 13]. 클라우드 환경에서의 인증은 위험성과 중요성에 대해서 이슈화되고 있지만, 위에서 언급한 여러 가지 이유로 인하여 사용자 권한에 따른 인증의 복잡함은 여전히 존재 한다. 따라서 클라우드 환경에서의 사용자들을 위한 개인정보보호와 데이터 보호의 효율적이고 안정적인 인증시스템이 필요하다.

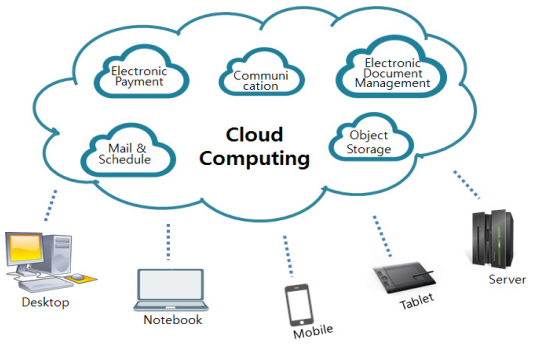

Fig. 1. Cloud Computing Architecture

2.2 기존의 인증방법

최근 많은 논문들에서 클라우드 환경에서의 재전송 공격, 키 위탁 문제, 키 노출 위협, 중간자 공격 등의 위협에 대한 사용자 인증 부분을 연구하고 있다.

Bethenecourt et al.이 제안한 CP-ABE 기법은 데이터 소유자가 자신의 액세스 정책을 정의 할 수 있게 한다[5]. 데이터를 가져 오려는 사용자 먼저 액세스 정책을 속성 집합과 일치 시킨다. 사용자가 정의해 놓은 속성을 이용하기 때문에 CP-ABE는 클라우드 데이터 공유를 위한 안전하고 세분화 된 액세스 제어 구축에 적합하다. 그러나 비밀키 관리에 있어서 이 기법의 ABE방식은 큰 위협을 가지고 있다. 이 기법의 ABE 방식은 사용자의 키를 관리하는 주 위탁 기관이 키 소유자의 사용 허가 없이 생성 된 소유자 개인키를 사용하여 모든 암호문을 해독 할 수 있기 때문에 주 위탁 기관을 완전히 신뢰할 수 있어야한다. 주 위탁 기관을 완전히 신뢰 할 수 없는 경우에는 키 위탁 문제, 키 노출 등의 보안 위협이 발생하며, 사용자 프라이버시의 침해가 발생한다.

Green et al. 기법은 암호문 크기와 암호 해독 비용이 ABE 기법을 사용하는데 있어서 단점이 발생하기 때문에 그 문제점을 극복하기 위해 CP-ABE와 KP-ABE에 암호 해독 (OD-ABE)을 사용하는 새로운 ABE 기법을 제안하였다[6]. 이 기법은 해독 연산의 대부분을 실행하는 프록시 서버를 만들어 사용하여 연산 및 암호화 과정에서의 성능의 효율성을 보장할 수 있게 하였다. 그러나 프록시 서버를 사용할 경우 키 노출 위협으로 인해 사용자의 프라이버시

가 위협받게 된다. 또한, 이 논문은 ABE 방법을 사용하는 기법의 연구들과 동일하게 키 권한을 가진 관리자가 암호문 소유자의 허락 없이 복호화 할 수 있다는 문제점을 갖는다. 이러한 경우에 키 위탁 문제가 발생한다. 따라서 개인 정보보안에 민감한 클라우드 환경에서의 사용은 부적절하다.

Hur 기법은 안전하고 효율적인 속성 기반 데이터 공유 시스템을 제안한 기법이다[7]. CP-ABE는 전통적인 공개키 기반하의 공개키 인증서를 저장·가공 할 필요성이 적어진다는 장점은 있지만, 이러한 장점은 키 위탁 문제라는 큰 단점을 만든다. 이 기법은 인프라를 추가하지 않고 키 생성 센터와 데이터 저장센터 간에 two-party computing(2PC)을 도입하여 단일 권한 시스템에서 핵심적인 키 위탁 문제에 대한 새로운 솔루션을 제공했다. 그러나 이 기법은 개인키가 여전히 front-end 장치에 모두 저장되어있는 경우 키 노출 위협이라는 문제가 개인키의 기밀성을 위협한다. 이러한 문제점이 증대되면 클라우드 환경에서는 보안 위협으로 인해 사용이 부적합하다.

