클라우드 스토리지 서비스에서의 TPM 기반의 안전한 데이터 접근 방법

김윤구 ○ 신재복, 박찬익
포항공과대학교 컴퓨터공학과
pi3wi2@postech.ac.kr, zstormx@postech.ac.kr, cipark@postech.ac.kr

Secure data accessing Method based on TPM in cloud storage service

Yungu Kim ○ Jae-Bok Shin, Chan-Ik Park
CSE, POSTECH

요 약

사용자 기기가 다양해지고, 기기 간 데이터 동기화 필요성이 증대되면서, 클라우드 스토리지 서비스에 대한 수요가 증가되고 있다. 이러한 클라우드 스토리지 서비스는 클라우드 서버의 보안 취약점을 공격하거나, 악의적인 목적을 가진 관리자에 의한 데이터 유출 위협이 항상 내포되어 있다. 따라서, 본 논문에서는 TPM을 기반으로 한 클라이언트에서의 데이터 암호화를 통해 이러한 문제점을 해결하고, 사용자의 다양한 기기들간의 데이터 이용을 안전하고 편리하게 할 수 있는 클라우드 스토리지 시스템을 제시한다.

1. 서론

무선 네트워크 인프라가 발달하고 모바일 기기의 사용량이 늘어나, 한 사용자가 여러 개의 네트워크 접속 기기를 사용하게 되었다. 이에 따라 각 기기들간의 데이터 동기화의 필요성이 발생하고, 이는 클라우드 서비스에 대한 수요의 증가로 이어지고 있다.

클라우드 서비스는 기본적으로 네트워크에 연결된 것을 전제로 하며, 사용자의 정보가 원격지 서버에 저장이 된다. 이 때문에 사용자의 정보가 유출되는 보안 문제가 발생할 수 있으며, 특히 클라우드 스토리지 서비스에는 다양한 데이터가 저장됨에 따라, 데이터 유출 문제가 발생할 경우 심각한 피해가 벌어질 수 있다.

클라우드 서비스에서 데이터의 유출 가능성은 항상 존재한다. 클라우드 서비스를 구성하는 Virtual Machine Monitor나 관리자 도메인의 경우, 수백만 라인의 코드를 가지고 있기 때문에, 취약성을 가진 코드를 내재할 가능성이 존재한다. 2010년 12월에 발표된 Common Vulnerabilities and Exposures에 따르면, 많이 사용되는 VMWare[10]와 Xen[11] 시스템의 경우, 각각 35개와 32개의 보안 취약성을 가지고 있다고 보고 되었다[9]. 이러한 클라우드 서버의 취약점을 공격한 크래킹에 의해 발생할 수도 있지만, 기본적으로 클라우드 서비스 관리자에게 아무런 장벽 없이 유출될 수 있다는 문제점을 가지고 있다. 이런 환경하에서 사용자는 안심하고 클라우드 스토리지 서비스를 사용할 수 없으며, 중요하고 민감한 정보를 클라우드 서비스에 저장하는 것을 꺼리게 된다.

또 다른 데이터 유출의 가능성은 사용자의 부주의에 있다. 사용자가 클라우드 서비스에 연결된 디바이스를 분실하거나, 사용자의 계정 정보가 유출될 경우, 클라우드 스토리지에 있는 데이터도 유출될 수 있다.

이러한 문제를 해결하는 방법으로 클라이언트에서 암호화를 하여 서버에 전송하는 방법이 있다. 하지만 여러 디바이스간에 암호화 키를 공통으로 사용하다 보면 디바이스간의 암호화 키 전송 과정이나, 디바이스 분실에 의해 암호화 키가 유출될 우려도 있다.

이러한 문제점들을 해결하기 위해, 하드웨어 기반으로 안전하게 암호화 키를 관리할 수 있는 스토리지 시스템을 제시한다.

2. 관련 연구

2.1 Dark Clouds on the Horizon[12]

클라우드 스토리지 서비스로 널리 사용되고 있는 Dropbox[4] 서비스에 대해 보안 취약점과 공격 시나리오 및 해결책을 제시하는 연구이다. 클라우드 스토리지 서비스의 보안 취약점을 이용해 다른 사람이 업로드한 파일의 내용을 다운로드 할 수 있거나, 사용자에게 할당된 저장용량을 초과하여 업로드를 하는 등의 문제가 보고되었다.

