클라우드 환경에서 다중 인가자와 계층적 속성기반 암호화를 활용한 접근제어 시스템에 대한 연구

이진아*, 정준권*, 정성민*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신대학
{jinalee,jkjung,smjung}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

A Study on Access Control System with Multi-Authority and Hierarchical Attribute-Based Encryption in Cloud Environment

Jin-A Lee*, Jun-Kwon Jung*, Sung-Min Jung*, Tai-Myoung Chung**
*Dept. of Electrical and Computer Engineering, Sungkyunkwan University
**School of Information Communication Engineering, Sungkyunkwan University

요 약

클라우드 시스템에서는 데이터 소유자가 아닌 클라우드 서비스 제공자가 각 개인의 데이터에 대한 저장과 관리를 책임진다. 따라서 클라우드 서버 상의 사용자 데이터에 대한 보안을 보장해 주는 것이 가장 중요한 이슈이다. 데이터 보안 문제는 안전하고 효율적인 접근제어 기술을 통해 해결할 수 있다. 기존 시스템에서 많이 이용되고 있는 RBAC(Role based access control)은 접근제어의 형태가 주로 수직적이고, 데이터 접근가능 여부를 역할이라는 고정적인 값에 따라 결정하기 때문에 동적인 클라우드 환경에 적합하지 않다. 반면 HASBE(Hierarchical attribute set based encryption) 모델은 ABAC(Attribute based access control)를 통해 유연하고 탄력적인 접근제어를 제공한다. 또한 HASBE 는 인가자(Authority)와 사용자의 관계 모델이 계층적인 구조를 갖고 있기 때문에 큰 조직에서 수많은 사용자들의 데이터 관리와 키 분배를 좀더 효율적으로 할 수 있다. 본 논문에서는 위의 계층적인 모델에서 더 나아가서, 실제 클라우드 환경에서 데이터가 가질 수 있는 복잡한 속성과 인가자의 관계를 고려해 다중 인가자의 개념이 더해진 모델을 제안한다.

1. 서론

클라우드 컴퓨팅 기술은 풍부한 컴퓨팅 자원과 저장공간 제공, 경제적 효율성과 같은 강점 때문에 학계와 산업계에서 주목 받고 있다. 뿐만 아니라, 이미 구글, 마이크로소프트, 아마존 등의 기업들은 클라우드 서비스를 제공하고 있다.

하지만 이러한 강점과 동시에 보안의 위협 또한 존재한다. 기존의 시스템에서 데이터의 소유자는 데이터를 본인의 저장장치에 저장 해 놓고 사용하기 때문에 데이터의 관리와 제어가 전적으로 데이터의 소유자에 달려있다. 반면에, 클라우드 데이터 서비스를 이용하면 내 데이터가 어디에 저장되어 있는지도 알 수 없고 데이터에 대한 관리 또한 제 3 자인 클라우드 서비스 제공자에게 맡겨야 한다. 이러한 특성들 때문에 클라우드 컴퓨팅에서 데이터 보안의 중요성은 더욱 부각된다. 실제로 최근 마이크로소프트의 한 조사에 따르면 일반인의 90%는 클라우드 서비스의 프라이버시와 보안에 대한 우려를 가지고 있다고 한다[1].

이러한 보안문제들은 클라우드 서버에 저장된 데이터에 누가 접근하는지 알지 못한다는 사실에서 생겨난다. 따라서 클라우드 시스템의 보안 성능은 데이터에 대한 접근제어(Access control)를 얼마나 효율적이고 안전하게 잘 하는가에 달려있다고 볼 수 있다.

접근제어 기술은 클라우드 컴퓨팅에 대한 관심이 높아지기 이전부터 활발히 연구되었던 분야이다. 대표적인 접근제어 방식 중 하나가 역할기반 접근제어(RBAC)이다[2]. 자원을 탄력적으로 제공하는 클라우드 시스템의 특성상 유연성 있는 접근제어의 제공이 상당히 중요한데, RBAC 은 사용자의 속성에 따른 유연성 있는 접근제어를 제공하기 힘들다는 문제가 있다. 따라서 본 논문에서는 ABE(Attribute based encryption) 기술을 이용한 속성기반 접근제어(ABAC)를 통해 사용의 속성에 따른 유연하고 동적인 접근제어가 가능하도록 한다[3].