Guoffeng et al. 기법은 키 위탁 문제와 키 노출의 위협을 해결하기 위해 collaborative key 관리 프로토콜을 제안한다[8]. 이 기법은 비밀키 CPK_1, CPK_2, CPK_3 를 Key Authority(KA), Cloud Server(CS), Client(CL)셋이서 나누어 갖고 각 사용자별 인증이 이루어진다. 이 기법은 키 위탁 문제와 키 노출 위협에는 효과적이다. 하지만, 공개키 생성시 효율성에서는 키생성시의 연산의 효율성과 암호문의 크기에서 다른 기법들이 제안한 방법보다 연산의 효율성이 낮다.

III. Intra 클라우드를 위한 속성 기반 키 설립 프로토콜

이 절에서는 Intra 클라우드 환경에서 사용자가 안전하게 서비스를 제공받기 위한 키 설립 프로토콜을 제안한다.

3.1 개요

제안 프로토콜은 Intra 클라우드 환경에서 사용자 역할기반의 권한 인증이 이루어진다. 사용자들의 역할에 따라 권한을 부여하고 그 부여된 권한 정보를 이용하여 비밀키와 접근제어키를 생성하고, 이때 키 생성은 인증서버에서 이루어진다. 접근제어 키 ak_i는 제공자와 일반사용자의 권한에 따라 사용되며, 비밀키 sk_i는 사용자의 인증을 수행하기 위해 사용된다. 키쌍은 인증서버와 클라우드 관리자, 클라우드 사용자가 각각 필요한 정보만을 유지한다.

제안 프로토콜은 역할기반의 키 설립 기법으로 사용자들의 역할에 따라 권한을 부여하고, 속성기반 키 설립은 사용자들의 속성에 따라 키 부여가 이루어진다. 역할기반과 사용자 속성기반 등의 키 설립 특성을 살펴보면, 사용자 속성기반 등의 기법은 사용자 수에 따라 속성 수가 생성되기 때문에 생성되는 키 개수가 사용자 × 속성 수만큼 필요하지만 제안 프로토콜과 같은 역할기반은 사용자의 역할에 따라서 접근 권한 키가 만들어지기 때문에 사용자 속성기반보다 생성되는 키의 수가 적다.

제안 프로토콜에서는 사용자의 속성과 생성된 랜덤수(t_1, t_2)를 함께 이용하여 사용자의 접근제어 키와 비밀키를 생성하고, 키 생성 후 사용자의 권한에 따라 사용자의 역할을 분류하기 때문에 사용자의 인증 절차에 사용되는 불필요한 동작과정을 줄일 수 있으며, 성능평가 결과 제안 프로토콜은 사용자의 접근제어 키와 비밀키를 분할하여 사용자를 인증함으로서 클라우드 환경에서 발생할 수 있는 키노출, 키위탁문제, 중간자공격등의 다양한 공격 유형에 대한 안전성을 보장할 수 있다.

그러나, 제안 프로토콜은 일반사용자의 인증 서버간에 안전한 인증을 수행하기 위해서 제3자가 악의적인 행동을 수행하지 못하도록 다음과 같은 보안 요구사항이 요구된다. 첫째, 제안 프로토콜은 키 생성에 필요한 사용자의 인식자는 사용자 수에 따라 임의의 비트 수열(0과1)로 구성된 N개 수열을 생성해야 한다. 둘째, 인증서버와 클라우드 관리자는 제안 프로토콜이 동작되는 동안 제3자로부터 안전하다고 가정한다. 특히, 클라우드 관리자는 클라우드 서비스 제공자들의 공모는 하지 않는다. 셋째, 인증 서버가 생성한 비밀키와 접근제어키는 사용자의 인증 권한에 함께 사용된다.

3.2 구성요소

제안 프로토콜이 동작되는 클라우드 구성요소는 인증서버, 클라우드관리자, 사용자 등으로 구성된다.

· 인증서버 : 클라우드 환경에서 사용자 역할기반의 권한에 따른 인증을 위한 비밀키 sk_i 와 접근제어키 ak_i 를 생성하여 사용자에 전송한다. 사용자는 인증서버로부터 받은 키 쌍을 이용하여 사용자 인증을 요청하고, 인증서버는 사용자의 인증요청에 대한 정당한 사용자임을 확인하는 인증확인절차를 수행한다.

· 클라우드 관리자 : 사용자들의 역할에 따른 권한 관리를 수행하고, 인증서버의 사용자에 대한 인증확인 후의 사용자 권한에 따른 서비스 허가를 수행한다. 또한, 클라우드 서비스 사용자의 서비스 사용을 위한 인증요청이 오면 가장 먼저 클라우드에 등록된 사용자인지를 확인하고 등록되어 있지 않은 사용자라면 사용자 등록 요청을 한다.

· 사용자 : 제안 프로토콜에서의 클라우드 사용자는 서비스를 사용하는 일반사용자와 서비스를 제공하는 클라우드 프로바이더로 구분된다. 서비스를 사용하는 사용자는 클라우드에 접속하면 가장 먼저, 인증서버에 사용자 등록을 하고 서비스 인증을 시작한다. 제안 프로토콜은 사용자 인증 중 클라우드 서비스 사용자 인증에 필요한 키 설립 프로토콜이다.

