클라우드 스토리지 서비스를 위한 ARM TrustZone 기반의 안전한 데이터 관리 기법

(A Secure Data Management Framework based on ARM TrustZone for Cloud Storage Services)

신 재 복, 김 윤 구, 박 우 람, 박 찬 익
포항공과대학교 컴퓨터공학과
(Jae-Bok Shin, Yun-Gu Kim, Woo-Ram Park, Chan-Ik Park)
(Department of Computer Science and Engineering, POSTECH)

Abstract : Today, anyone can have multiple mobile devices like smart phones and tablet PCs, and also can handle variety of user data at any time, in any place. For efficiently sharing or synchronizing the user data across multiple devices, many people are using cloud storage services. Although cloud storages provide flexibility and scalability in storing data, security issues should be handled especially when mobile devices try to access data stored in cloud storage. Currently, typical cloud storage services offer data encryption for security purpose but we think such method is not secure enough. Because we recognized that managing encryption keys by software and identifying users by simple ID and password are main defectives of current cloud storage services.

In this paper, we propose a secure data access method to cloud storage in mobile environment. Our framework supports hardware-based key management, attestation on the client software integrity, and secure key sharing across the multiple devices. To achieve these features, we implemented our prototype based on ARM TrustZone technology[10] and TPM emulator[1, 2] which is running on secure world of the TrustZone environment.

Keywords : Mobile Security, ARM TrustZone, TPM, Cloud Storage

I. 서 론

스마트 폰, 태블릿 PC와 같은 모바일 기기가 대중화 되면서 다양한 사용자 데이터가 모바일 기기에 저장되거나 처리되고 있다. 이와 함께 개인이 여러대의 모바일 기기를 소지하는 일이 흔해짐에 따라 효율적인 기기들간의 데이터 공유 또는 동기화를 위하여 클라우드 스토리지 서비스를 많이 사용한다. 하지만 사용자의 데이터가 원격지에 저장되는 클라우드 스토리지의 특성으로 인하여 사용자 데이터의 유출, 변경, 또는 유실 등의 보안 위협이 발생할 수 있다. 따라서 보안성을 중요시하는 개인 사용자나 기업 환경에서 클라우드 스토리지 서비스의 확산이 저하되는 원인이 된다.

사용자 데이터 유출은 클라우드 스토리지 서비스의 서버 측 또는 클라이언트 기기에서 발생할 수 있다. 서버 측에서는 클라우드 서비스를 구성하는 가상화 환경의 취약점[3]을 이용한 악의적인 공격이나[4, 5] 서비스 관리자와 같은 내부인에 의해서 사용자 데이터가 유출된다. 클라이언트에서는 키로거로 인한 사용자 인증정보 유출 그리고 악성 코드 등에 의해서 데이터가 유출된다.

따라서 현재 많이 사용되는 클라우드 스토리지

* 교신저자(Corresponding Author)
신재복, 김윤구, 박우람, 박찬익 : 포항공과대학교 컴퓨터공학과

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 "IT명품인재양성사업(C1515-1121-0003)" 및 "대학 IT연구센터 지원사업(NIPA-2012-H0301-12-3002)"의 연구결과로 수행되었음

서비스들은 사용자 데이터 보안을 위하여 암호화 기법을 사용하고 있다. 대표적인 예로 Amazon Simple Storage Service (Amazon S3[6])는 사용자의 선택에 따라 서버 기반 데이터 암호화 또는 클라이언트 기반 데이터 암호화를 지원하고 Dropbox[7]는 서버 기반 데이터 암호화 방식을 채택하였다. 이러한 방식을 사용하여 데이터 유출의 위협을 줄일 수 있지만 여전히 유출 위협이 존재하며 그 이유는 다음과 같다.

첫 번째는 암호화에 사용되는 데이터 키는 소프트웨어적으로 관리된다는 점이다. 따라서 악성 코드, 악의적인 관리자, 또는 크래킹에 의한 데이터 키의 유출로 암호화 된 데이터를 복호화 할 수 있다. 두 번째로 서버 기반 암호화의 경우 실제 암호화가 일어나기 전에는 데이터가 플레인텍스트로 존재한다는 점이다. 따라서 서버의 취약점을 이용한 공격에 의해 데이터가 유출될 수 있다. 세 번째는 사용자의 인증 과정이 간단한 ID와 패스워드 만으로 진행이 되기 때문에 키로거에 의한 인증정보 유출이 여전히 가능하다는 점이다.

