클라우드 컴퓨팅 환경에서 효율적인 분산 저장 서버 관리를 위한 그룹키 확립 프로토콜+

김수현*, 홍인식**, 이임영*
순천향대학교 컴퓨터소프트웨어공학과
e-mail:[kimsh, ishong, imylee]@sch.ac.kr

Group Key Establishment Protocol for Efficient Distributed Storage Server Management in Cloud Computing

Su-Hyun Kim, In-Sik Hong, Im-Yeong Lee

Department of Computer Software Engineering Soonchunhyang University*

Department of Computer Engineering Soonchunhyang University**

요 약

클라우드 컴퓨팅환경에서 사용되는 분산 파일 시스템은 데이터를 저장하는 분산 저장 서버와 각 데이터의 메타데이터를 저장하는 마스터 서버로 구성되어 있다. 마스터 서버와 분산 저장 서버는 수시로 서버의 상태나 메타데이터의 정보를 교환하지만, 통신 시 암호화가 전혀 고려되지 않아, 제 3자에 의한 도청이나 위변조시 사용자의 데이터에 대한 가용성을 보장받지 못할 수 있다. 이에 대한 방지 대책으로 통신 과정을 암호화함으로써 해결할 수 있지만, 무한히 확장 가능한 분산 저장서버에 대해 단일 마스터 서버와의 통신과정을 암호화하게 된다면 수많은 키에 대한 관리 대책을 필요로 하게 된다. 하지만 이 때, 분산저장서버를 하나의 그룹으로 묶어 그룹키를 사용하여 통신과정을 암호화한다면 보다 효율적으로 해결할 수 있다. 따라서 본 논문에서는 분산 저장 서버와 마스터 서버 간 안전하고 효율적인 암호화 통신을 위한 그룹키 확립 프로토콜을 제안하였다.

1. 서론

최근 국내외로 클라우드 컴퓨팅에 관한 관심이 높아지며 많은 연구가 진행되고 있다. 많은 기업들이 IT기술의 성장을 발판으로 다양한 분야로 확장 가능하고, 컴퓨팅 파워의 효율적인 사용이 가능한 클라우드 컴퓨팅에 관심을 가지고 있다. 하지만 클라우드 컴퓨팅의 도입을 가장 꺼려하는 이유 중 하나가 바로 보안적 문제점이 존재한다는 것이다. 사용자들의 데이터를 보호하기 위해 클라우드 서비스 업체들은 다양한 방식을 통해 안전성을 높이고 있지만, 사용자들은 자신의 민감한 데이터가 어디에 저장되어 있는지 혹은 기업들에 의해 어떻게 관리되는지에 대한 불안감을 떨칠 수 가 없게 된다. 또한 분산 저장된 데이터가 악의적인 사용자에게는 통신로 상에 노출되는 데이터보다 언제든지 접근이 가능한 서버에 저장되어 있는 데이터가 더 쉬운 목표물이 될 수 있다. 이러한 이유로 대부분의 클라우드 컴퓨팅 시스템에서는 사용자의 데이터를 암호화하여 저장한다. 기존 클라우드 컴퓨팅에서 사용되는 분산 파일 시스템은 데이터를 저장하는 분산 저장 서버와 각 데이터의 메타데이터를 저장하는 마스터 서버로 구성되어 있다[1]. 하지만 각 분산 저장 서버와 마스터 서버 간 통신 시 암호화가 전혀 고려되지 않아, 제 3자에 의한 도청이나 위변조시 사용자의 데이터에 대한 가용성을 보장받지 못할 수 있다. 이에 대한 방지 대책으로 통신 과정을 암호화함으로써 해결할 수 있지만, 무한히 확장 가능한 분산 저장서버에 대해 단일 마스터 서버와의 통신과정을 암호화하게 된다면 수많은 키에 대한 관리 대책을 필요로 하게 된다. 하지만 이 때, 분산저장서버를 하나의 그룹으로 묶어 그룹키를 사용하여 통신과정을 암호화한다면 보다 효율적으로 해결할 수 있다. 따라서 본 논문에서는 분산 저장 서버와 마스터 서버 간 안전하고 효율적인 암호화 통신을 위한 그룹키 확립 프로토콜을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 기법의 이해를 돕기 위한 관련 기술들을 소개하고, 3장에서는 클라우드 컴퓨팅 환경이 갖추어야 할 기본적인 보안 요구사항에 대하여 알아보고, 4장에서는 제안 방식에 대하여 설명한다. 5장에서는 제안 방식의 안전성을

+ 이 논문은 2012년도 한국산업기술재단의 재원으로 지역산업기술개발사업의 지원을 받아 수행된 연구임 (과제번호:A001100086)

분석한고, 마지막으로 6장에서는 결론 및 향후 연구 방향으로 마치도록 한다.