3.3 용어정의

제안 프로토콜에서 사용되는 표기법은 아래 Table 1.과 같다.

Table 1. Notation

Symbol	Definition
PID	User ID
(t_1, t_2)	Generate user′ random number
K_2	Shared key, Between AS and CM
K_1	Session Key, Between AS and CM
Z_q^*	A set of prime numbers q
ak_i	Access control key according to user authority
sk_i	User private key
sn	Sequence number
G_i , G_2	A set of N sequences consisting of arbitrary bit sequences (0 and 1)
pi_k	Authority Information for User
PI	User attribute information value

3.4 키 설립 프로토콜

제안 프로토콜의 키 설립은 다음과 같이 크게 2가지 과정(초기화 과정, 인증과정)으로 구성된다.

3.4.1 초기화 과정

초기화 과정은 인증 서버와 사용자 간에 원활한 인증을 수행하기 위해서 사전에 사용자 정보 등록을 위한 키 설립 과정이다.

클라우드 사용자는 권한에 따라 클라우드 서비스를 제공하는 제공자와 클라우드 서비스를 사용하는 일반 사용자로 구분되며, 아래 과정은 클라우드 서비스 사용자의 키 설립 초기화 과정이다. 키 설립 초기화 과정의 세부과정은 Fig. 2.와 같이 7단계로 동작된다.

• 단계 1 :

사용자는 클라우드 서비스를 제공받기 위해서 클라우드 관리자에 사용자의 인식자를 전달한다. 이때, 클라우드 관리자에 전달되는 사용자 인식자는 사용자의 권한에 따라 식 (1)~식 (3)과 같이 생성한다.

$$Generate\ s_i \cong \{0,1\}^* \rightarrow \{0,1\}^N \quad (1)$$

$$idx = \{h_i | s_i, i \in N\} \quad (2)$$

$$PID = h_i \oplus idx \quad (3)$$

식 (1) 에서 사용자의 인식자는 사용자 수에 따른 임의의 비트 수열(0과1로 구성된 수열)로 N개 생성한 후, 식 (2)에서 무작으로 해쉬체인 h_i을 선택하여 사용자의 인덱스 값 idx를 생성한다(Step1-1). 식 (3)에서는 생성된 idx와 함께 XOR하여 생성된 PID를 인증 서버에 전달된다(Step1-2).

• 단계 2 :

클라우드 관리자는 클라우드 사용자로부터 전달받은 사용자 PID가 데이터베이스 사용자 테이블에 등록된 사용자인지 확인하기 위한 확인요청 쿼리를 데이터베이스에 보낸다(Step 2-1). 데이터베이스는 클라우드 관리자로부터 받은 사용자 PID 확인요청 결과(등록된 사용자 또는 등록되지 않은 사용자임의

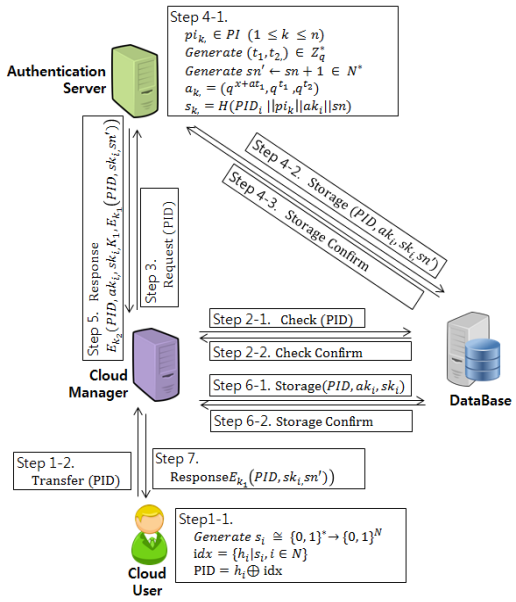

Fig. 2. Initialization Process

확인 결과)를 알려준다. 이때, 데이터베이스는 전달 받은 PID가 등록된 사용자 PID이면 시퀀스 넘버(sn')의 만기여부를 확인하여 클라우드 관리자에 알려준다(Step 2-2).

- 단계 3 :

데이터베이스로부터 받은 확인요청 결과 사용자 테이블에 등록되어 있지 않은 사용자 PID 라면, 클라우드 관리자는 인증서버에 사용자 PID를 전송하여 사용자 등록을 요청 하고, 등록 되어 있는 사용자 중에 시퀀스 넘버(sn')가 만기 되어 초기화가 필요한 경우에 클라우드 관리자는 인증서버에 사용자 sn'의 초기화 설정을 요청한다.