HASBE(Hierarchical attribute set based encryption)는 ABE 기술을 이용하는 모델 중 하나이다. 이 모델은 ABE 를 통해 유연성을 보장함과 동시에 키와 속성들을 관리하는 인가자들을 계층적인 구조로 설계함으로

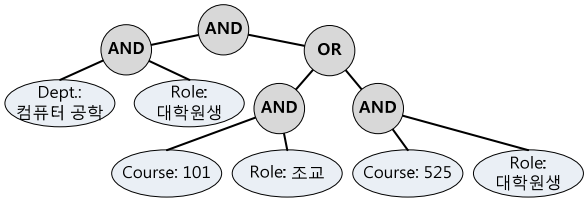
써 시스템 관리의 효율성을 높였다[4]. 하지만 이 모델도 한 인가자가 그에 속한 사용자의 모든 속성을 관리하는 구조로, 실제 클라우드 환경의 데이터 특성에 적용하기에는 한계가 있다. 따라서 본 논문에서는 여러 인가자로부터의 속성 관리가 가능한 다중인가자(Multi-authority)의 개념이 추가된 모델을 제안한다.

먼저 2 장에서는 ABE 와 ABE 의 확장인 ASBE 그리고 계층적 모델을 제안한 HASBE 에 대해 알아보고 3 장에서 HASBE 의 한계와 본 논문의 제안 모델에 대해 상세히 설명한 후 4 장에서 결론을 맺는다.

2. 관련연구

2.1. Attribute Based Encryption (ABE)

ABE 는 데이터와 사용자에 속성을 부여해서 사용자의 속성이 데이터의 속성을 만족시킬 때만 데이터를 복호화할 수 있도록 하는 암호화 방식이다. 각각의 데이터 파일은 그 파일의 특성을 설명할 수 있는 어떤 속성 집합과 연관 지어진다. 모든 속성들에는 그에 해당하는 공개키가 정의되고 데이터 파일은 각자의 속성 집합에 해당하는 공개키 집합을 이용해서 암호화된다. 데이터 접근에 대한 정책은 접근구조(Access structure) 라는 것을 통해 나타내는데, 접근 구조는 (그림 1)과 같이 데이터 속성에 대한 접근트리(Access tree)를 이용해 표현한다. 접근트리의 내측 노드들은 임계값 게이트(Threshold gate)가 되고, 리프노드들은 속성들과 연관되어있다. 클라우드 시스템에서는 사용자와 데이터의 양이 고정되어 있지 않고 동적으로 변하는 경우가 대부분이다. 속성기반 접근제어를 이용하면 데이터를 암호화 하는 계산에 대한 복잡도가 데이터 파일에 관련된 속성의 수에 관련되기 때문에 사용자가 늘어나더라도 계산의 효율성이 떨어지지 않는다는 장점이 있다.

