Research Article

pISSN : 2287-8920
https://doi.org/10.22716/sckt.2020.8.4.035

모바일 클라우드 스토리지 상의 EMRA 기반의 멀티미디어 콘텐츠 공유 및 보안을 위한 데이터 접근 관리 기법에 관한 연구

이종섭[1], 문석재[2]
[1]세명대학교 교양대학 교수, [2]광운대학교 정보과학교육원 정보보호학 교수

A Study on Data Access Management Technique for Sharing and Security of Multimedia Contents Based on EMRA on Mobile Cloud Storage

Jong-Sub Lee[1] and Seok-Jae Moon[2]
[1]Professor, Semyung University, College of General Education
[2]Professor, KwangWoon University, Institute of Information Technology

[2]**Corresponding author:** msj8086@kw.ac.kr

Received November 1, 2020; **Revised** December 17, 2020; **Accepted** December 18, 2020

ABSTRACT

모바일 클라우드 스토리지 서비스는 다수의 이동 장치에 걸쳐 사용자의 멀티미디어 콘텐츠 데이터를 효율적으로 공유 또는 동기화에 사용된다. 이 모바일 클라우드 스토리지는 다양한 콘텐츠 데이터를 저장하기에 여러 가지 이질성을 해결하는 유연성 및 확장성을 제공하여 공유 과정에서 반드시 보안 문제를 처리해야 한다. 현재 일반적으로 클라우드 스토리지 서비스는 보안 목적을 위해 콘텐츠 데이터를 암호화하여 이용한다. 하지만, 이 방법은 소프트웨어에 의한 단순 ID, 비밀번호와 암호화 키를 관리하는 방법으로 불량 사용자를 식별하기 때문에 안전하지는 않다. 본 논문은 모바일 클라우드 스토리지 상의 EMRA(Extended MetaData Registry Access) 기반의 멀티미디어 콘텐츠 공유 및 보안을 위한 데이터 접근 관리 모델과 기법을 제시한다. 이는 하드웨어 기반의 키 관리, 클라이언트 소프트웨어 무결성에 대한 증명 및 보안 키 공유를 지원한다. 또한 제시한 모델은 각각 서로 다른 형태의 멀티미디어 콘텐츠들을 저장/관리하고 있어 검색 시 발생하는 문제를 EMRA 기반의 메타데이터 관계성을 구축하여 검색에 신뢰성을 높인다. 그리고 본 논문은 ARM TrustZone과 TrustZone 환경의 보안 세계에서 실행되는 TPM 에뮬레이터를 사용하여 프로토타입을 제시한다.

Mobile cloud storage services are used to efficiently share or synchronize users' multimedia content data across multiple mobile devices. This mobile cloud storage provides flexibility and scalability to solve various heterogeneities to store various content data, so security issues must be addressed in the sharing process. Currently, cloud storage services generally encrypt and use content data for security purposes. However, this method is not secure because it identifies rogue users by managing simple IDs, passwords and encryption keys by software. This paper presents a data access management model and technique for sharing and security of multimedia contents based on EMRA on mobile cloud storage. It supports hardware-based key management, proof of client software integrity, and secure key sharing. In addition, the proposed model stores/manages different types of multimedia contents, so that problems arising during search can be solved by establishing a metadata relationship based on EMRA to increase the reliability of the search. And this paper presents a prototype using ARM TrustZone and a TPM emulator running in the secure world of TrustZone environment.

Keywords: EMRA, ARM TrustZone, TrustZone, Mobile cloud, Cloud storage, Data access