- 단계 4 :

클라우드 관리자로부터 사용자 등록 요청을 받은 인증 서버는 식 (4)~ 식 (5)와 같이 사용자의 속성과 함께 랜덤수 $(t_1,\ t_2)$를 생성한다. 이때, 인증 서버는 식 (4)처럼 사용자의 권한 정보 $pi_k \in$ PI에 따라 역할이 구분되게 된다. 예를 들어, 제안 프로토콜에서 PI는 public, authentication, private의 등급을 의미하며 세부적인 PI사용 예는 Table. 2와 같다. 등급이 낮을 경우 서비스 Reject, 등급이 높은 경우 서비스 Permit을 의미한다. Table. 2에서

Table 2. PI Information

PI Information			Level
Public	Authentication	Private	
0	0	0	0(Reject)
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7(Permit)

처럼 등급은 PI정보 (public, authentication, private)에 따라 Reject과 Permit 사이에 등급을 세분화할 수 있다. PI정보 (public, authentication, private)는 서비스 제공 유무에 따라 0과 1로 나타낸다.

$$pi_k \in \mathrm{PI}\ (1 \le k \le n) \tag{4}$$

$$Generate\ (t_1,\ t_2) \in Z_q^* \tag{5}$$

인증서버는 사용자의 역할 구분(pi_k)에 따라 사용자의 접근제어 키 ak_i와 비밀키 sk_i를 식 (6) ~ 식 (8)과 같이 생성한다. 여기서, ak_i는 클라우드 관리자가 사용자의 등록 요청에 관한 접근제어를 위해 사용되는 키를 의미하고, sk_i는 인증서버 사용자의 서비스 요청을 인증하기 위해 사용되는 키를 의미한다. ak_i는 솟수 원소로 구성된 집합 Z_q^* 중 사용자의 속성과 함께 랜덤수 $(t_1,\ t_2)$를 식(7)처럼 묶어 생성한다. 식 (8)에서 sk_i는 식 (7)에서 생성한 ak_i에 따라 해쉬 값이 달라지기 때문에 제3자로부터 sk_i의 악용을 방지할 수 있다.

$$Generate\ sn' \leftarrow sn+1 \in N^* \tag{6}$$

$$ak_i = (q^{x+at_1},\ q^{t_1},\ q^{t_2}) \in Z_q^* \tag{7}$$

$$sk_i = H(PID_i || pi_k || ak_i\ || sn') \tag{8}$$

식 (6)에서 sn'은 시퀀스 넘버로 클라우드 사용

자 서비스 요청을 위한 메시지와 함께 전송되며, 인증요청마다 1씩 증가한다. 시퀀스 번호가 일정 횟수가 넘어가면 인증서버에서는 새로운 값으로 sn'을 초기화한다(Step 4-1). 이때, 식 (4) ~ 식 (8)의 과정으로 생성된 사용자의 PID, 접근 제어 키 ak_i, 비밀키 sk_i, sn'는 저장 쿼리를 이용하여 사용자 관리 테이블에 저장한다(Step 4-2). 사용자의 정보의 저장이 정상적으로 이루어 진 후, 데이터베이스는 인증서버에 확인 응답을 한다(Step 4-3).

• 단계 5 :

인증서버는 클라우드 관리자에게 클라우드 관리자의 사용자 정보테이블에 저장 될 사용자 정보 PID, 접근제어 키 ak_i, 비밀키 sk_i와 클라우드 사용자에게 전송되어질 암호화된 사용자의 등록 정보 $E_{k_1}(PID,\ sk_i,\ sn')$을 암호화 하여 전송한다. 여기서 K_1은 인증서버와 클라우드 관리자 간 세션이 열려졌을 경우에만 임시적으로 사용하는 세션키를 의미한다. 이때, 암호화 키는 인증서버와 클라우드 관리자 간에 사전 공유된 공유키 K_2로 암호화하여 식 (9)와 같이 전달한다.

$$E_{K_2}(PID,\ ak_i,\ sk_i,\ K_1,\ E_{k_1}(PID,\ sk_i,\ sn')) \quad (9)$$

• 단계 6 :

인증 서버로부터 클라우드 관리자에게 전달된 식 (9)는 클라우드 관리자와 인증 서버간에 사전에 공유된 공유키(E_{K_2})로 복호화 한 후, 사용자 정보 (PID, ak_i, sk_i)를 데이터베이스 사용자 관리 테이블에 저장한다(Step 6-1). 데이터베이스는 사용자 관리 테이블에 사용자의 정보 (PID, ak_i, sk_i)를 저장하고 저장 확인 응답을 클라우드 관리자에게 한다(Step 6-2).

