Journal of Korea Multimedia Society Vol. 14, No. 2, February 2011(pp. 238-248)

USIM 기반 안드로이드 플랫폼에서의 어플리케이션 라이센스 관리 기법

이윤석[†], 김 은[††], 정민수[†††]

요 약

모바일을 통한 다양한 형태의 서비스는 스마트 폰의 등장으로 더더욱 확대되고 있다. 사람들은 WiFi를 통해 자유롭게 통신을 하며, 정보를 가공, 재배포 한다. 이러한 자유로움이 때로는 문제를 일으키기도 하는데, 안드로이드 플랫폼에서의 어플리케이션 라이센스가 그러하다. 어플리케이션에 대한 재배포의 문제는 상업과 비상업을 구분하지 않고 무차별적으로 이루어지고 있다. 안드로이드 플랫폼에서는 이를 해결하기 위하여 라이센스 서버를 통해 어플리케이션의 라이센스를 검증하는 절차를 거치게 되는데, 이 방법은 두 가지 문제점을 가진다. 하나는 무선 인터넷이 동작하지 않는 공간에서의 정확한 라이센스 인증이 불가능 하다는 것이고, 다른 하나는 이 라이센스 데이터가 스마트 폰에 직접 저장되어 이에 대한 공격이 가능한 것이다. 본 논문에서는 이러한 문제를 해결하기 위하여, USIM내에 라이센스 데이터를 저장하기 위한 파일 구조를 설계 및 구현하여, 안전하게 라이센스 데이터를 관리하고, 저장된 라이센스 데이터를 인증하는 기본 그리고 외부 인증 방법을 제시하여, 온라인 또는 오프라인 상에서의 안전한 라이센스 인증이 가능하도록 하였다.

Management Method of an Application License in the Android Platform Based on USIM

Yun-Seok Lee[†], Eun Kim[††], Min-Soo Jung[†††]

ABSTRACT

Various services through mobile have been expanding since the development of the Smartphones. People freely communicate, remake and redistribute the information through Wi-Fi. People's freedom can cause lots of problems. There are application licenses of various types in the Android platform. However, problems about redistribution are indiscriminately distributed what can't be disentangled commercial and noncommercial use. To solve these problems, an application program has to go through procedures of license check in the Android platform. This method has two problems. One, an exact license authentication is impossible in the place where Wi-Fi does not work. The other thing, this license information is stored directly into the Smartphone. Therefore, attacks are possible about the stored license information. To solve these problems, we designed and implemented the file structure for storing the license information into USIM. We can store and authenticate them. Thus, we propose above the authentication method that can be authenticated the license authentication safely in on or off-line.

Key words: License(라이센스), Content Protection(컨텐츠 보호), Android Platform(안드로이드 플랫폼)

※ 교신저자(Corresponding Author) : 정민수, 주소 : 경상남도 창원시 마산합포구 월영동 449번지 경남대학교 1공학관 8층 자바OS 실습실(631-701), 전화 : 010)6574-7633, FAX : 055)248-2554, E-mail : msjung@kyungnam.ac.kr
접수일 : 2010년 11월 17일, 수정일 : 2011년 1월 20일
완료일 : 2011년 2월 10일
[†] 준회원, 경남대학교 컴퓨터공학과
(E-mail: lysis2jt@naver.com)
[††] 준회원, 경남대학교 컴퓨터공학과
(E-mail: sil7777@nate.com)
[†††] 종신회원, 경남대학교 컴퓨터공학과
※ 본 연구는 2010년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2010-0017069)

1. 서 론

모바일 환경의 발전으로, 사용자는 모바일 단말을 통해 다양한 형태의 작업을 할 수 있게 되었다. 그리고 이러한 모바일 환경은 안드로이드 플랫폼의 등장으로 크게 성장하고 있다. 안드로이드 플랫폼의 특징은 뛰어난 UI와, 무선 인터넷 기술의 결합으로, 기존 데스크톱 환경에서 동작하던 다양한 형태의 어플리케이션이, 모바일 환경에 탑재되기 시작하였다[1-5]. 이처럼 다양한 형태의 어플리케이션은 마켓이라는 공간에서 손쉽게 다운로드 되고, 쉽게 배포되기도 하며, 또한 모바일에서 모바일로의 이동 역시도 가능하게 되었다[1-5]. 이와 같이 손쉬운 배포는, 사용자에게 어플리케이션에 대한 손쉬운 접근성을 제공하지만, 이와 반대로 무분별한 배포로 건전한 어플리케이션 시장의 성장을 방해하는 요소가 되기도 한다. 이와 같은 저작권에 위배되는 어플리케이션의 배포를 방지하기 위하여, 안드로이드 플랫폼에서는 LVL (License Verification Library)을 제공하고 있다[6]. 하지만 이 LVL의 경우에는 라이센스 서버와 어플리케이션의 통신에 의해 라이센스를 검증하는 방식으로, 무선인터넷이 동작하지 않는 공간에서의 라이센스 검증에 대한 부분이 없으며, 또한 라이센스 데이터가 안드로이드 플랫폼 내에 존재함으로 해서 디컴파일과 같은 방식으로 손쉽게 공격이 가능하다는 것이 검증되었다[7].