1. 서 론

모바일 폰, 태블릿과 같은 단말기가 보급화 되면서 개인이 여러 대의 단말기를 소지하는 일이 흔해지고 있다. 그래서 사용자는 멀티미디어 콘텐츠 데이터를 효율적으로 공유하고 동기화 하기 위한 방안으로 모바일 클라우드 스토리지 서비스를 많이 이용하게 되었다[1]. 하지만 사용자의 데이터가 원격지에서 저장되는 모바일 클라우드 스토리지에 특성 때문에 사용자 멀티미디어 콘텐츠 데이터가 유출 및 변경되는 일이 빈번하게 발생 된다. 따라서 데이터 유출 및 변경에 대한 보안성을 중요시 하는 개인 사용자나 기업은 모바일 클라우드 스토리지 서비스를 기피하는 원인으로 이어지게 된다. 본 논문에서는 모바일 클라우드 스토리지 상의 멀티미디어 콘텐츠 데이터 유출을 최소화하고, 검색 시 안전하게 접근 및 공유할 수 있는 기법을 제시한다. 제시한 기법은 하드웨어 기반의 키 관리, 클라이언트 소프트웨어 무결성에 대한 증명 및 보안 키 공유를 지원한다. 그리고 본 논문은 ARM TrustZone[2,3]과 TrustZone 환경의 보안 세계에서 실행되는 TPM 에뮬레이터[4]를 사용하여 보안 모델을 제시하였다. 그리고 제시한 모델은 각각 서로 다른 형태의 멀티미디어 콘텐츠들을 저장/관리하고 있어 검색 시 발생하는 매핑 문제를 EMRA(Extended MetaData Registry Access)[5] 기반의 메타데이터 관계성을 구축하여 검색에 신뢰성을 높인다. 본 논문에서 제안하는 보안 모델을 적용된 모바일 클라이언트 단말기들은 ARM TZ(TrustZone)이 적용되어 있다고 가정한다. ARM TZ은 ARM 어플리케이션 프로세스에 하드웨어적으로 구현되어 있으며, Normal World, Secure World 두 개의 운영체제들이 독립적으로 실행되도록 지원한다. Secure World는 하드웨어적으로 보호되는 Safety 영역으로 주로 모바일 단말기내에서 보안이 허용되는 어플리케이션이 이 영역에서 실행된다. 따라서 보안이 요구되는 작업을 Secure World에서 실행시켜 사용자 정보 보호, 암호화 키 등을 안전하게 관리할 수 있다. 제안 모델은 ARM TZ을 이용하여 하드웨어 기반으로 키 관리 방법을 지원한다. 모바일 클라이언트 단말기의 Normal World는 모바일 단말기내에서 보안 기법이 허용되지 않는 어플리케이션에서 실행되기 때문에 악성 어플리케이션이 상주할 수 있다. 그래서 이 영역은 신뢰할 수 없는 환경이라고 가정한다. 또한 모바일 클라우드 스토리지 서비스 제공자도 본 논문에서 언급한 멀티미디어 콘텐츠 데이터 유출 위협으로 인하여 신뢰할 수 없다고 가정한다. 이는 모바일 클라우드 스토리지 서비스에서 일어날 수 있는 모든 데이터 유출에 초점을 두었다. 또한 본 논문은 제시한 보안 기법을 이용하여 모바일 클라우드를 통해 단말기간의 멀티미디어 콘텐츠 데이터를 검색 및 공유 할 때 다양한 메타 충돌 문제를 EMRA 기반으로 콘텐츠를 관리 할 수 있게 하였다. 본 논문의 구성은 2장은 관련 연구에 대해 기술한다. 3장은 본 논문에서 제안 모델 구성, 사용자 인증 및 플랫폼, 데이터 암호호 키 생성 및 관리, 데이터 암호화키 공유, 콘텐츠 파일 업로드 및 다운로드, 그리고 보안 평가에 대해서 기술한다. 마지막으로 4장 결론을 기술한다.

2. 관련 연구

클라우드 서비스는 IT 자원을 소유하지 않고 일부 또는 모두를 아웃소싱하는 형태이므로 필수적으로 부안 문제가 제기될 수 밖에 없다. 클라우드 서비스를 이용함에 보안적인 측면에서 우려되는 문제점으로는 보안 및 데이터 유출, 서비스 안정성과 가용성, 기존 애플리케이션 서비스와의 연동 이슈, 서비스 제공업체의 안정성, 법규제, 서비스 비용 등을 들 수 있다. 이러한 문제점들 중에서도 클라우드 서비스 핵심 보안 위협 요소를 정리해보면 다음과 같다[6].

- 가상화 취약점 상속: 악성 코드 감염 및 확산 위협, 서비스 가용성 침해
- 정보 위탁에 따른 정보 유출의 위협: 소유와 관리 분리에 따른 정보 유출, 내부자에 의한 정보 유출
- 사용 단말의 다양성과 분실에 따른 정보 유출: 단말기 분실 등에 의한 정보 유출
- 자원 공유 및 집중화에 따른 서비스 장애: 시스템 장애 시 모든 고객의 서비스 중단, 중앙 시스템 노출시 DDoS등의 공격대상

이 되기 쉬움

- 분산 처리에 따른 보안 적용의 어려움: 자원공유와 가상 머신 동적 재배치로 인증/접근제어 복잡도 상승, 분산 컴퓨팅 시스템에 일괄적인 인증/접근제어 적용의 어려움.
- 법규 및 규제의 문제: 정보 유출시 책임소재 불분명, 자원공유에 따라 감사증적이 어려움.