• 단계 7 :

클라우드 관리자는 인증서버로부터 전달받은 $E_{k_1}(PID,\ sk_i,\ sn')$을 사용자에게 전송한다. 클라우드 사용자는 클라우드 관리자로부터 인증 정보를 받고, 인증서버와 사전에 공유된 공유키 K_1로 복호화한 후 비밀키 sk_i와 시퀀스 넘버 sn'을 획득한다.

3.4.2 인증과정

인증 과정은 사용자가 클라우드 공급자가 제공하는 클라우드 서비스를 받기위한 사용자 서비스 인증 과정이다. 사용자의 PID는 사용자의 권한에 따라 일반사용자와 클라우드 서비스 공급자로 구분되어지는 식별자이다.

인증 과정의 세부과정은 Fig. 3. 과 같이 6단계로 동작된다.

• 단계 1 :

사용자는 클라우드 서비스를 제공받기 위해서 클라우드 관리자에게 사용자 정보를 식(11) 과 같이 전송하고 서비스 인증을 시작한다(Step 1-2). 이때 식(10) 의 sn'은 사용자 등록과 함께 인증서버로부터 부여 받은 시퀀스 넘버로 메시지 요청을 할 때마다 증가되어 서비스 요청 메시지와 함께 전송된다(Step 1-1). 시퀀스 번호가 일정 회수가 넘어가면 새로이 초기화하여 사용한다.

$$Generate\ sn' \leftarrow sn+1 \quad \in \quad N^* \quad (10)$$

$$req_{k_i} = (H(sk_i),\ PID,\ sn') \quad (11)$$

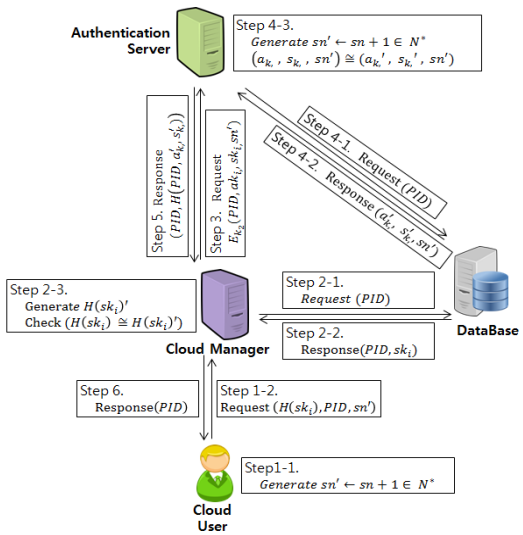

Fig. 3. Authentication Process

• 단계 2 :

사용자로부터 req_{k_i} 서비스 인증요청을 받은 클라

우드 관리자는 데이터베이스의 사용자 정보 테이블에서 PID와 일치하는 비밀키 sk_i값을 쿼리요청한다(Step 2-1). 사용자 정보 요청을 받은 데이터베이스는 클라우드 관리자가 요청한 PID와 일치하는 비밀키 sk_i를 쿼리응답 한다(Step 2-2). 클라우드 관리자는 데이터베이스 쿼리응답으로 얻어낸 사용자의 비밀키 sk_i를 식 (12) 와 같이 해쉬값으로 생성한다. 사용자 비밀키 생성 후, 식 (13) 과 같이 사용자가 보내온 해쉬값 $H(sk_i)$ 과 사용자 관리 테이블에 있는 사용자 비밀키의 해쉬값 $H(sk_i)'$을 비교한다(Step 2-3). 사용자가 보내온 사용자 비밀키 (sk_i)와 클라우드 관리자의 사용자 관리 테이블에 있는 사용자 비밀키$(sk_i)'$가 일치한다면 인증과정을 진행하고, 일치하지 않는다면 더 이상 인증을 진행하지 않고 종료한다.

$$Gererate\ H(sk_i)' \tag{12}$$

$$H(sk_i) \cong H(sk_i)' \tag{13}$$

• 단계 3 :

등록된 사용자임이 확인이 되면 클라우드 관리자는 식(14) 와 같이 사용자 정보를 암호화 하여 인증서버에 사용자 인증을 요청한다. 이때, 사용자 정보는 클라우드 관리자와 인증 서버간에 사전에 공유된 공유키E_{K_2} 로 암호화하여 전송한다.

$$E_{k_2}(PID, ak_i, sk_i, sn') \tag{14}$$

• 단계 4 :