2. 관련연구

본 장에서는 본 논문에서 제안하는 기법의 이해를 돕기 위한 관련 기술과 기존 제안방식들을 소개한다.

2.1 Google File System(GFS)

구글에서 사용하는 구글 파일 시스템은 대용량의 데이터에 적합하도록 개발되었으며 핵심 데이터 저장 및 검색 엔진에 최적화 되어 있다. GFS는 대용량의 데이터를 저장하기 위해 수많은 저비용 스토리지 서버를 사용하여 분산 저장하게 된다. GFS의 구조는 마스터 서버와 청크 서버로 구성이 되며, 데이터는 다수의 청크 서버에 64메가바이트 단위로 나눠진다. 마스터 서버에는 각 청크 서버의 생성 시점에 고유의 64비트 레이블을 할당하고, 논리적 매핑을 이용해 청크 서버와 연결을 유지한다. 하지만 다수의 청크 서버를 사용함으로써 빈번하게 발생할 수 있는 장애를 대처할 수 있는 방안을 필요로 하는 단점이 존재한다.

2.2 Apache Hadoop Distributed File System(HDFS)

HDFS는 기성 하드웨어에서 실행 가능하도록 제작된 파일 시스템으로 기존의 분산 파일 시스템과 많은 유사점을 가지고 있다. 하지만 많은 차이점도 보이는데 높은 장애복구 기능과 저가의 하드웨어에 적용이 가능하도록 설계되었다. HDFS는 현재 Amazon, IBM, Yahoo 등과 같은 글로벌 IT 기업들의 클라우드 컴퓨팅 플랫폼의 기반이 되는 분산 파일 시스템으로 가장 널리 활용이 되고 있다.

HDFS의 설계와 구현을 위해 도출된 사항들을 살펴보면 GFS와 대부분 동일하며 플랫폼 간의 손쉬운 이식성을 보장하기 위해 자바를 사용하여 구현되었다는 점이 크게 다르다. 높은 이식성을 지닌 자바 언어의 사용은 자바를 지원하는 다양한 서버들에서 HDFS가 구동할 수 있다는 장점을 지닌다[2].

3. 보안요구사항

클라우드 컴퓨팅의 핵심 메커니즘 중 하나는 바로 키 관리를 통한 데이터의 암호화이다. 키 저장 서버의 장애 발생 시 다수 사용자의 데이터 접근이 불가능해지므로 키 관리 방안에 대한 연구가 필요하다. 이러한 다양한 장애를 사전에 방지하기 위해 다음과 같은 보안요구사항을 필요로 하게 된다[3].

· 안전한 키 저장소 : 키 저장소(key stores)는 다른 민감한 데이터와 마찬가지로, 반드시 자체적으로 보호해야 한다. 키 저장소는 저장소 내에서, 전송 중에, 그리고 백업 중에 반드시 보호되어야 한다. 부적절한 키 저장소는 모든 암호화된 데이터를 손상시킬 수 있다.

· 키 저장소로의 접근 : 키 저장소로의 접근은 특별히 개인키를 필요로 하는 엔티티로 제한해야 한다. 또한 키 저장소를 관리하는 정책들은 접근 통제를 돕는 역할을 분리해서 사용해야 하는데 키를 공급하는 엔티티와 키를 저장하는 엔티티는 달라야 한다.

· 키 백업 및 복구 : 키의 손실은 필연적으로 키를 보호하는 데이터의 손실을 뜻한다. 데이터를 파괴하는 효과적인 방법이지만, 업무상 중요한 데이터를 보호하는 키의 돌발적인 분실은 비즈니스에 엄청난 손실을 미친다. 그래서 안전한 백업 및 복구 솔루션은 반드시 구현되어야 한다.

· 전방향 안전성 : 그룹을 탈퇴한 멤버를 포함하여 이전 그룹키를 알고 있는 공격자는 새 그룹키를 알 수 없어야 한다.

· 후방향 안전성 : 그룹을 새롭게 가입한 멤버를 포함하여 현재 그룹키를 알고 있는 공격자는 이전 그룹키를 알 수 없어야 한다.

4. 제안방식

기존 클라우드 컴퓨팅에서 사용되는 분산 파일 시스템은 데이터를 저장하는 분산 저장 서버와 각 데이터의 메타데이터를 저장하는 마스터 서버로 구성되어 있다. 하지만 각 분산 저장 서버와 마스터 서버 간 통신 시 암호화가 전혀 고려되지 않아, 제 3자에 의한 도청이나 위변조시 사용자의 데이터에 대한 가용성을 보장받지 못할 수 있다.