국외의 클라우드 보안 지침은 2008년 11월 20일에 개최된 보안실무자 컨퍼런스인 ISSA Forum에서 탄생한 CSA(Cloud Security Alliance)에서는 클라우드 컴퓨팅의 안전성 증진과 사용자 교육을 목적으로 만든 비영리기관이다. CSA는 클라우드 도입에 있어 필요한 가이드라인을 크게 3개 영역의 14가지 항목으로 제시하고 있으며, 특히 위험할 수 있는 8가지 위협에 대해 발표하였다[7]. 제시된 CSA의 8가지 보안 위협요소는 기술적인 부분보다는 사람과 관련된 것들이 대부분이다. 이는 곧 기술적으로 위협에 대응책이 있더라고 결국 내부인력에 의해 의도된 위협은 막기 어렵다는 것이며, 내부자 보안 인식 교육이 클라우드 서비스 보안을 위해서는 매우 중요한 요소가 될 것으로 보고 있다. 국내의 클라우드 보안 지침은 한국인터넷진흥원(KISA: Korea Internet & Security Agency)에서는 2010년 10월에 클라우드 서비스에 대한 제공자 측면의 운영적, 기술적 가이드라인과 이용자 측면의 클라우드 도입 가이드라인을 수립하기 위해서 “클라우드 서비스 보안관리 가이드라인”이라는 보고서를 발표하였다[8]. 서비스 제공자의 경우에는 크게 운영적인 측면의 보안과 기술적인 보안의 측면으로 분류하였다. 운영적인 측면의 보안에 관해서는 클라우드 컴퓨팅 서비스 보안 정책 및 규정의 수립, 보안 조직운영, 자산 분류 및 통제, 사고관리 절차 수립, 서비스 연속성, 컴플라이언스와 관련된 권고를 하고 있고, 그 내용은 통상적인 IT서비스 제공자에 대한 보안권고와 크게 다르지 않다. 기술적인 보안에 대해서는 크게 ‘네트워크 및 시스템 보안’, ‘데이터 및 스토리지 보안’, ‘애플리케이션 보안’, ‘이용자 식별 및 접근관리’로 나누고 있다. 서비스 이용자 보안가이드에는 서비스 사업자 선택, 서비스 사용 보안 수칙, 개인 이용자 보안규칙, 서비스 이전 계획 등으로 나뉘며, 그 외에도 클라우드 서비스 보안을 위한 체크리스트가 있다.

본 논문에서 적용하는 EMRA은 미국의 Lawrence Berkeley National Laboratory에서 ISO/IEC 11179 메타데이터 레지스트리의 향상된 표준을 개발, 제안, 테스트하는 프로젝트이다[9]. MDR의 파트 3을 확장하여 메타데이터 레지스트리 시스템에서 데이터 요소(Data Element), 용어(Term), 개념(Concept) 구조들의 의미를 저장, 관리를 향상을 목적으로 한다. XMDR에서는 개념 시스템(Concept System)이라는 개념을 도입하여 여러 가지 도메인의 메타데이터 레지스트리 시스템들을 각각의 Concept_System에 관리할 수 있고, 온톨로지 개념의 적용을 통해 서로 다른 메타데이터 레지스트리 시스템의 메타데이터간 관계 정의가 가능하도록 개발되었다. 다시 말해, 메타데이터 레지스트리에서 온톨로지의 다양한 클래스와 개념들의 정의, 속성, 관계 등 다양한 온톨로지의 특성을 저장, 관리할 수 있도록 확장하였다. 사용자가 모바일 클라우드 에이전트에서 단말기간의 멀티미디어 콘텐츠 데이터를 검색 및 공유 할 때 다양하게 나타나는 메타 충돌 문제를 EMRA 이용한다.

3. 제안 모델

3.1 제안 모델 구성

본 논문에서는 모바일 클라우드 스토리지 서비스 상의 멀티미디어 콘텐츠 데이터 유출을 최소화하고, 안전하게 접근할 수 있는 기법을 제시한다. 또한 제시한 보안 기법을 이용하여 모바일 클라우드를 통해 단말기간의 멀티미디어 콘텐츠 데이터를 공유시 발생하는 이질성에 대한 문제를 EMRA로 해결한다. Fig. 1은 Mobile Device, Mobile Cloud Agent, Cloud Storages로 구성되며, Mobile Device는 Secure World, Normal World로 구성되고, ARM에 TZ(TrustZone) 기술이 적용된 Processor Hardware로 구성된다. Mobile Cloud Agent는 Key Distributor, Attestation Service, Hash Store, EMRA를 포함한다. 그리고 Cloud

Storages는 Data Block 저장소 형태로 멀티미디어 콘텐츠 데이터(이하 콘텐츠 데이터)를 관리한다. 다음은 본 논문의 핵심인 Mobile Device 구성 요소와 Mobile Cloud Agent 구성요소에 대한 설명이다.