인증서버는 클라우드 관리자로부터 전송 받은 암호화된 사용자 인증 정보 $E_{k_2}(PID, ak_i, sk_i, sn')$를 사전에 공유된 공유키로 복호화 한다. 인증서버는 사용자 인증확인을 위해 사용자 정보 테이블이 저장되어져 있는 데이터베이스로 PID와 일치하는 사용자 정보의 접근제어키ak_i, 비밀키 sk_i, 시퀀스 넘버 sn'을 요청하는 쿼리를 보낸다(Step 4-1). 클라우드 관리자는 사용자 관리 테이블로부터 사용자 정보 PID와 일치하는 접근제어키 ak_i', 비밀키 sk_i', 시퀀스 넘버 sn'를 쿼리응답으로 받는다(Step 4-2). 식 (16)과 같이 인증을 요청한 사용자정보와 저장된 사용자 정보가 일치하는지를 비교한다(Step 4-3). 이때, 인증서버는 식(15) 와 같이 키 설립 시에 생성한 시퀀스 넘버 sn를 증가시켜 sn'를 생성한다.

$$Generate \quad sn' \leftarrow sn+1 \quad \in \quad N^* \tag{15}$$

$$(ak_i, sk_i, sn') \cong (ak_i', sk_i', sn') \tag{16}$$

• 단계 5 :

인증서버는 사용자의 인증요청이 정상적인 사용자의 서비스 요청에 대한 인증임이 확인이 되면 클라우드 관리자에게 식(17) 과 같이 사용자 PID와 사용자 정보의 해쉬값을 전송한다. 만약, 비정상적인 사용자의 서비스요청임이 확인되면 서비스 인증을 종료한다.

$$resp_{k_i}(PID, H(PID \,||\, sk_i \,||\, ak_i)) \tag{17}$$

• 단계 6 :

인증서버로부터 사용자인증을 확인받은 클라우드 관리자는 클라우드 서비스를 이용하는 일반사용자에 서비스 인증확인하고 클라우드 서비스를 개시한다.

IV. 평 가

4.1 보안평가

이 절에서는 클라우드 환경에서 발생할 수 있는 다양한 공격 유형(중간자 공격, 재전송 공격, 완전 순방향 비밀성 보장, 키 위탁 문제, 키 노출 등)에 따른 제안 프로토콜의 안정성을 평가한다.

4.1.1 중간자 공격

제안 프로토콜은 사용자가 클라우드 관리자에 클라우드 서비스 요청을 할 때, 서비스 인증 요청은 $req_{k_i} = (H(sk_i), PID, sn')$ 으로 한다. 전송되는 사용자의 비밀키는 일방향함수인 해쉬함수를 이용하였기 때문에 공격자는 사용자의 비밀키를 알 수가 없다. 또한, 전송되는 비밀키 sk_i는 사용자의 접근제어키와 시퀀스 넘버를 해쉬함수를 이용하여 $H(PID_i||pi_k||ak_i||sn')$ 으로 만든 해쉬값이므로 공격자는 사용자의 접근 제어키 ak_i 를 추출 혹은 추측하기

어렵다. 결국 전송 중간에 공격자가 도청을 하였다 하더라도 비밀키는 일방향 함수인 해쉬값으로 전송되기 때문에 그 값을 알기 어렵고, 만약 알아냈다 하더라도 접근제어 키인 ak_i를 추축할 수 없어 중간자 공격을 성공할 수 없다.

4.1.2 재전송 공격

클라우드 사용자는 클라우드 서비스를 요청(req_{k_i}) 할 때, 비밀키 sk_i 를 일방향 함수 해쉬값 $sk_i = H(PID_i||pi_k||ak_i||sn')$ 으로 만들어 전송한다. 이 때 sk_i의 sn의 값은 서비스 요청을 할 때마다 매번 바뀌는 시퀀스 넘버이다. 사용자의 비밀키를 가로채기 등의 공격으로 공격자가 알아냈다 하더라도, 비밀키를 재사용 할 때 sn'의 값이 매번 바뀌기 때문에 똑같은 값으로 서비스 요청 $req_{k_i} = (H(sk_i), PID, sn')$ 이 온다면 클라우드 관리자는 재전송공격임을 알아챌 수 있다. 따라서 공격자의 재전송 공격이 성공 할 수 없다.

4.1.3 완전순방향비밀성 보장

제안하는 키 설립 프로토콜은 공격자의 도청 및 가로채기가 성공하여 비밀키 sk_i가 공개되어도, $sk_i = H(PID_i||pi_k||ak_i||sn')$와 같이 일방향 해쉬암호화 방식으로 sk_i를 생성하였기 때문에 사용자의 접근제어 키인 ak_i을 알 수 없고, 비밀키 sk_i 안에 시퀀스 넘버 sn'이 전송 할 때마다 바뀌는 값이기 때문에 완전순방향비밀성을 보장 할 수 있다.

