모바일 컴퓨팅 환경에 적합한 데이터 관리 기법1)

박수완*, 김정녀*, 이덕규**
*한국전자통신연구원
**서원대학교 정보보안학과
e-mail:parksw10@etri.re.kr, jnkim@etri.re.kr, deokgyulee@gmail.com

Data Management Scheme for Mobile Computing

Su-Wan Park*, Jeong-Nyeo Kim*, Deok Gyu Lee**
*Electronics & Telecommunications Research Institute
**Dept of Information Security, Seowon University

요 약

모바일 클라우드 컴퓨팅 환경에서 다양한 데이터 서비스가 가능해지면서 데이터의 분산관리가 주요 이슈로 떠오르고 있다. 개인이 아닌 그룹 내에서 공동으로 사용하는 모바일 퍼블릭 클라우드의 경우 그룹 내 사용자 모두 공동의 그룹키를 사용하여 데이터를 암호화 하게 된다. 하지만 그룹 내의 기존 사용자가 그룹을 탈퇴할 경우 그 사용자가 접근 가능했던 데이터가 탈퇴한 그룹원에 의해 노출되게 된다. 이를 방지하기 위해 탈퇴한 멤버에 의한 데이터 유출을 막기 위해서 새로운 그룹키를 갱신하여 모든 데이터를 다시 암호화 해야 한다. 하지만 이 과정에서 대용량 데이터의 암복호화 과정에서 막대한 오버헤드가 발생하게 된다. 이러한 문제점을 해결하기 위해 본 논문에서는 이러한 그룹 멤버의 가입과 탈퇴에 독립적인 그룹키 관리 방식을 제안한다.

1. 서론

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. 개인이 아닌 그룹 내에서 공동으로 사용하는 시스템의 경우 그룹 내 사용자 모두 데이터를 서비스 받기 위해 공동의 그룹키를 사용하여 데이터를 암호화 하게 된다. 추후에 그룹 내의 기존 사용자가 그룹을 탈퇴할 경우 그 사용자가 접근 가능했던 데이터가 탈퇴한 그룹원에 의해 노출되게 된다. 이를 방지하기 위해 탈퇴한 멤버에 의한 데이터 유출을 막기 위해서 새로운 그룹키를 갱신하여 모든 데이터를 다시 암호화 해야 한다. 하지만 이 과정에서 대용량 데이터의 암복호화 과정에서 막대한 오버헤드가 발생하게 된다. 이러한 문제점을 해결하기 위해 본 논문에서는 이러한 그룹 멤버의 가입과 탈퇴에 독립적인 그룹키 관리 방식을 제안한다.

1) 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음. [2014(10043959), 모바일 단말의 비인가 접근 차단 및 안전한 운영환경 보장을 위한 EAL 4급 군사용 융합 보안 솔루션 개발]

2. 에이전트 개발도구의 요구사항

・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.

3. 제안방식

기존 연구[1]에서는 그룹 구성원의 탈퇴 시 탈퇴한 구성원이 보유한 키로 접근 가능했던 모든 데이터에 대해 새로운

그룹키를 갱신하여 새로운 그룹키로 다시 암호를 해야 하는 문제점이 존재한다. 이런 문제점을 해결하기 위해 본 논문에서는 그룹 구성원에게 그룹키를 노출시키지 않고, 데이터 접근이 가능하며, 그룹 구성원의 탈퇴 시 마다 발생하는 오버헤드를 줄일 수 있는 기법을 제안하였다. 또한 분산 서버로부터 수집하는 데이터에 대해 통신로 상에서 안전하게 암호화 하며, 분산서버의 서명을 동시에 제공하는 사인크립션 기법을 적용하였다.

3.1 분산 서버와 마스터 서버와의 통신

사용자의 데이터는 마스터 서버를 통하여 분산서버에 여러 조각으로 나뉘어 저장된다. 사용자로부터 데이터 요청이 들어왔을 경우 분산되어 있는 데이터를 수집하는 과정이 필요하다. 이 때, 분산 서버는 마스터 서버와 물리적, 논리적으로 분리되어 있기 때문에, 각각의 분산서버에 대한 인증과 조각 데이터 전송 시 통신로 상에서의 기밀성을 제공해야 한다. 기존의 공개키 암호 시스템을 사용할 경우, 비용적, 연산적 측면에서 매우 비효율적이므로 본 논문에서는 사인크립션기법을 적용하여 보다 효율적인 통신이 이루어지도록 제안하였다.

Step 1: 분산서버에서 실행되는 사인크립션(Signcryption)

덤 선택x
$-\ w = y_s^x \bmod p$
$-\ k =\ (w)$
$-\ r = H(m, Server_{\infty o}, w)$
$-\ s = x/(r + x_{ss}) \bmod q$
$-\ c = E\ (m)$
$(m = E_K\ (data))$
$-\ (c, r, s)$ 전송

Step 2: 마스터서버에서 실행되는 언사인크립션 (Unsigncryption)

$-\ w = (y_{ss} \cdot g^r)^{s \cdot x_s} \bmod p$
$-\ k = G(w)$
$-\ m = D_k(c)$
$-\ \text{if}\ (r? = H(m, Server_{\infty o}, w)$

4. 제안방식 분석

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

4.1 후방향 안전성

그룹을 구성하여 통신을 하는 경우, 그룹의 멤버가 탈퇴 시 새로운 그룹키를 확립하여 데이터를 암호화 하여야 한다. 새로운 그룹키를 사용하는 경우, 저장되어 있는 데이터 또한 모두 다시 암호화과정을 거쳐야 하기 때문에 엄청난 오버헤드를 가져오게 된다. 하지만 본 제안방식은 그룹 구성원에게 데이터 암호화 키, 즉 그룹키 자체를 노출시키지 않기 때문에, 탈퇴나 가입이 이루어지더라도 그룹키를 새롭게 갱신할 필요가 없다.

4.2 인증 및 기밀성

사용자의 데이터는 마스터 서버를 통하여 분산서버에 여러 조각으로 나뉘어 저장된다. 사용자로부터 데이터 요청이 들어왔을 경우 분산되어 있는 데이터를 수집하는 과정이 필요하다. 이 때, 분산 서버는 마스터 서버와 물리적, 논리적으로 분리되어 있기 때문에, 각각의 분산서버에 대한 인증과 조각 데이터 전송 시 통신로 상에서의 기밀성을 제공해야 한다. 기존의 공개키 암호 시스템을 사용할 경우, 비용적, 연산적 측면에서 매우 비효율적이다.

5. 결론 및 향후 연구 방향

본 논문에서는 그룹구성원에게 노출되지 않는 그룹키 관리 기법과 사인크립션을 이용한 분산 서버 관리 기법을 제안하였다. 기존의 그룹키 관리방식에 비해 효율적으로 키 개수를 줄였고, 사인크립션을 이용하여 서버 간 인증과 데이터의 기밀성을 유지하였다. 향후 사인크립션을 이용하여 사용자에 대한 인증과 그룹키에 대한 기밀성을 유지할 수 있는 연구가 가능할 것으로 생각된다.

참고문헌

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

[2] A. Shamir. "How to Share a Secret," communication of the ACM, Vol. 22, No. 11, pp.612-613, 1979.

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