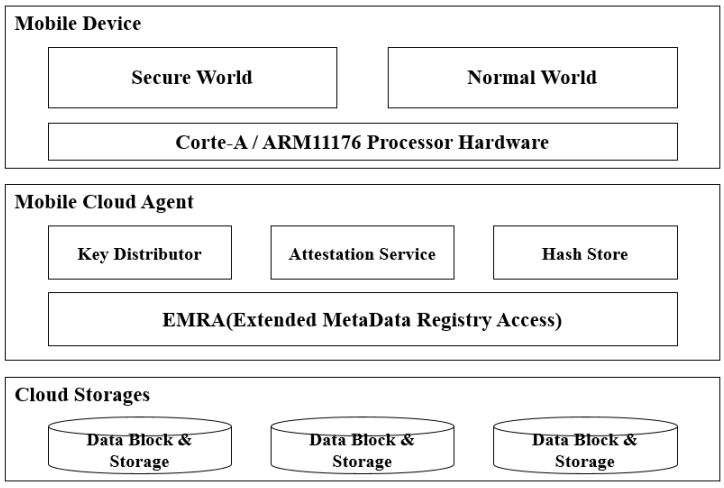

Fig. 1. Proposed model overview

- Mobile Device: 이 구성 요소는 사용자가 단말기에서 사용자 인증을 이용하여 콘텐츠 데이터에 대한 접근 관리를 하고, 또한 데이터 키 관리, 플랫폼 무결성 검증, 파일 암호화 및 복호화 기능을 제공한다.
- Mobile Cloud Agent: 이 구성 요소는 사용자가 콘텐츠 데이터 공유에 대한 서비스 받도록 지원해주는 에이전트이다. 이는 키 분배기, 파일 검증 서비스, 해쉬 저장소를 이용하여 서로 다른 사용자 또는 다른 단말기기들 간에 데이터 암호화 키 교환 프로토콜을 지원하고, EMRA를 통해 콘텐츠 데이터에 대한 이질성을 해결하는 기능을 제공한다.
- Cloud Storages: 이 구성 요소는 모바일 단말기에 멀티미디어 콘텐츠 데이터가 클라우드 스토리지에 관리되는 기능을 제공한다. Fig. 2는 본 논문에서 제시한 모델의 모바일 단말기와 모바일 클라우드 에이전트 간의 콘텐츠 데이터 보안 및 접근 동작하는 과정이다.

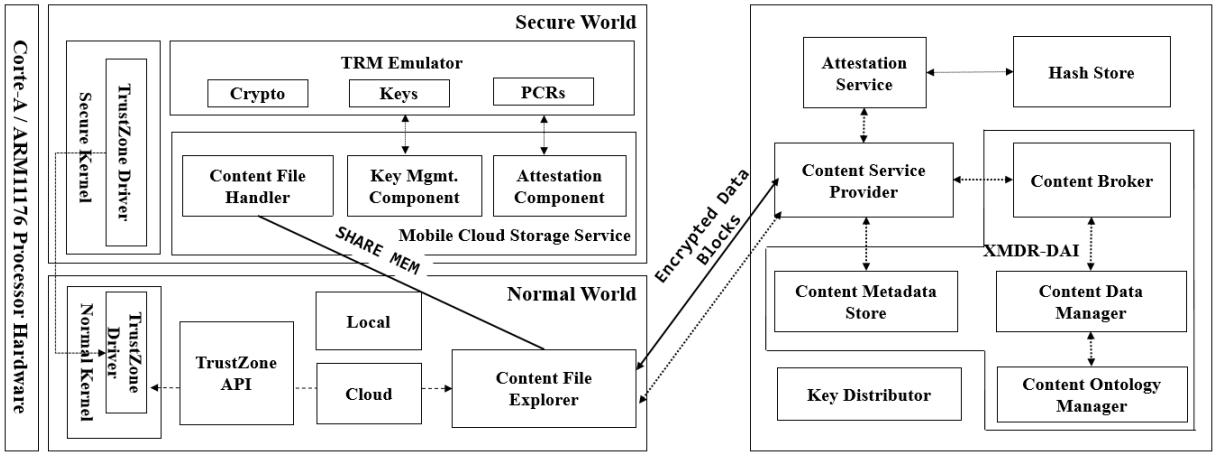

Fig. 2. Data access and security between mobile cloud agent and ARM TrustZone of mobile terminal

본 논문에서 제시한 모델은 Fig. 2와 같이 ARM TZ(TrustZone)이 적용된 MD(Mobile Device), MCA(Mobile Cloud Agent), CS(Cloud Storages)로 구성요소를 가진다. 사용자는 모바일 단말기의 Normal World에서 실행하는 콘텐츠 파일 탐색기 어플리케이션을 통하여 사용자 인증, 로컬 및 클라우드 콘텐츠 파일 탐색, 그리고 업로드와 다운로드, 삭제, 이동 등 콘텐츠 파일 오퍼레이션을 수행한다. 또한 모바일 클라우드 에이전트에서 데이터 키 관리, 플랫폼 무결성 검증, 파일 암호화 및 복호화 기능을 제공하는 보안 프로세스들은 Secure World에서 실행한다. 사용자가 콘텐츠 파일 탐색기를 통하여 클라우드 파일에 접근하면, TZ API가 호출되고 TZ 모니터에 의해 두 World가 전환되면서 해당하는 요청 메시지가 전달된다. Secure World의 클라우드 스토리지 서비스 프로세스가 전달받은 메시지에 대한 처리를 하며, 키 관리 기능들은 Secure World의 TPM Emulator를 이용하여 처리한다. MCA는 클라이언트의 플랫폼 무결성 검증을 수행하고 암호화된 사용자 데이터를 클라우드 에이전트로부터 전송 받은 후, 클라우드 스토리지 API를 사용하여 클라우드 스토리지에 저장하며 유저 ID, 파일명, 그리고 실제 저장된 클라우드 스토리지 위치로 이루어진 콘텐츠 파일 메타데이터를 기록한다. 클라우드 에이전트에서 다운로드가 수행될 때 메타데이터를 검색하여 파일이 실제로 저장된 클라우드 스토리지 위치를 클라이언트로 전송함으로써 클라우드 에이전트로부터 해당 클라우드 스토리지에 직접 접근이 가능하도록 한다. 또한 Key Distributor를 통하여 서로 다른 사용자 또는 다른 단말기기들 간에 데이터 암호화 키 교환 프로토콜을 지원한다. 모바일 클라우드(모바일 단말기) API 계층은 여러 개의 다른 상용클라우드 스토리지를 선택적으로 사용할 수 있게 한다. 따라서 사용자 데이터는 클라우드 스토리지 선택 정책에 따라 임의적인 위치에 저장될 수 있다.