4.1.4 키 위탁 문제

제안 키 설립 프로토콜은 클라우드 인증서버에서 사용자의 인증을 위한 키 쌍을 만들고 인증 서버와 클라우드 관리자가 각각 사용자 비밀키(sk_i)와 접근제어키(ak_i)를 분산하여 보관하고 있고, 사용자는 자신의 비밀키 $sk_i = H(PID_i||pi_k||ak_i||sn')$를 보관하고 서비스 요청을 위한 인증을 시작할 때 사용자 자신의 비밀키 sk_i 를 이용하여 인증을 시도한다. 따라서 제3자인 인증서버나 클라우드 관리자의 인증 시스템에서 키의 변경이 일어났다면 사용자 및 다른 서버에서 변경을 알아낼 수 있다. 따라서 키 위탁 문제 발생을 사용자는 알아챌 수 있고, 공격자의 키 위탁 문제를 이용한 공격을 차단할 수 있다.

4.1.5 키 노출

클라우드 사용자가 인증을 위해 사용자 자신의 비밀키(sk_i) 를 클라우드 관리자에게 전송할 때 전송되는 비밀키는 $h(sk_i)$와 같이 일방향 해쉬암호 방식으로 전송 되기 때문에 사용자의 비밀키 sk_i는 공격자가 알기 어렵다. 또한, 클라우드 사용자의 접근제어키(ak_i) 는 $sk_i = H(PID_i||pi_k||ak_i||sn')$와 같이 사용자 비밀키($sk_i$) 안에 일방향 해쉬암호로 계산되어 있기 때문에 사용자의 접근제어키(ak_i)의 키 노출은 발생하지 않는다. 제안 키설립 프로토콜에서는 사용자의 접근제어키(ak_i)를 인증서버가 생성한 후 클라우드 관리자에게 암호화하여 전송하고 인증서버와 클라우드 관리자는 각각 데이터베이스에 사용자의 접근제어키(ak_i)를 저장 관리한다. 사용자의 인증 요청을 받은 클라우드 관리자는 자신이 관리하는 테이블에서 인증 요청이 온 사용자의 접근제어키(ak_i)를 확인한 후, 인증서버에 사용자 인증을 요청한다. 결국 사용자의 권한에 따른 접근제어키 ak_i는 인증 요청시에 클라우드 관리자가 인증서버에 전송하지 않기 때문에 키 노출이 되지 않는다. 따라서 키 노출 위협으로부터의 안전성이 보장된다.

4.2 공개키/비밀키/암호문 크기

[8] 의 프로토콜 비교 분석 결과를 토대로 제안 프로토콜의 성능평가를 비교 분석한다. Table 3에서 G_1과 G_2는 0과 1의 조합으로 구성된 곱셈 순환 그룹의 솟수 q를 의미한다. H 함수는 0과 1로 구성된 곱셈군의 비트열($\{0,1\}^*$)로 구성된 그룹(G_1)으로 해쉬한다. H_1 는 곱셈군의 비트열($\{0,1\}^*$)로 구성된 그룹(G_2) 중 솟수 q에 대한 Z_q^*를 포함한 해쉬 함수이다. H_2는 $G_1 \times G_2$으로 나타낸 집합(G_3)의 해쉬 함수이다. Table 3은 제안 프로토콜과 ABE를 변형한 대표적인 방법인 Bethenecourt et al. [5], Green et al. [6], Hur [7], Guoffeng et al. [8]의 Public Key size, CipherText size, Private Key size의 크기를 비교 분석한 표이다. 제안 프로토콜 Public Key size는 $|G_1|+|G_2|$로

Table 3. Comparison of size of Public Key, Private Key and Ciphertext size

Protocol	Public Key size	Cipertext size	Private Key size
[5]	$\|G_1\|+\|G_2\|$	$(2\sum+1)\|G_1\|+\|G_2\|+\|G_3\|$	$(2\|S\|+1)\|G_1\|$
[6]	$\|G_1\|+\|G_2\|$	$(2\sum+1)\|G_1\|+\|G_2\|+\|G_3\|$	$\|Z_q^*\|+(\|S\|+2)\|G_1\|$
[7]	$\|G_1\|+\|G_2\|$	$(2\sum+1)\|G_1\|+\|G_2\|+\|G_3\|$	$(2\|S\|+2)\|G_1\|$
[8]	$(\sum+1)\ \|G_1\|+\|G_2\|$	$(\sum+1)\|G_1\|+\|G_2\|+\|G_3\|$	$\|Z_q^*\|+(\|S\|+2)\|G_1\|$
Proposed protocol	$\|G_1\|+\|G_2\|$	$(\sum+1)\|G_1\|+\|G_2\|+\|G_3\|$	$\|Z_q^*\|+(\|S\|+1)\|G_1\|$

$H : \{0,1\}^* \rightarrow G_1,\ G_2 \qquad H_1 : G_2 \rightarrow Z_q^* \qquad H_2 : G_1 \times G_2 \rightarrow G_3$