본 논문에서는 이와 같은 문제를 해결하기 위하여, 라이센스 데이터를 안전한 USIM 내에 저장 및 관리할 수 있도록 USIM 파일 시스템을 설계 및 구현하였으며, 무선인터넷이 가능한 공간과, 가능하지 않는 공간에서도 라이센스 데이터를 인증할 수 있는 인증 기법을 제안하여, 언제 어느 곳에서든 어플리케이션을 사용할 수 있도록 하였으며 또한, 사용자가 기기를 변경하더라도 어플리케이션을 재 구매하는 절차 없이 USIM 칩의 교환으로 어플리케이션을 사용할 수 있도록 하여 어플리케이션의 저작권을 보호 할 수 있다.

본 논문의 구성은, 2장에서는 안드로이드 플랫폼과, 라이센스와 관련된 OMA DRM 그리고 LVL에 대한 분석을, 3장에서는 제안방식의 설계, 4장에서는 제안 방식의 구현 및 테스트, 5장에서는 제안방식의 분석 그리고 마지막 6장에서는 결론을 도출한다.

2. 관련연구

안드로이드 플랫폼은 현재 3G 기반 운용 방식이다. 이 때문에 각각의 스마트 폰 내에는 USIM 칩이 내장되어 있다. 이 USIM에는 사용자의 고유 식별 정보와, 폰 북과 같은 개인 정보들이 저장된다[8]. 본 논문에서 제안하는 방식은 라이센스 데이터를 USIM 내에 안전하게 저장하고 인증하는 방식으로 관련연구로 안드로이드 플랫폼과, OMA에서 제안하는 DRM 방식, 그리고 구글에서 제안하는 라이센스 관리방식에 대해 기술한다.

2.1 안드로이드 플랫폼 구조

안드로이드 플랫폼은 그림 1에서 보는 것과 같이 크게 4개의 계층으로 이루어져 있다[1-5]. 먼저 가장 하위에는 Linux Kernel로 하드웨어에 대한 드라이버와, 스마트 폰의 구동과 직접적으로 관련된 부분이 구현되어있다[1-5]. 그리고 그 위에는 Libraries로 안드로이드의 운용을 위해 필요한 보안, DB, 매체에 대한 라이브러리들을 포함하고 있으며, 또한 여기에는 안드로이드 런타임이 한 영역을 차지하고 있다. 안드로이드 런타임의 경우에는 Dalvik VM과 Core Libraries가 존재하는데, 안드로이드는 리눅스 커널을 기반으로 하여, 자바를 모태로한 가상기계가 탑재되게 되어있다. 그리고 이 가상기계의 모든 명령에 대한 실행과 공통 핵심 라이브러리가 바로 안드로이드 런타임이다. 그리고 그 위의 계층에는 어플리케이션 프레임워크가 존재한다[1-5].

어플리케이션 프레임워크에는 안드로이드에 향후 탑재될 어플리케이션들이 원활히 스마트 폰의 자원을 활용할 수 있도록 각종 관리자와, 라이브러리의 손쉬운 사용을 위한 연결 작업을 수행한다. 그리고 어플리케이션과 위젯은 이러한 어플리케이션의 프레임워크와 라이브러리들을 활용하는 사용자에게 직접적인 서비스를 제공하는 어플리케이션이다[1-5]. 그리고 본 논문에서 라이센스를 등록하여 안전하게 관리하는 것 역시 이 어플리케이션이다.