3.2 사용자 인증 및 플랫폼 검증

본 제안 모델에서는 CRTM(Core Root of Trust for Measurement)을 Secure World로 가정한다. 모바일 단말기의 부팅과정에서 Secure World의 Attestation Component가 Normal 부트로더 및 OS 이미지의 무결성을 검증하고 검증 결과 값을 TPM Emulator(emul.)의 PCR(Platform Configuration Register)에 저장한다. Normal OS의 부팅 이후에 IMA(Integrity Measurement

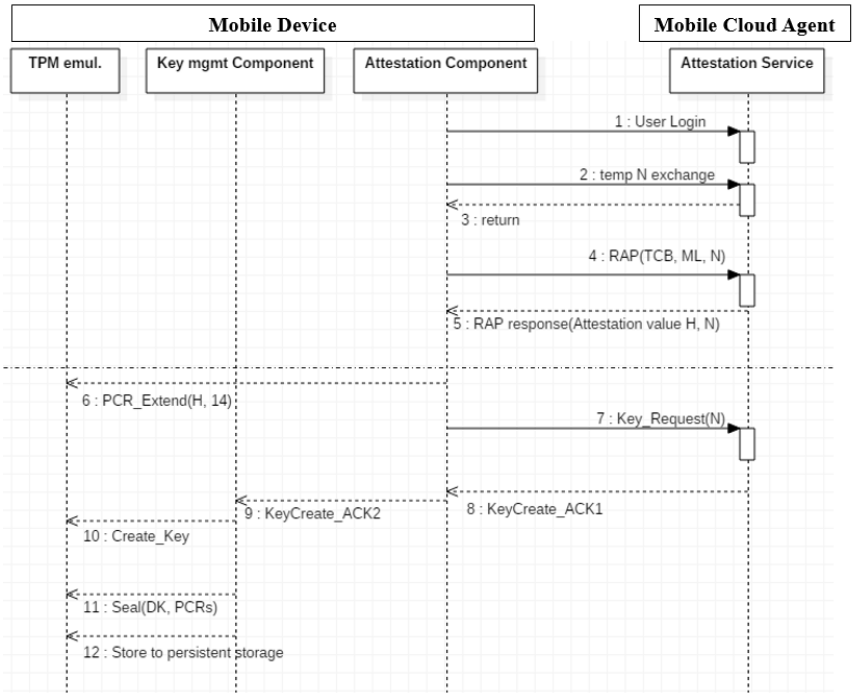

Fig. 3. User and platform authentication process and data key generation and management protocol

Architecture)에 의해서 라이브러리, 실행된 프로세스 및 시스템 설정 파일들의 무결성을 측정하고 TZ API를 통하여 PCR에 extend 한다. 사용자는 클라우드 콘텐츠 파일 탐색기를 실행하여 자신의 인증정보 (ID/password)로 로그인하게 되고 모바일 클라우드 에이전트와 모바일 단말기간에 temp값을 교환함으로써, 세션이 시작된다(Fig. 3).

원격 검증 프로토콜에 의하여 TPM Emulator에서 quote 명령으로 PCR에 저장된 무결성 검증 정보와 ML(Measurement Log) 값, 그리고 N값을 모바일 클라우드 에이전트에 전송함으로써 플랫폼 검증을 수행한다. 모바일 클라우드 에이전트의 Attestation Service는 Hash Store에 저장된 값과 전송 받은 무결성 정보를 비교한 다음 현재 모바일 단말기가 안전한 상태에 있는지 검사하고 결과를 모바일 단말기로 전송한다. 인증이 성공하면 모바일 클라우드 에이전트가 보내는 원격 검증 결과 메시지에는 사용자에게 부여된 인증 값 H가 같이 전송되고, 이는 모바일 단말기 TPM Emulator의 PCR에 저장한다. 이는 추후 사용자 암호화 키를 모바일 단말기 내 저장소에 안전하게 보관하기 위한 프로토콜에서 사용된다. 인증이 실패하면 사용자는 자신의 클라우드 스토리지에 접근 할 수 없다.