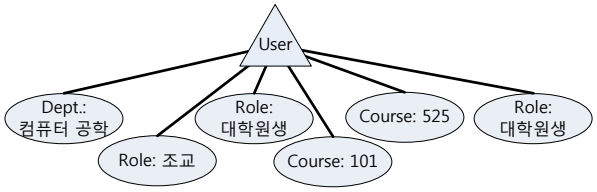


(그림 1) 접근트리

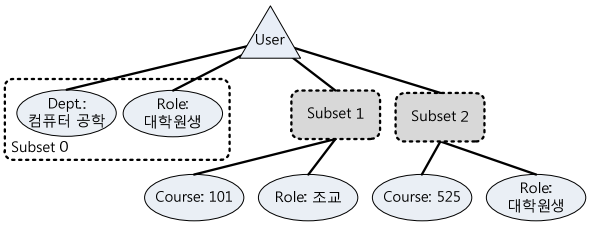
ABE 는 이러한 접근정책이 키와 연관되는지 암호문과 연관되는지에 따라 KP-ABE(Key-policy attribute-based encryption) 와 CP-ABE(Ciphertext-policy attribute-based encryption) 로 나누어진다[5, 6]. KP-ABE 에서는 각 사용자가 자신이 어떤 데이터에 접근할 수 있는지에 대한 접근트리를 가지고 있고, 사용자의 비밀키는 이 접근구조를 반영하도록 할당된다. 이 방식은 사용자가 데이터 접근 정책을 갖고 있기 때문에 데이터를 암호화하는 주체는 자신이 암호화한 데이터에 누가 접근하는지에 대한 결정권이 없다는 한계가 있다. 이와 반대로 데이터 소유자가 자신의 데이터에 누가 접근 할 수 있는지에 대한 정책을 결정하도록 하는 방식이 CP-ABE 방식이다. 메시지를 암호화 하려고 하는 주체가 접근트리를 이용해서 메시지에 대한 접근정책을 명시하기 때문에 암호화된 메시지와 접근트리가 연관되고, 사용자들은 본인의 속성에 해당되는 개인 키를 할당 받아 반드시 속성들이 이 정책을 만족하는 경우에만 메시지를 복호화 할 수 있다.

2.2. Attribute Set Based Encryption (ASBE)

R. Bobba 는 좀 더 복잡한 속성 관계를 표현하기 위해 CP-ABE 의 확장인 ASBE(Attribute set-based encryption)을 제안하였다[7]. CP-ABE 가 단일 집합 내의 원소들의 형태로 속성을 명시한다면, ASBE 는 속성을 반복적인 집합의 형태로 구성한다. 즉, 속성 집합의 원소가 단일 원소가 아닌 또 다른 원소의 집합이 될 수 있는 키 구조(Key structure)이다. 키 구조의 깊이는 트리에서 깊이의 의미와 비슷하게, 반복 집합에서의 반복의 단계를 나타낸다. (그림 1)의 접근트리 내의 속성들을 가진 사용자의 키 구조는 (그림 2)와 같이 나타낼 수 있는데, 이를 ASBE 를 이용해 깊이가 2 인 키 구조로 나타내면 (그림 3)과 같다.



(그림 2) 깊이 1 인 키 구조



(그림 3) 깊이 2 인 키 구조

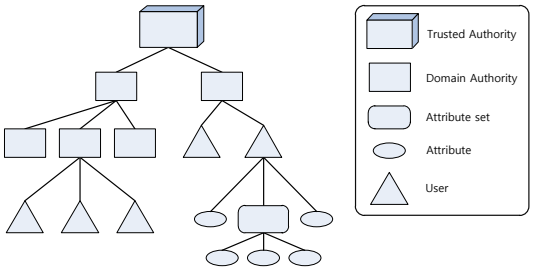
위 예제는 컴퓨터 공학과의 대학원생이면서 수업 101 의 조교를 맡고 있고 수업 525 를 수강하는 학생에게 할당되는 키 구조이다. 집합 내에 또 다른 집합의 형태로 원소를 갖는 것을 허용함으로써 ASBE 는 좀 더 구체적이고 유연하게 실질적인 시나리오를 지원할 수 있다. CP-ABE 를 이용하면, 사용자가 수업 525 를 수강하는 것인지 그 수업의 조교를 맡고 있는 것인지 구분할 수 없기 때문에 525 와 조교, 두 가지 속성을 결합해 허용되어선 안될 권한을 획득할 수 있다. 이 때, ASBE 를 이용하면 수업 525 와 조교, 이 두 가지 속성은 서로 다른 집합에 속해 있기 때문에 잘못된 결합을 막을 수 있다.