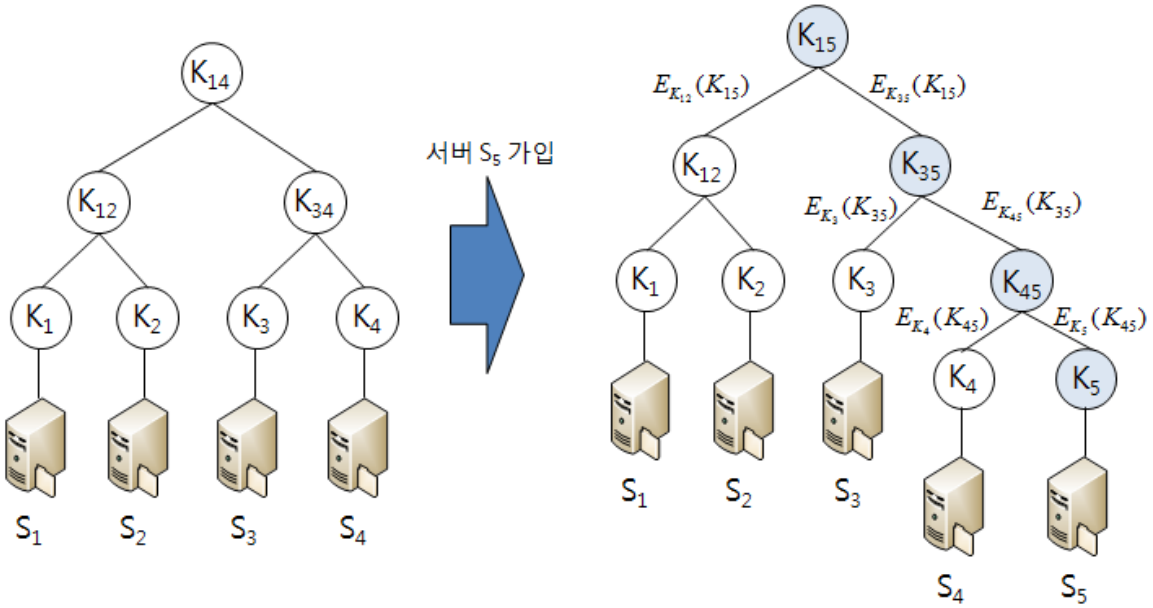
이에 대한 방지 대책으로 통신 과정을 암호화함으로써 해결할 수 있지만, 무한히 확장 가능한 분산 저장서버에 대해 단일 마스터 서버와의 통신과정을 암호화하게 된다면 수많은 키에 대한 관리 대책을 필요로 하게 된다. 하지만 이 때, 분산저장서버를 하나의 그룹으로 묶어 그룹키를 사용하여 통신과정을 암호화한다면 보다 효율적으로 해결할 수 있다.

따라서, 본 논문에서는 분산 저장 서버와 마스터 서버 간 안전하고 효율적인 암호화 통신을 위한 그룹키 확립 프로토콜을 제안하였다. 전체적인 구조는 Wong 등에 의한 LKH 방식을 기반으로 구성되었다[4].

4.1 초기 설정

각 분산 저장 서버에 대한 가입 및 탈퇴에 대해 효율적으로 관리하기 위해 논리적 트리 구조를 이용한다. 이 트리에서 각 노드마다 하나의 키가 유지되며, 트리의 루트 노드에 있는 키가 그룹키가 된다. 각 서버마다 하나의 단말 노드와 연관되며, 서버는 그 노드와 그 노드의 조상 노드에 있는 모든 키를 받아야 한다. 예를 들어, 서버 S_1은 K_1, K_{12}, K_{14}, K_{18}을 유지해야 한다(그림 1).

따라서 초기 설정 프로토콜은 초기 가입된 서버들을 이용하여 논리적 키 계층구조를 형성하고, 마스터 서버는 각 서버에게 각 서버가 유지해야 하는 키를 각 서버와 공



(그림 1) 가입 프로토콜

유된 비밀키로 전달하여 준다.

4.2 가입 프로토콜

클라우드 컴퓨팅 환경에서 분산 저장 서버는 유연한 확장을 제공하므로, 꾸준히 증가하게 된다. 수시로 증가하는 분산 저장 서버에 대해 그룹키를 효율적으로 관리하기 위해 다음과 같이 가입 프로토콜이 구성된다.