3.3 데이터 암호화 키 생성 및 관리

사용자 인증 및 플랫폼 검증이 성공적으로 끝나면 데이터 암호화에 사용할 암호화 키를 TPM에 불러오는 과정이 시작 된다. 먼저 모바일 클라우드 에이전트에 사용자의 키 생성 기록을 조회한 후 (Key_Request), 모바일 클라우드 에이전트의 응답에 따라 키 생성 (Key_Creation_ACK), 키 로드(Key_Loading_ACK), 또는 키 공유 프로토콜(Key_Remote_ACK)을 수행한다. 인증을 통과한 사용자가 처음 접속한 사용자이면 모바일 클라우드 에이전트에 키 생성 정보가 없으므로 클라이언트에서 자신의 암호화 키를 생성하여야 한다(Fig. 3). 따라서 Key Mgmt. Component가 TPM Emulator에게 키 생성 요청을 하고 TPM Emulator에 의해서 사용자의 클라우드 스토리지 데이터 암호화에 사용할 데이터 키 DK를 생성한다. 모바일 단말기는 키가 생성 되면 모바일 에이전트에 키 정보와 모바일 기기 정보를 등록한다. 사용자 클라우드 스토리지로의 접근이 종료되어 데이터 키가 사용되지 않을 때, TPM Emulator내 즉, Secure World의 메모리 영역에 로드되어 있는 데이터 키를 모바일 기기의 로컬 스토리지에 안전한 형태로 보관 해한다. 따라서 TPM Emulator의 Seal 명령을 통하여 sealing 된다. Sealing 과정에서 사용되는 값은 현재 기기의 무결성 정보를 담은 PCR 값과 사용자 및 플랫폼 인증의 결과로 서버로부터 전송받은 인증 값 가 쓰인다. Sealing된 키가 다시 사용되어야 할 때에는 (이미 키 생성 기록을 가진 사용자이고 해당 모바일 기기가 키를 가지고 있는 경우) unsealing 과정을 통하여 TPM Emulator에 불러 온다. Sealing 때와 마찬가지로 unsealing 될 때의 모바일 단말기 무결성 정보 (PCR 값)와 모바일 에이전트로부터 전송 받은 인증 값 H가 사용되며 sealing 될 때와 다를 경우 사용자는 데이터 키를 사용할 수 없다.

3.4 데이터 암호화 키 공유

사용자 암호화 키가 생성이 되었지만 현재 모바일 기기에 sealing 되어 보관되지 않은 경우 원격 키 공유 프로토콜을 사용하여 키를 마이그레이션 해야 한다. 또한 클라우드 스토리지의 사용자간 또는 공유 그룹간의 공유가 필요할 때에도 동일한 키 공유 프로토콜을 사용할 수 있다. 데이터 키를 가진 클라이언트는 해당 키가 다른 기기 또는 다른 사용자 (그룹)에게 공유할 필요성 (요청)이 있으면 이를 중앙 서버에 전송하여 저장을 하고 공유 대상 클라이언트에서이를 전송 받아 공유 클라우드 스토리지의 암호화된 데이터에 접근 할 수 있다. 먼저 공유 대상 클라이언트 (target, Device B)는 서버와의 secure network session을 통하여 자신의 public 키 K를 중앙 서버에 등록 한다(Fig. 4). 공유할 키를 가진 클라이언트 (source, Device A)는 공유 요청에 따라 Device B의 KB를 서버로부터 전송 받고 sealing 된 키를 unsealing 과정을 거쳐 TPM Emulator로 불러 들인 후 (DKA), KB로 암호화 한다. 암호화 된 Device A의 데이터 키는 서버에 임시 저장된다. Device B는 그림 2의 인증과정을 거친 후 서버에 Key_Request 요청을 하고 서버로부터 Key_Remote_ACK 메시지와 함께 {DKA} KB를 전송 받는다. TPM Emulator에서 Device B의

private key로 복호화된 DKA는 공유된 클라우드 스토리지 데이터에 접근할 때 사용되고 클라우드 스토리지 사용이 종료 되면 sealing 과정을 거쳐 Device B의 로컬 스토리지에 안전하게 보관된다.