2.3. Hierarchical Attribute Set Based Encryption

Z. Wan 은 CP-ASBE 를 이용하고 시스템 사용자를 계층적으로 구성해 키 관리의 효율성을 높인 HASBE 라는 새로운 시스템 모델을 제안하였다[4].

데이터 소유자는 공유하려고 하는 데이터 파일을

암호화해 클라우드 서버에 올려 데이터 소비자와 공유한다. 데이터 소비자는 클라우드로부터 암호화된 데이터를 다운받아 복호화하여 사용한다. 이때 데이터 소유자와 데이터 소비자는 각각 해당하는 부모 도메인 인가자(DA)가 관리한다. 이 DA 는 또다시 부모 DA 나 신뢰 인가자(TA)에 의해 관리된다. 전체적인 구성은 (그림 4)와 같다. TA 는 신뢰 루트(trusted root)로써 최상위 DA 들을 관리한다. 최상위 레벨의 DA 들은 가장 상위에 있는 기관을 나타내고, 그 아래 레벨의 DA 는 각 기관에 속한 하위 기관이다. DA 는 자신에게 속한 또 다른 하위 DA 나 사용자(데이터 소유자/소비자)를 관리할 수 있다. 시스템 내의 각 사용자는 사용자의 복호화 키와 관련되는 속성들을 명시하는 키 구조를 할당 받는다.



(그림 4) 계층적 구조

위의 모델에서 한 유저는 오직 자신이 속해 있는 하나의 인가자로부터만 모든 속성을 할당받는다. 하지만 실제로 데이터를 암호화 하는 경우를 생각 해보면 서로 다른 여러 인가자로부터 데이터의 속성을 얻게 되는 경우가 많다. 따라서 본 논문에서는 HASBE 의 계층적인 구조에 다중 인가자로부터의 속성 할당 기능이 더해진 모델을 제안한다[8].

3. 제안 모델

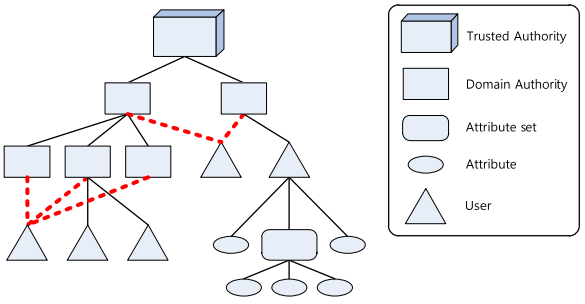
3.1. HASBE 의 한계

HASBE 모델의 암호화 방식에서는 사용자가 자신의 속성에 해당하는 복호화 키를 얻기 위해서 신뢰할 수 있는 관리자에게로 가서 자신의 신원(Identity)과 자신이 해당하는 속성을 가지고 있다는 사실을 증명해야 한다. 이것은 곧 키를 생성하는 하나의 서버가 모든 속성 집합을 관리한다는 것을 의미한다. 하지만 이런 구조를 이용해 실제로 사용할 수 있는 클라우드 시스템을 구현하기에는 부족한 점이 있다. 예를 들어 어떤 사용자가자가 데이터를 클라우드 서버에 올릴 때, A 대 재학생, 수원지역 거주자, 운전면허 보유자 이 세 가지 속성 중 2 개 이상을 만족하는 사용자인 경우에만 데이터를 이용 할 수 있도록 접근제어를 하고 싶다고 하자. 위의 세 가지 속성을 가지고 있는 사용자는 인가자에게 가서 자신을 증명하고 속성에 맞는 비밀키를 받아야 할 것이다. 하지만 실제 상황에서 A 대 재학생에 대한 정보와 수원지역 거주자 정보, 운전면허에 대한 정보를 관리하는 기관은 모두 다르기 때문에 이 모든 속성들을 하나의 인가자가 관리하는 것은 사실상 불가능하다고 할 수 있다. 클라우드 데이터 서버는 특정한 한 기관이 소유하고 관리하는 것이 아니라 다양한 기관과 개인으로부터의 데이터를 저장하고 공유하도록 하기 때문에 클라우드 환경에서 이와 같은 상황은 흔하게 발생할 수 있다. 이런 문제점을 해결하기 위해 제안 모델은 다중 인가자 기반 암호화 방식을 이용한다. 단일 인가자를 사용하는 기존 모델인 (그림 4)에서는 도메인 인가자와 사용자간의 관계가 1 대 n 으로 연결된다. 하지만 다중 인가자를 이용하면 (그림 5)에서 볼 수 있듯이 도메인 인가자와 사용자의 관계가 k 대 n 으로연결된다.