2.2 Trusted Platform Module

Trusted Platform Module (TPM)[1]은 Trusted Computing Group에서 제안한 하드웨어 보안 칩이다. TPM의 가장 큰 특징은 암호화에 사용되는 키나 인증에 필요한 인증서와 같은 중요한 데이터를 저장할 수 있는 저장공간을 제공하고, 이러한 데이터에 접근할 수 있는 인터페이스를 하드웨어적으로 제한하고 있다는 점이다. 또한, TPM은 Storage Root Key와 Endorsement Key를

기반으로 하여, key tree를 생성하고, TPM에서 사용되는 모든 키들이 종속성을 가지도록 구성하였다. TPM은 위와 같은 특성을 이용하여 암호화 키를 소프트웨어적인 관리 방식보다 보다 안전하게 사용할 수 있는 방법을 제시하였으며, 본 논문에서는 이 방법을 이용하여, 기기 간 안전한 키 관리를 구현하였다.

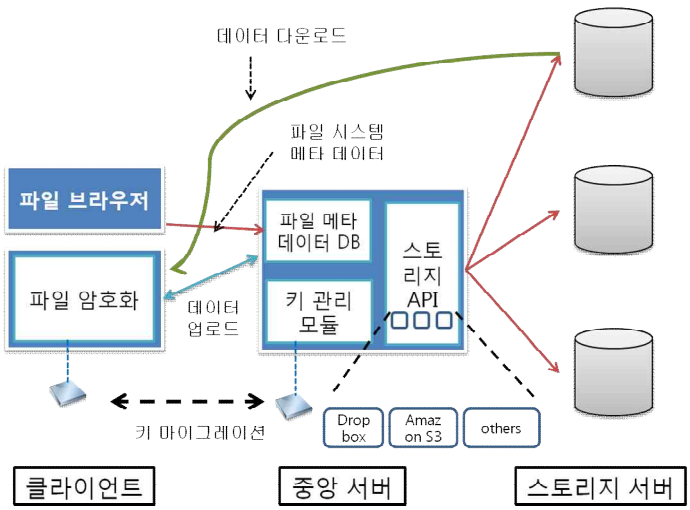

그림 1. 시스템 구조

3. 시스템 구조

시스템은 클라이언트와 중앙 서버, 스토리지 서버로 나누어진다. 클라이언트는 사용자에게 파일 목록을 보여주며, 사용자는 원하는 파일을 생성, 수정 및 업로드와 다운로드가 가능하다. 파일의 암호화 및 복호화는 클라이언트에서 수행한다. 중앙 서버는 사용자의 계정 정보 및 각 사용자 별 파일에 대한 메타데이터를 관리하며, 암호화 키 역시 관리한다. 스토리지 서버는 암호화된 형태의 파일을 저장하며 사용자의 필요에 따라 원하는 파일을 클라이언트에서 다운로드 할 수 있도록 해준다.

3.1 파일의 암호화 및 업로드

사용자는 클라이언트 프로그램에서 새로운 파일을 생성하거나 기존에 사용자의 디바이스에 존재하는 파일을 클라이언트 프로그램에 추가할 수 있다. 이 경우, 클라이언트에 새롭게 추가된 파일은 암호화를 거친 후 중앙 서버로 업로드가 되고, 중앙 서버에 사용자의 파일 메타 데이터가 업데이트 된다. 또한 중앙 서버에 업로드된 파일은 스토리지 서버로 전송된다. 하나의 파일이 하나의 스토리지 서버에 저장될 수도 있고, 여러 서버에 분산되어 저장될 수도 있다. 또한 파일의 크기가 스토리지 서버에서 지원하는 최대 파일 크기를 넘을 경우, 파일을 분리하여 저장한다.