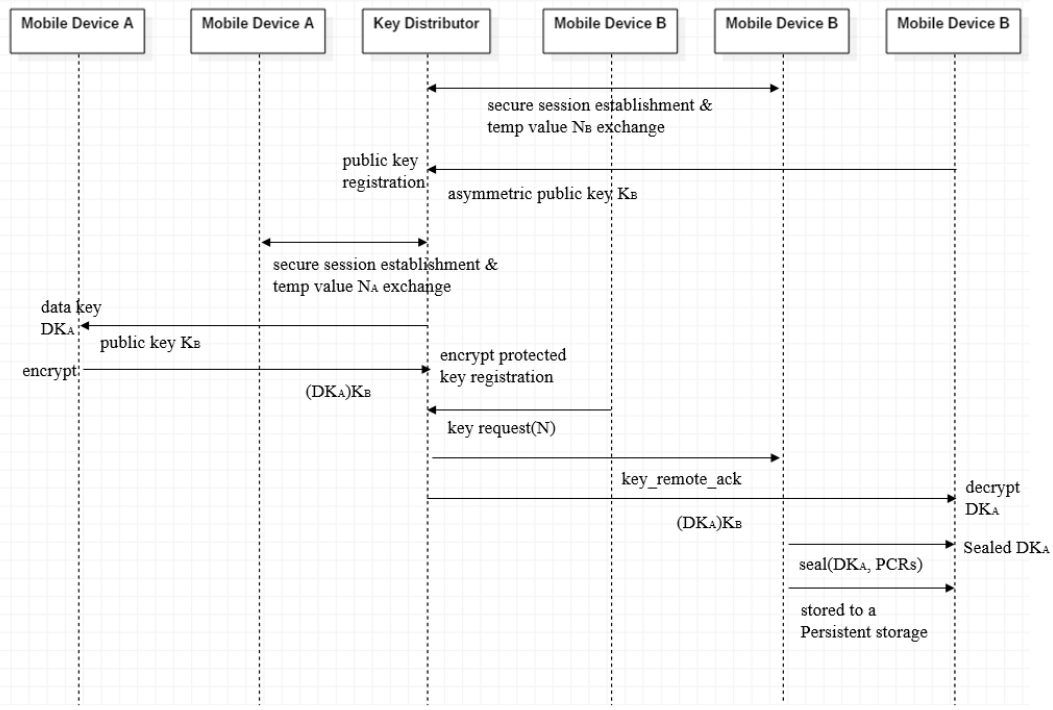

Fig. 4. User data encryption key sharing protocol

3.5 콘텐츠 파일 업로드 및 다운로드

사용자가 로컬 파일을 클라우드 스토리지에 업로드할 때 먼저 Secure World와 Normal World의 공유메모리에 콘텐츠 데이터를 불러들인다. TZ API를 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. 콘텐츠 파일 탐색기는 공유 메모리 상의 암호화된 데이터를 서버로 전송한다. 클라우드 스토리지의 데이터를 다운로드 할 때에는 서버로부터 전송 받은 데이터의 실제 위치로 직접 접근하여 공유메모리상으로 불러들인다. 암호화 과정과 비슷한 과정을 거쳐 Secure World의 File Handler에 의해 복호화 된 이후 로컬 스토리지에 저장한다.

본 모델을 적용한 모바일 클라우드 스토리지 서비스의 업로드 및 다운로드 성능 측정을 위하여 다음과 같이 구현하였다. ARM Fastmodel[3]을 사용하여 에뮬레이트 된 Cortex-A15[10] 임베디드 보드 위에 Open Virtualization 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 Fig. 5와 같다. 각각 사용자에 의해서 업로드 또는 다운로드 명령을 내렸을 때로부터 완료되기까지의 시간을 측정하여 나타내었다. 공유메모리를 사용하지 않고 암호화를 수행할 경우 512 바이트의 데이터 마다 World 전환이 일어난다. 즉 TrustZone Secure Monitor를 거쳐서 전달되는 메시지의 최대 크기가

512 바이트 이다. 따라서 512 바이트 데이터 마다 World 간의 전환이 일어나 암호화 오버헤드가 그에 비례하여 커지는 것을 볼 수 있다 (512 바이트의 경우 Normal World 암호화와 Secure World 암호화의 차이는 World 간의 전환에서 발생하는 오버헤드이고 평균 0.05 초이다.). 이는 World 전환의 메시지 전달 채널 구현상의 한계로 인하여 발생하는 오버헤드이며 이를 해결하기 위한 한 가지 방법으로 본 프로토타입에서는 두 World의 데이터 전달을 위한 공유 메모리 방식을 사용하였다. 실험 결과에서 나타나듯 Normal World에서 암호화를 수행하는 시간에 데이터 전달 오버헤드가 더해지는 Secure World 암호화와는 다르게 공유 메모리를 사용하면 한 번의 컨트롤 메시지 교환만으로 암호화를 수행할 수 있으므로 Normal World에서 암호화를 수행하는 시간과 비슷한 결과를 얻을 수 있다.