가입 프로토콜은 (그림 2)와 같이 서버를 트리에 추가하고, 추가에 따라 변경되는 키들을 다른 서버에게 전달한다. 서버가 추가될 때 고려해야 하는 것은 후방향 안전성이다. 즉, 이 가입 서버는 기존에 사용된 키들을 알 수 없어야 한다. 따라서 이 서버에게 전달해야 하는 키들을 모두 바꾸어야 한다.

S_5가 가입되는 경우 현재 트리에 추가한다. (그림 2)에서는 S_4의 노드를 한 단계 아래로 내리고, 그 노드와 형제 노드가 되도록 추가하였다. 따라서 기존 키 중에서 K_{34}와 기존 그룹키였던 K_{14}를 변경해야 한다. 노드의 키가 변경되면 기존 서버들에게도 그 변경된 값을 전달해야 한다.

4.3 탈퇴 프로토콜

클라우드 컴퓨팅 환경에서 가입 프로토콜에 비해 발생하는 횟수는 적지만, 분산 저장 서버의 고장 및 오류발견에 의해 정상적인 작동이 불가능하다고 판단되는 경우 다음과 같은 탈퇴 프로토콜이 구성된다.

탈퇴 프로토콜에서 보장해야 하는 것은 전방향 안전성이다. 즉, 탈퇴한 사용자는 더 이상 새로운 그룹키를 얻을 수 없어야 한다. 이를 위해 탈퇴할 사용자가 기존에 알고 있던 키들을 모두 바꾸어야 한다. (그림 3)에서 서버 S_3가 탈퇴하는 경우를 보면, 그 형제 노드를 한 단계 위로 올리고 그것의 모든 조상 노드의 키를 바꾸어야 한다. 즉, K_{14}와 K_{18}이 변경된다.

5. 제안방식 분석

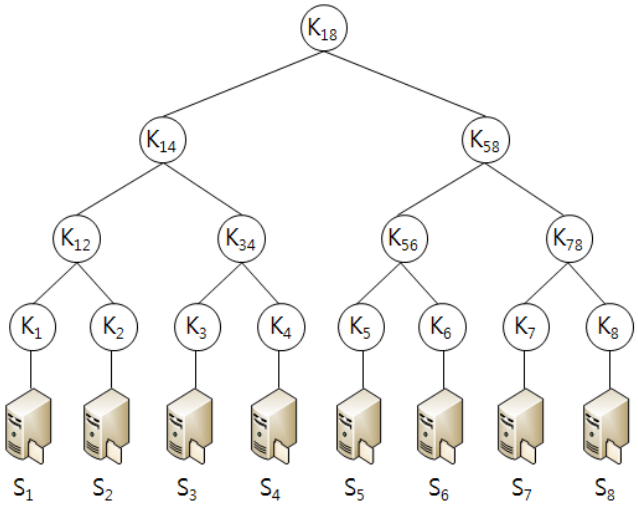
본 논문에서 제안한 기법은 공격자가 Admin을 제외한 객체를 공격하여도 얻고자 하는 데이터를 얻을 수 없는 안전한 시스템을 보장한다.

5.1 전방향 안전성

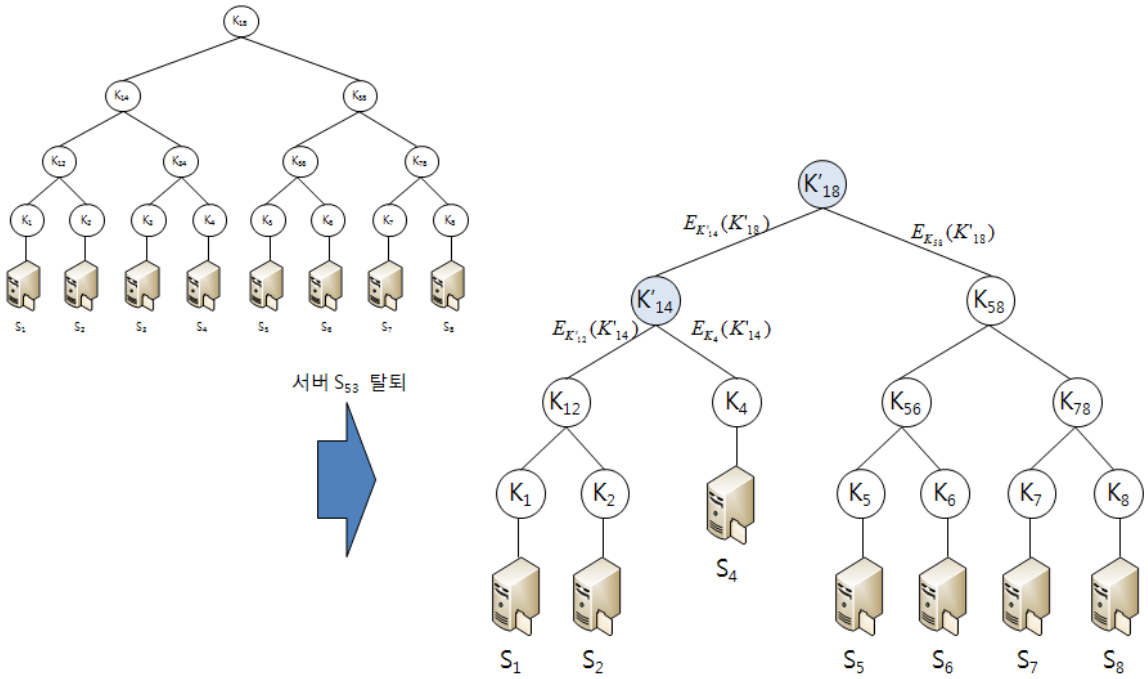
그룹을 탈퇴한 서버를 포함하여 이전 그룹키를 알고 있는 공격자는 새 그룹키를 알 수 없도록 하기 위해 그룹 탈퇴 시 마스터 서버는 각 분산 저장 서버에게 논리적 트리구조를 이용하여 그룹키를 갱신함으로써 전방향 안전성을 제공한다.