Guoffeng et al. [8] 이 제안한 기법의 Public Key size $(\sum+1)\ |G_1|+|G_2|$ 보다 $(\sum+1)$ 만큼 작은 크기를 갖는다. 또한 제안 프로토콜의 CipherText size는 $(\sum+1)|G_1|+|G_2|+|G_3|$로 Guoffeng et al. 프로토콜[8] 이 제안한 방법과는 같지만 Bethenecourt et al. [5], Green et al. [6], Hur [7] 의 기법보다 $(\sum+1)$ 만큼 작은 크기를 갖는다. 제안 프로토콜의 Private Key size 는 $|Z_q^*|+(|S|+1)|G_1|$ 로서 다른 제안 방법들과 비교 했을 때 Green et al. [6], Guoffeng et al. [8]의 크기보다 $(|S|+1)$ 만큼 작은 크기를 나타낸다.

V. 결 론

클라우드 컴퓨팅 기술이 발전하고 다양한 형태의 클라우드 서비스가 이용되면서 사용자 인증에 대한 요구사항이 증가하고 있다. 본 논문에서는 Intra 클라우드 서비스를 사용하는 사용자의 역할 권한에 따라 비밀키와 접근 제어키를 사용하는 분할 인증이 가능한 키 설립 프로토콜을 제안하였다. 제안 프로토콜에서는 사용자의 속성과 함께 생성된 랜덤 수(t_1, t_2)를 이용하여 사용자의 접근제어 키와 비밀키를 생성하였기 때문에 사용자의 권한에 따라 인증 절차가 줄여졌다. 성능평가 결과 제안 프로토콜은 사용자의 접근제어 키와 비밀키를 분할하여 사용자를 인증하기 때문에 클라우드 환경에서 발생할 수 있는 중간자 공격, 재전송 공격, 키 노출 위협, 완전순방향비밀성, 키 위탁 문제 등의 공격 유형에 대한 안전성이 보장이 되었고, 키 설립에 이용된 암호문의 크기는 기존 프로토콜보다 $\sum+1$ 만큼 줄일 수 있었다. 향후 연구에서는 본 연구 결과를 기반으로 인터 클라우드 환경에서의 키 설립 프로토콜에 적용하고 한다.

References

[1] Do-hyeon Choi and Jung-oh Park, "Multi-session authentication scheme for secure authentication and session management of cloud services environment," Journal of the Korea Institute of Information and Communication Engineering, Vol. 19, No. 9, pp. 2056~2063, Sep. 2015

[2] R Charanya and M Armudhan, "Survey on Access Control Issues in Cloud Computing," Emerging Trends in Engineering, Technology and Science (ICETETS), International Conference on. pp.164~167, Feb. 2016

[3] Yannan Li, Young Yu, Geyong Min, Willy Susilo, Jianbing Ni and Kim-Kwang Raymond Choo, "Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems," IEEE Transactions on Dependable and Secure Computing, Journal of Latex Class Files, Vol. 14. No. 8, pp. 1~12, Aug. 2015

[4] Kevin Walsh and John Manferdelli, "Intra-Cloud and Inter-Cloud Authe-

ntication," Cloud Computing (CLOUD), IEEE 10th International Conference on. pp.1~8, Jun, 2017
〔5〕 J. Bethencourt, A and Sahai, B.Waters, "Ciphertext-policy attribute-based encryption," in Proc. IEEE Symp. Secur. Privacy, pp. 321~334, May. 2007
〔6〕 M. Green, S. Hohenberger and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in Proc. USENIX Secur. Symp., pp. 34~40. Aug, 2011
〔7〕 J. Hur, "Improving security and efciency in attribute-based data sharing," IEEE Trans. Knowl. Data Eng., Vol. 25, No. 10, pp. 2271~2282, Oct. 2013
〔8〕 Guofeng Lin, Hanshu Hong and Zhixin Sun. "A Collaborative key Management Protocol in ciphertext policy Attribute-Based encryption for Cloud Data Sharing," IEEE Acess. vol 5. pp. 9464~9475. May. 2017
〔9〕 KISIA, "Changes in the IT ecosystem, according to a spreading cloud services and Countermeasure," Korea IT Service Industry Association, 2012
〔10〕 Peter Mell and Timothy Grance. "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, Sep. 2011
〔11〕 Sung-Jae Jung and Yu-Mi Bae, "Trend analysis of Threats and Technologies for Cloud Security," Journal of Security Engineering, 10(2), pp.199~212, April, 2013
〔12〕 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
〔13〕 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

〈저자소개〉

최 정 희 (Jeong-hee Choi) 정회원
1999년 2월: 서원대학교 상업교육학과 졸업
2002년 8월: 충북대학교 컴퓨터과학과 석사
2013년 3월~현재: 충북대학교 컴퓨터과학과 박사과정
〈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