파일의 암호화에는 TPM을 사용하며, 사용자의 디바이스가 하드웨어적으로 TPM을 지원하지 않을 경우 소프트웨어 방식으로 TPM 에뮬레이터[2]를 사용할 수 있다. 단, 소프트웨어적으로 안전한 TPM 에뮬레이터를 사용하기 위해서는 Arm TrustZone[3]과 같은 기술이 동반되어야 한다. TPM으로 암호화 키를 생성하거나 기존의 사용자가 썼던 암호화 키를 마이그레이션 하여 재사용할 수 있다. 암호화는 데이터가 서버에 전송되기 전에 클라이언트에서 이루어진다.

3.2 파일의 다운로드 및 복호화

사용자는 클라이언트 프로그램을 통해 자신의 스토리지에 저장된 파일의 목록을 조회할 수 있으며, 원하는 파일을 다운로드 할 수 있다. 사용자가 클라이언트 프로그램을 통해 다운로드를 요청하면, 중앙 서버에서 해당 파일이 저장된 스토리지 서버를 조회하고, 그 스토리지 서버를 클라이언트에서 접근할 수 있도록 요청을 보낸다. 그러면 스토리지 서버에서는 해당 데이터를 클라이언트에서 접근할 수 있는 URL을 중앙 서버로 보내고, 중앙 서버에서 다시 클라이언트로 알려준다. 클라이언트에서는 해당 URL을 통해 데이터를 다운로드하며, 다운로드가 완료되면 복호화를 거쳐서 사용자에게 파일을 보여준다.

파일의 복호화에도 클라이언트 TPM을 사용한다. TPM을 이용하여 기존의 사용자가 썼던 암호화 키를 현재 사용자가 접속한 디바이스로 마이그레이션을 하여, 그 키를 사용하여 복호화를 한다.

3.3 스토리지 서버

스토리지 서버는 기존에 서비스되고 있는 클라우드 스토리지 중, API를 통한 접근을 허용하는 스토리지 서비스를 사용한다. Dropbox, Amazon S3[5] 또는 Openstack Swift[6] 기반으로 구축된 공용 클라우드나 비공개 클라우드 서비스등을 사용할 수 있다.

3.4 암호화 키 마이그레이션

암호화 키 마이그레이션을 하는 이유는 한 명의 사용자가 데스크톱 PC, 노트북, 스마트폰 등 여러 개의 디바이스를 사용할 수 있기 때문이다. 이런 경우 각각의 디바이스가 별개의 암호화 키를 생성하여 사용하면 디바이스간의 데이터를 공유할 수 없다. 이 문제를 해결하기 위해, 클라이언트에서 중앙 서버로의 사용자 인증이 완료되면, TPM을 사용하여 서버에 저장된 암호화 키를 사용자의 현재 디바이스로 마이그레이션을 한다. 그러면 다른 디바이스에서 암호화하여 업로드한 파일이라도 현재 디바이스에서 복호화가 가능하다.

암호화 키 마이그레이션은 클라이언트 TPM과 중앙 서버 TPM 간의 TPM Key Migration 기술[7]을 통하여 이루어진다. 최초의 암호화 키는 클라이언트의 TPM에 의해 생성되며, 이 암호화 키는 중앙 서버 TPM의 공개키로 암호화 되어 중앙 서버의 TPM에 보내진다. 중앙 서버에서는 중앙 서버의 TPM의 공개키에 의해 암호화된 암호화 키를 중앙 서버의 DB에 저장한다.

중앙 서버에서 다른 클라이언트로 암호화 키를 보낼 때는, DB에 저장된 암호화된 키를 중앙 서버의 TPM에 입력한 후, 중앙 서버의 TPM의 비밀키를 이용하여 복호화를 하고, 키를 받을 클라이언트의 공개키로 암호화를 한 후, 클라이언트의 TPM으로 전송한다. 이 암호화 키는 클라이언트 TPM에서 비밀키를 통하여 복호화 한 후, TPM에 등록하여 사용할 수 있다.

서버에 저장된 암호화 키는 서버 TPM의 공개키로 암호화되어 저장되며, 서버 TPM의 비밀키는 서버 관리자도 알 수 없기 때문에 서버 관리자가 클라이언트의 암호화 키를 복호화할 수는 없다.