따라서 본 논문에서는 모바일 클라우드 스토리지 서비스에서 데이터 유출을 최소화하는 안전한 데이터 관리 기법을 제안한다. 본 기법은 하드웨어 기반의 키 관리, 클라이언트 소프트웨어의 무결성 검증 그리고 여러 기기간의 안전한 키 공유 방법을 제공한다. 이를 위하여 ARM TrustZone 기술과 TrustZone 환경의 Secure World에서 동작하는 TPM 에뮬레이터를 이용해서 프로토타입을 구현하였다.

Ⅱ. 위협 모델

본 논문에서 제안하는 프레임워크를 적용한 모든 모바일 클라이언트들은 ARM TrustZone 기술이 적용되어 있다고 가정한다. ARM TrustZone 기술은 ARM 어플리케이션 프로세서(AP)에 하드웨어적으로 구현이 되어 있으며 Normal World, Secure World 두 개의 운영체제들이 독립적으로 동작하도록 지원한다. Secure World는 하드웨어적으로 보호되는 안전한 영역이다. 따라서 보안이 요구되는 작업을 Secure World에서 동작시켜 사용자 정보 보호, 암호화 키 관리 등을 안전하게 할 수 있다. 본 프레임워크는 ARM TrustZone 기술을 이용하여 하드웨어 기반의 키 관리 방법을 제공한다.

모바일 클라이언트의 Normal World에는 악성 프로그램이 상주할 수 있으므로 Normal World는 신뢰할 수 없는 환경을 가정한다. 또한 클라우드 스토리지 서비스 제공자도 앞서 서론에서 언급한 데이터 유출 위협으로 인하여 신뢰 할 수 없다. 본 프레임워크는 클라우드 스토리지 서비스에서 일어 날 수 있는 서버 측 그리고 클라이언트 측 데이터 유출에 초점을 두었다. 데이터 무결성과 서비스 사용성, 데이터 일관성 또한 중요한 보안 이슈이지만 여기에서는 다루지 않는다.

Ⅲ. 시스템 구조

1. 시스템 개괄

본 프레임워크(그림 1)는 ARM TrustZone 기술이 적용된 모바일 클라이언트, 중앙 서버, 그리고 클라우스 스토리지 제공자의 구성요소를 가진다. 사용자는 클라이언트의 Normal World에서 동작하는 파일 탐색기 응용프로그램을 통하여 사용자 인증, 로컬 및 클라우드 파일 탐색, 그리고 업로드와 다운로드, 삭제, 이동 등 파일 오퍼레이션을 수행한다. 한편으로 모바일 클라이언트 상에서 데이터 키 관리, 플랫폼 무결성 검증, 파일 암호화 및 복호화 기능을 제공하는 보안 프로세스들은 모두 Secure World에서 동작한다. 사용자가 파일 탐색기를 통하여 클라우드 파일에 접근하면 TrustZone API(TZAPI)가 호출되고 TrustZone 모니터에 의해 두 World가 전환(Context Switching)되면서 해당하는 요청에 관한 메시지가 전달된다. Secure World의 클라우드 스토리지 서비스 프로세스가 전달받은 메시지에 대한 처리를 하며 키 관리 기능들은 Secure World의 TPM Emulator를 이용하여 처리한다.

중앙 서버는 클라이언트의 플랫폼 무결성 검증을 수행하고 암호화된 사용자 데이터를 클라이언트로부터 전송받아 클라우드 스토리지 API를 사용하여 클라우드 스토리지에 저장하며 유저 ID, 파일명, 그리고 실제 저장된 클라우드 스토리지 위치로 이루어진 파일 메타데이터를 기록한다. 클라이언트에서 다운로드가 수행될 때 메타데이터를 검색하여 파일이 실제로 저장된 클라우드 스토리지 위치를 클라이언트로 전송함으로써 클라이언트로부터 해당 클라우드 스토리지에 직접 접근이 가능하도록 한다. 또한 Key Distributor를 통하여 서로 다른 사용자 또는 다른 기기들 간에 데이터 암호화 키 교환 프로토콜을 지원한다. 클라우드 API 계층은 여러개의 다른 상용 클라우드 스토리지를 선택적으로 사용할 수 있게 한다.