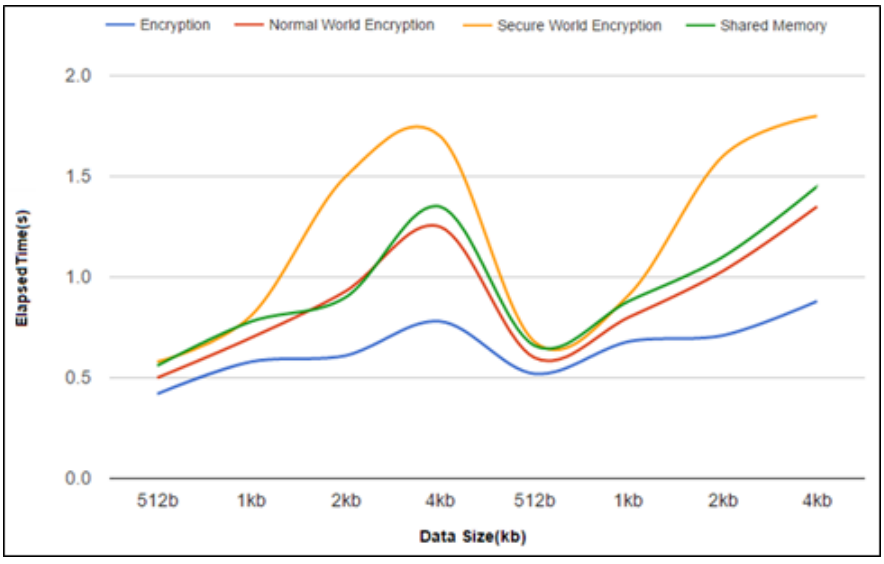

Fig. 5. Experimental test results

3.6 보안 평가

본 논문에서는 모바일 클라이언트 단말기에서 모바일 클라우드 스토리지 서비스를 사용할 때 발생할 수 있는 사용자 데이터 유출을 막고자 하는데 중점을 두고 제시하였다. 모바일 클라우드 스토리지 측의 데이터 유출 즉, 인프라 보안 허점을 악용한 경우에 대응하고자 모바일 클라이언트 단말기 측에 암호화를 사용하였고, 암호화에 사용되는 키를 하드웨어적으로 보호하기 위한 방법으로 ARM TZ를 이용하여 일반 사용자의 작업 공간과 독립된 실행환경에서 TRM Emulator를 실행시켰다. 멀티미디어 콘텐츠 데이터 암호화/복호화를 비롯한 보안 관련 모듈을 보안 환경에서 실행시키고 키가 실제로 사용될 때 원격 플랫폼 무결성 검증을 통하도록 함으로써, 일반 사용자 작업 공간에서의 비정상적인 접근을 차단하고 악성 프로그램의 설치 유무를 감지할 수 있다.

4. 결 론

본 논문에서는 ARM TrustZone 기반 모바일 클라우드 스토리지를 위한 안전한 데이터 관리 기법에 대해서 제시하였다. 모바일 클라우드 스토리지 서비스를 사용할 때 발생할 수 있는 멀티미디어 콘텐츠 데이터 유출 문제에 대응하기 위하여 본 논문에서는 모바일 클라이언트 단말기 측에 암호화 방식을 사용하고 암호화에 사용되는 키를 하드웨어적으로 관리하기 위하여 ARM TrustZone 기술 및 TPM Emulator를 이용하였다. 또한 모바일 클라우드 스토리지로부터 모바일 클라이언트 단말기에 대한 원

격 플랫폼 검증을 수행한 후, 암호화 키를 사용하게 함으로써 악성 코드로부터의 데이터 유출 위협을 줄일 수 있었다. 성능 오버헤드를 줄이고자 데이터를 암호화 하는데 공유메모리를 사용하였고, Normal World와 Secure World 사이의 전환을 최소화하였지만, 암호화 자체의 오버헤드를 줄이는 것이 필요하다. 또한 Normal World에서 복호화된 데이터의 보호 방법도 향후 과제로 남아있다.

Acknowledgement

※ 이 논문은 2020년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임.

References

1. Cui, Yong, Zeqi Lai, and Ningwei Dai. “A first look at mobile cloud storage services: architecture, experimentation, and challenges.” IEEE Network 30.4 (2016): 16-21.
2. Wan, Shengye, et al. “RusTEE: Developing Memory-Safe ARM TrustZone Applications.” Annual Computer Security Applications Conference. 2020.
3. Nicholas, Geraldine Shirley, Yutian Gui, and Fareena Saqib. “A Survey and Analysis on SoC Platform Security in ARM, Intel and RISC-V Architecture.” 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS). IEEE, 2020.
4. Wesemeyer, Stephan, et al. “Formal Analysis and Implementation of a TPM 2.0-based Direct Anonymous Attestation Scheme.” Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. 2020.
5. Lee, Jong-Sub, and Seok-Jae Moon. “Business Collaborative System Based on Social Network Using MOXMDR-DAI+.” International Journal of Advanced Culture Technology 8.3 (2020): 223-230.
6. https://www.itu.int/en/ITU-T/studygroups/2017-2020/17/
7. Chandra, Angelia. “Measurement of the Cloud Security Level at Company using Cloud Control Matrix.”
8. https://www.kisa.or.kr/
9. https://www.iso.org/standard/68766.html/
10. McLaurin, Teresa, Frank Frederick, and Rich Slobodnik. “The DFT challenges and solutions for the ARM® Cortex™ -A15 Microprocessor.” 2012 IEEE International Test Conference. IEEE, 2012.