3.5 디바이스의 분실시 데이터 유출 최소화

사용자 인증을 받은 상태에서 디바이스를 분실한 경우, 해당 디바이스를 통한 데이터 유출을 막을 필요가 있다. 크게 두 가지 방법으로 접근할 수 있는데, 첫 번째는 서버에 등록된 키를 제거하는 것이다. 그렇게 하면 분실한 디바이스로 더 이상 키를 전송할 수 없기 때문에 암호화된 데이터를 복호화 하지 못한다. 단, 이 방법을 사용할 경우 다른 디바이스에서도 복호화를 할 수 없다.

두 번째 방법으로, 사용자가 자신이 사용할 디바이스의 고유 식별번호를 미리 등록을 해 놓고, 등록되지 않은 디바이스로부터의 접근을 허용하지 않는 것이다. 디바이스를 분실한 경우 해당 디바이스의 식별번호를 통해 등록한 것을 해제하면, 더 이상 해당 디바이스로 암호화 키가 전송되지 않게 되며, 데이터가 유출되지 않는다. 고유 식별번호로는 네트워크 카드의 맥 어드레스나, 스마트폰의 디바이스 아이디 등을 이용할 수 있다.

4. 평가

4.1 스토리지 서버의 취약점 보완

스토리지 서버에 보안 취약점을 통해 데이터가 외부로 유출된 경우, 데이터가 암호화되어 있기 때문에 데이터의 원본은 외부로 노출되지 않는다.

4.2 서버 관리자에 의한 데이터 유출 방지

스토리지 서버 관리자가 악의적인 의도로 서버에 저장된 데이터를 열람하거나 유출시키더라도 데이터는 암호화되어 있기 때문에 데이터의 원본은 노출되지 않는다. 암호화 키 또한 서버에 암호화되어 저장되며, 암호화 키의 복호화 역시 서버 관리자가 임의로 할 수 없기 때문에 안전하다.

4.3 디바이스 분실에 의한 데이터 유출 방지

사용자가 클라우드 스토리지에 인증을 완료한 디바이스를 분실하였을 경우, 디바이스 분실을 확인한 즉시 중앙 서버에 해당 디바이스의 인증을 해제하도록 요청함으로써 더 이상 해당 디바이스로 데이터를 받을 수 없도록 설정할 수 있다.

5. 결론

본 논문에서는 클라우드 스토리지 서비스를 사용함에 있어 문제가 될 수 있는 데이터 유출을 방지하기 위해, TPM을 이용한 클라이언트에서의 데이터 암호화 방법을 제시한다. 또한, 여러 디바이스를 사용함에도 불편함이 없도록 하였으며, 디바이스를 분실했을 때의 대응책도 제시한다.

6. Acknowledgements

본 연구는 지식경제부 및 정보통신산업진흥원의 “IT명품 인재 양성 사업”의 (C1515-1121-0003) 연구결과와 “대학 IT연구센터 지원사업”의 (NIPA-2012-H0301-12-3002) 연구결과로 수행되었음.

참고 문헌

[1] TPM. http://www.trustedcomputinggroup.org/resources/tpm_main_specification

[2] Software-based TPM Emulator, http://tpm-emulator.berlios.de/

[3] ARM TrustZone technology, http://www.arm.com/products/processors/technologies/trustzone.php

[4] Dropbox. http://www.dropbox.com

[5] Amazon Simple Storage Service. http://aws.amazon.com/s3/

[6] Swift is a highly available, distributed, eventually consistent object/blob store. http://swift.openstack.org/

[7] TCG Architecture Overview, version 1.4

[8] M. Vrable, S. Savage, and G. M. Voelkey, BlueSky: A Cloud-Backed File System for the Enterprise (FAST'12)

[9] Common vulnerabilities and exposures. http://cve.mitre.org/

[10] VMWare. http://www.vmware.com

[11] Xen. http://www.xen.org/

[12] M. Mulazzani, S. Schrittwieser, M. Leithner and M. Huber, “Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space,” in USENIX Security, 2011