5.2 후방향 안전성

그룹을 가입한 서버를 포함하여 현재 그룹키를 알고 있는 공격자는 이전 그룹키를 알 수 없도록 하기 위해 새



(그림 2) 논리적 키 계층 구조



(그림 3) 탈퇴 프로토콜

로운 서버 가입 시 마스터 서버는 논리적 트리구조를 이용하여 그룹키를 갱신함으로써 후방향 안전성을 제공한다.

5.3 효율성

클라우드 컴퓨팅을 구성하는 기존의 분산 파일 시스템의 경우, 중앙 집중 방식의 형태로 구성된다. 이 방식은 마스터 서버와 통신 시 암호화가 전혀 고려되지 않았을 뿐만 아니라, 암호화 통신을 한다고 가정하더라도 다수의 분산 저장 서버와 마스터 서버간의 공유된 그룹키를 관리하기에 비효율적인 구성이라 할 수 있다. 중앙 집중 방식은 그룹키를 갱신할 때마다 마스터 서버와 각 분산 저장 서버와 공유된 비밀키로 암호화하여 전달하게 되는데, n이 그룹 멤버의 크기일 때, 설정, 가입, 탈퇴 프로토콜은 모두 O(n)의 유니캐스트 통신비용이 소요된다. 이 방식은 각 프로토콜이 가입된 그룹 멤버의 수에 비례하므로, 클라우드 컴퓨팅의 장점이라 할 수 있는 유연한 확장성을 제공하기에 부적합하다 할 수 있다.

하지만 본 방식에서는 논리적인 트리구조를 사용함으로써 그룹키 갱신 시 일부 구성원에게만 멀티캐스트를 하기 때문에 트리 높이의 2배만큼의 메시지가 소요된다. 즉, $O(\log_2 n)$만큼의 통신비용이 소요됨으로써 기존방식에 비해 보다 효율적이라 할 수 있다.

6. 결론 및 향후 연구 방향

본 논문에서는 클라우드 컴퓨팅을 구성하는 분산 파일 처리 시스템에서 마스터 서버와 분산 저장 서버 간의 통신 시 암호화에 필요한 그룹키를 관리하는 방식에 대하여 제안하였다. 기존 클라우드 컴퓨팅의 분산 저장 서버에 대한 그룹키 관리방식을 보다 효율적으로 구성하였고, 마스터 서버와 통신 시 데이터에 대한 기밀성을 유지하였다. 향후 논리적 트리의 높이에 독립적인 그룹키 관리방식과 경량화된 논리적 트리를 구성하는 방안에 대해 연구가 필요할 것으로 사료된다.

참고문헌

[1] 민영수, 김홍연, 김영균, “클라우드 컴퓨팅을 위한 분산 파일 시스템 기술”, 정보과학회지 제27권 제5호, 2009

[2] Who uses Hadoop, http://wiki.apache.org/hadoop/

[3] Ludwig Seitz, Jean-Mar c Pierson, Lionel Brunie, "Key management for encrypted storage in distributed systems," Second IEEE International Security in Storage Workshop, 2003.

[4] Chung K. Wong, Mohamed Gouda, and Simon S. Lam, "Secure Group Communications using Key Graphs," IEEE/ACM Trans. on Networking, Vol. 8, No. 1, pp. 16 - 30, 2000.