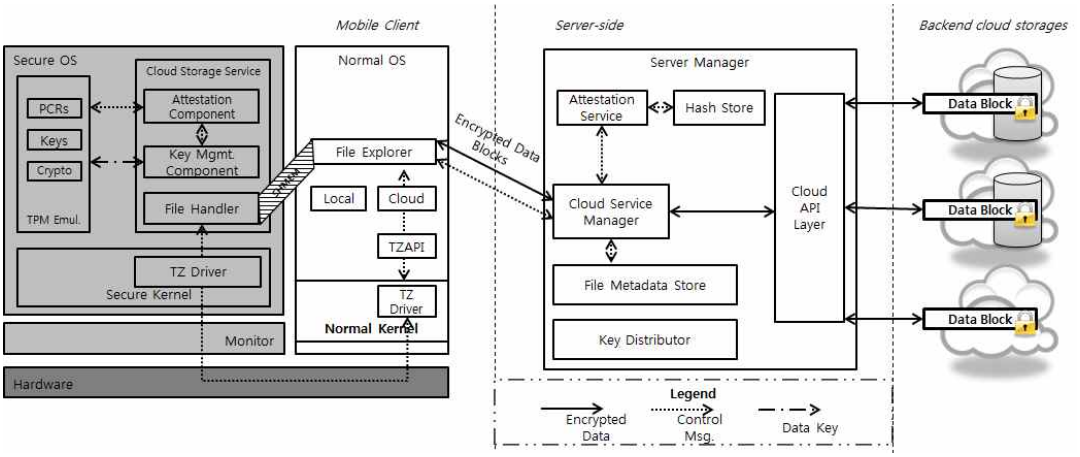

그림 1. 시스템 구조 (빗금친 부분은 공유메모리를 나타낸다.)
Fig 1. System Architecture

따라서 사용자 데이터는 클라우드 스토리지 선택 정책에 따라 임의적인 위치에 저장될 수 있다.

2. 사용자 인증 및 플랫폼 검증

본 프레임워크에서는 Core Root of Trust for Measurement(CRTM)을 Secure World로 가정한다. 모바일 클라이언트의 부팅과정에서 Secure World의 Attestation Component가 Normal 부트로더 및 OS 이미지의 무결성을 검증하고 검증 결과값을 TPM Emulator의 Platform Configuration Register(PCR)에 저장한다. Normal OS의 부팅 이후에 Intergrity Measurement Architecture(IMA)에 의해서 라이브러리 및 실행된 프로세스 무결성을 측정하고 TZAPI를 통하여 PCR에 extend한다.

사용자는 클라우드 파일 탐색기를 실행하여 자신의 인증정보(ID/password)로 로그인하게 되고 이 인증정보와 함께 TPM quote 명령으로 PCR에 저장된 무결성 검증 정보와 Measurement Log (ML) 값을 서버에 전송함으로써 원격 플랫폼 검증 프로토콜을 시작한다. 서버의 Attestation Service는 Hash Store에 저장된 값과 전송 받은 무결성 정보를 비교한 다음 현재 모바일 클라이언트가 안전한 상태에 있는지 검사하고 결과를 클라이언트로 전송한다. 인증이 실패하면 사용자는 자신의 클라우드 스토리지에 접근 할 수 없다.

3. 데이터 암호화 키 생성 및 관리

사용자 인증 및 플랫폼 검증이 성공적으로 끝나면 데이터 암호화에 사용할 암호화 키를 TPM에 불러오는 과정이 시작된다. 먼저 서버에 사용자의 키 생성 기록을 조회한 후, 키 생성, 키 로드, 또는 키 공유 프로토콜을 수행한다. 인증을 통과한 사용자가 처음 접속한 사용자이면 서버에 키 생성 정보가 없으므로 클라이언트에서 자신의 암호화 키를 생성하여야 한다. 따라서 Key Mgmt. Component가 TPM Emulator에게 키 생성 요청을 하고 서버에 생성 정보를 등록한다. 생성된 키는 추후 사용되지 않을 때 TPM Seal 명령을 통하여 sealing 되어 모바일 기기의 로컬 스토리지에 안전하게 보관된다. 반면에 이미 키 생성을 기록을 가진 사용자이고 해당 클라이언트가 키를 가지고 있는 경우에는 unsealing 과정을 통하여 TPM Emulator에 불러 온다. 마지막으로 키 생성 기록이 있고 다른 클라이언트가 키를 가지고 있는 경우 키 공유 프로토콜을 수행하여 안전하게 키를 현재 클라이언트로 받아 온다.

4. 파일 업로드 및 다운로드

사용자가 파일 탐색기를 통하여 로컬 파일을 클라우드 스토리지에 업로드할 때 먼저 Secure World와 Normal World의 공유메모리에 데이터를 불러들인다. TZAPI를 통하여 파일 암호화 요청을 하면 Secure World로 전환되고 File Handler에 의하여 공유 메모리상의 데이터를 암호화 키로 암호화 한 후 완료 메시지와 함께 Normal World로 전환한다. 파일 탐색기는 공유 메모리 상의 암호화된 데이터를 서버로 전송한다. 클라우드 스토리지의 데이터를 다운로드 할 때에는 서버로부터 전송받은 데이터의 실제 위치로 직접 접근하여 공유메모리상으로 불러들인다.