3.2. 다중 인가자 방식의 HASBE

제안 모델에서는 클라우드 서버를 신뢰할 수 없다고 가정한다. 따라서 사용자는 데이터를 저장하기에 앞서 암호화한 데이터를 클라우드 서버에 저장한다. 사용자들은 데이터의 속성으로 구성된 접근 트리를 만들어 접근 정책을 표현하고, 속성에 해당하는 공개키로 데이터를 암호화한다. 이 속성들은 k 개의 인가자가 관리한다. 클라우드 서버에 저장된 데이터를 사용하고 싶은 사용자는 자신이 갖고 있는 속성에 해당하는 복호화 키를 받아야 한다. k 개 인가자로부터 받은 비밀키를 결합하여 데이터를 복호화 하는데 사용하는데, 이때 사용자 간의 공모가 가능하지 않아야 한다. 다중 인가자가 아닌 기존의 시스템에서는 한 인가자가 사용자 별로 다른 난수 값을 생성해 이 값을 이용해서 키를 만들기 때문에 서로 다른 난수 값을 이용한 사용자들의 키를 결합하는 것은 불가능하다. 이 방식은 한 인가자 내에서의 키 결합을 막는 데는 효과적이지만 서로 다른 인가자의 관리 아래에 있는 키를 결합해 사용하는 경우는 막을 수 없다. 둘 이상의 사용자가 서로 다른 인가자로부터의 속성을 결합할 수 없게 하기 위해 모든 사용자들은 전역 ID 를 갖도록 한다. 단일 인가자 방식에서 사용했던 난수 값 대신 사용자 별로 모두 다른 값을 갖는 전역 ID 를 비밀키 생성에 이용함으로써 서로 다른 사용자가 다른 인가자로부터 생성한 비밀키를 결합해서 사용하는 공격을 막을 수 있다.

각 사용자의 전역 ID 를 입력으로 받아서 초기 키(Setup key)를 생성하기 위해서는 전체 사용자를 관리하는 중앙 인가자(CA)가 필요하다. CA 는 신뢰할 수 있는 기관으로써 시스템의 모든 메시지를 복호화 할 수 있다. (그림 5)의 계층 구조의 가장 위에 있는 TA 가 이 CA 의 역할을 맡게 된다.



(그림 5) 다중 인가자 모델

<표 1> 제안 모델에서 사용하는 주요 함수

함수 이름	기능
Setup	- 시스템이 사용할 키 구조의 깊이, 보안 인자를 입력으로 시스템 공개키와 마스터 키를 생성 - 다중 인가 환경에서 서로 다른 인가자로부터 분배한 키를 구분하기 위한 공개키/비밀키 쌍을 생성
Create DA	- 시스템 공개키, 마스터 키, 해당 인가자가 관리하는 속성 집합을 입력으로 받아 최상위 DA 의 마스터 키 생성
Central Key Generation	- TA 에서 실행되는 함수로, 시스템 마스터 키, 각 사용자의 고유한 전역 ID, 의사 난수 발생 함수(PRF)를 이용해 초기 키 생성해 각 DA 들에게 전달
Attribute Key Generation	- 각 DA 가 사용자에게 비밀 키를 분배하기 위해 사용하는 함수로, 해당 사용자가 속해 있는 DA 의 마스터 키, 사용자의 속성 집합, Central Key Generation 함수에서 생성한 초기키를 입력으로 이용 - 사용자에 따라 다른 비밀키를 부여하기 위한 전역 ID 값, Setup 함수에서 생성한 인가자의 비밀키 값을 이용해 각 DA 에서 사용자에게 비밀키를 분배
Encryption	- Setup 함수에서 생성했던 해당 인가자의 공개키를 이용해 데이터를 암호화 - 접근트리를 입력으로 받아서 데이터에 대한 접근 정책을 만족시키는 사용자만 복호화 할 수 있는 암호문 생성
Decryption	- 접근정책에 따라 암호화 된 메시지를 복호화 해서 파일을 암호화한 키를 알아내는 함수 - 사용자의 속성에 대한 비밀키와 Encryption 함수에서 생성된 암호문을 입력으로 받아 각 인가자별로 복호화 계산을 한 결과들을 결합해서 최종 복호화 키 생성 - 알아낸 비밀키로 메시지를 복호화

본 논문에서 제안하는 모델의 주요 함수는 Setup, Create DA, Central Key Generation, Attribute Key Generation, Encryption, Decryption 여섯 가지로, 계층적 구조에 다중 인가자 방식을 추가적으로 지원하기 위해 사용한다. 각각의 함수의 구체적인 기능은 <표 1>과 같다.