암호화 과정과 비슷한 과정을 거쳐 Secure World의 File Handler에 의해 복호화 된 이후 로컬 스토리지에 저장한다.

IV. 실험결과

본 프레임워크를 적용한 클라우드 스토리지 서비스의 업로드 및 다운로드 성능 측정을 위하여 다음과 같이 구현하였다. ARM Fastmodel[9]을 사용하여 에뮬레이트 된 Cortex-A15 임베디드 보드 위에 Open Virtualization[8] 에서 제공하는 TrustZone 소프트웨어 스택을 사용하였다. Normal World에서는 리눅스(커널 2.6.38)와 파일 탐색기가 동작하고 Secure World에서는 암호화 모듈이 동작한다. 현재 암호화 알고리즘으로 RC4 알고리즘(256-bit key)을 사용한다.

실험으로 512B, 1KB, 2KB, 4KB 데이터에 대해서 암호화를 하지 않은 업로드 및 다운로드 시간 측정, Normal World에서의 암호화, Secure World에서의 암호화, 그리고 공유메모리를 사용한 암호화 네 가지에 대해서 수행하였고 결과는 그림2와 같다.

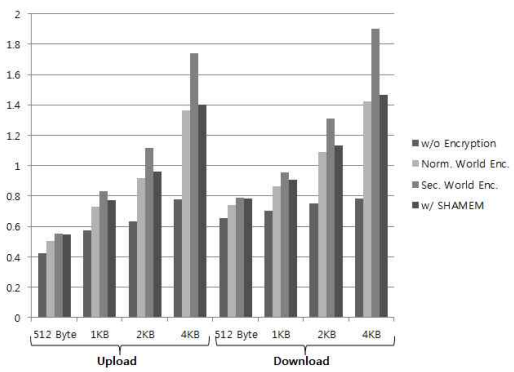

그림 2. 실험 결과
Fig 2. Experimental Result

공유메모리를 사용하지 않고 암호화를 수행할 경우 512B 마다 World 전환이 일어난다. 즉 Monitor를 거쳐서 전달되는 메시지의 최대 크기가 512B이다. 따라서 512B의 배수만큼 World 간의 전환이 일어나 암호화 오버헤드가 커지는 것을 볼 수 있다. 반면에 공유 메모리를 사용하면 한 번의 컨트롤 메시지 교환만으로 암호화를 수행할 수 있으므로 Normal World에서 암호화를 수행하는 시간과 비슷한 결과를 얻을 수 있다.

V. 결 론

본 논문에서는 ARM TrustZone 기반 모바일 환경에서 클라우드 스토리지를 위한 안전한 데이터 관리 기법에 대해서 논의하였다. 클라우드 스토리지 서비스를 사용할 때 발생할 수 있는 데이터 유출 문제에 대응하기 위하여 우리는 클라이언트 측 데이터 암호화 방식을 사용하고 암호화에 사용되는 키를 하드웨어적으로 관리하기 위하여 ARM TrustZone 기술 및 TPM Emulator를 이용하였다. 또한 중앙 서버로부터 모바일 클라이언트에 대한 원격 플랫폼 검증을 수행한 후 암호화 키를 사용하게 함으로써 악성 코드로 부터의 데이터 유출 위협을 줄일 수 있었다.

성능 오버헤드를 줄이고자 데이터를 암호화 하는데 공유메모리를 사용하였고 두 World 사이의 전환을 최소화하였지만 암호화 자체의 오버헤드를 줄이는 것이 필요하다. 또한 Normal World에서 복호화된 데이터의 보호 방법도 향후과제로 남아있다.

참 고 문 헌

[1] Trusted Platform Module (TPM) http://www.trustedcomputinggroup.org/developers/trusted_platform_module/
[2] Software-based TPM Emulator http://tpm-emulator.berlios.de/
[3] National Vulnerability Database. http://nvd.nist.gov/.
[4] CVE-2008-0923. http://eve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-0923
[5] The Blue Pill Project. http://bluepillproject.org/
[6] Amazon S3, “Using Data Encryption” http://docs.amazonwebserrvices.com/AamzonS3/latest/dev/UsingEncryption.html
[7] Dropbox. http://www.dropbox.com
[8] Sierraware, “Open Virtualization for TrustZone Overview”, 2011.
[9] ARM, “ARM Fast Model Reference Manual”, http://infocenter.arm.com/help/topic/com.arm.doc.dui0423m/DUI0423M_fast_model_rm.pdf
[10]ARM, “ARM Securtiy Technology, Building a Secure System using TrustZone Technology”, 2009