제안된 모델은 단일 인가자를 이용하는 HASBE 의 한계를 극복하기 위해 다중 인가자 방식을 이용하였다. 또한 다중 인가자 모델에 좀더 복잡한 접근 정책을 표현할 수 있는 접근 트리를 이용하여 유연한 접근제어가 가능하도록 함과 동시에 계층 구조를 이용해 관리의 효율성을 높였다.

4. 결론

클라우드 컴퓨팅이 학계와 산업계에서 큰 주목을 받고 있고 실제로 많은 업체들이 이미 서비스를 제공하고 있다. 하지만 대부분의 사용자들은 아직 클라우드 시스템의 보안에 대한 우려를 가지고 있다. 이러한 우려들의 근본적인 원인은 클라우드 서버상의 내 데이터에 누가 접근하는지 확실히 알지 못한다는 점이다. 따라서 데이터에 대한 접근제어를 효율적이고 안전하게 하는가에 클라우드 시스템 보안에 대한 신뢰도가 달려있다고 볼 수 있다.

본 논문에서는 동적인 클라우드 환경에서의 효과적인 접근제어를 위한 시스템 모델을 제안하였다. 사용자에 따른 유연성 있는 접근제어를 위해 데이터 암호화 방식으로는 속성 기반 암호화 방식을 확장한 ASBE 를 이용하였다. 인가자와 사용자들의 관계를 계층적인 구조로 조직하는 HASBE 를 이용해 클라우드 서비스를 이용하는 수많은 사용자와 기관들을 관리하는 데 있어 효율성을 높였다. 또한, 한 사용자와 데이터에 관련된 속성들은 모두 하나의 인가자가 관리해야 한다는 한계를 극복하기 위해서 다수의 인가자가 각각 필요한 속성과 키를 관리하도록 하는 다중 인가 방식을 더해 좀 더 실질적인 활용이 가능한 모델을 구성하였다.

향후 제안 모델의 TA 및 각 구성 요소의 보안 강화 방법에 대한 연구와 시스템이 사용하는 여섯 가지 주요 함수에 대해 수식을 활용해서 분석을 진행 할 예정이다.

Acknowledgment

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신)의 일환으로 수행하였음. [10041244, 스마트 TV 2.0 소프트웨어 플랫폼]

참고문헌

[1] http://research.microsoft.com/pubs/80240/dwork-tcc09.pdf, Mar 2013

[2] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, “Role-Based Access Control Models”, IEEE Computer, vol. 29, no. 2, pp. 38-47, Feb 1996

[3] A. Sahai, B. Wataers, “Fuzzy Identity-Based Encryption”, Eurocrypt Advances in Cryptology, LNCS, vol. 3494, pp. 457-473, May 2005

[4] Z. Wan, J. Liu, R.H. Deng, “HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing”, IEEE Information Forensics and Security, vol. 7, no. 2, pp. 743-754, Apr 2012

[5] V. Goyal, O. Pandey, A. Sahai, B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data”, ACM CCS, pp. 89-98, Oct 2006

[6] J. Bethencourt, A. Sahai, B. Waters, “Ciphertext-Policy Attribute-Based Encryption”, IEEE Security and Privacy, pp. 321-334, May 2007

[7] R. Bobba, H. Khurana, M. Prabhakaran, “Attribute-Sets: A Practically Motivated Enhancement to Attribute-Based Encryption”, ESORICS Computer Security, LNCS, vol. 5789, pp. 587-604, Sept 2009

[8] M. Chase, “Multi-authority Attribute Based Encryption”, TCC, LNCS, vol. 4392, pp. 515-534, Feb 2007