2.2 OMA DRM 인증 방식

OMA DRM은 Open Mobile Alliance에서 제안하는 DRM 방식으로 현재 2.0까지 제안되었다[9,10]. 이

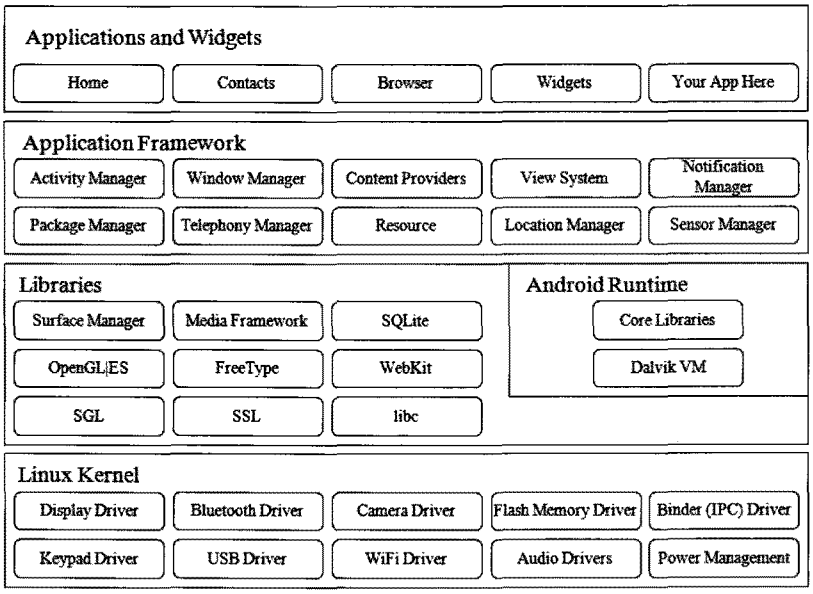


그림 1. 안드로이드 플랫폼 구조

는 그림 2에서 보는 것과 같은 기능적 구조를 가지고 있다[9,10]. 각각의 컨텐츠를 키를 통해 암호화 하고 이 키는 디바이스 도메인 즉 사용자가 소유한 다양한 디바이스에 배포가 가능한 키의 형태로 라이센스를 발급하여, 라이센스를 발급받은 사용자는 이 키를 통해 자신이 소유한 컨텐츠를 디바이스들에 모두 배포 사용할 수 있는 구조이다[9,10]. 기본적으로 PKI 기반 운용방식이다. 이 방식은 컨텐츠와 권리객체(RO:Right Object)의 포맷과 보호방법, 암호키 관리에 대한 보안 모델을 정의하고 있다[9,10].

이와 같은 OMA DRM 방식의 경우에는 자신의 도메인 단위 라이센스를 구매하여 해당하는 도메인에 속한 디바이스에 모두 배포가 가능하다는 장점과 권리 객체와 미디어 객체가 분리되어 있어 미디어 객체 즉 컨텐츠를 다른 곳으로 옮겨도 권리 객체가 없으면 실행할 수 없도록 하는 구조이다. 하지만, 해당하는 DRM에이전트가 모두 공개키와 개인키 쌍을 가지고 있어야 하며, 개인의 휴대 단말에 대한 기기 변경 시에 권리객체를 다시 발급 받아야 한다는 단점이 있다[9,10].

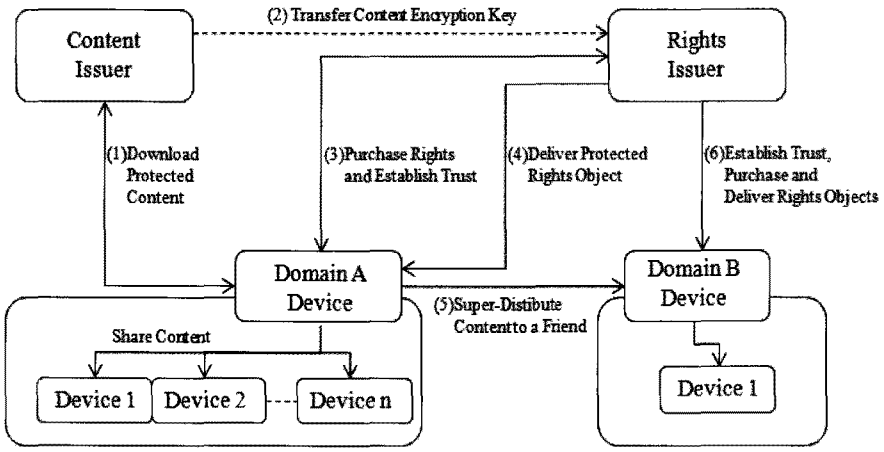


그림 2. OMA DRM 2.0 방식의 기능적 구조

2.3 안드로이드 라이센스 인증 방식

안드로이드 플랫폼에서의 라이센스 인증 방식은 그림 3에서 보는 것과 같다[6,7]. 안드로이드 플랫폼을 제작한 구글에서는 어플리케이션에 대한 라이센스를 관리할 수 있는 라이브러리를 제공하고 있다. 이 라이브러리가 바로 LVL(License Verification Library)이다[6,7]. 이 LVL은 안드로이드 라이브러리에 추가되어 어플리케이션의 메인 엑티비티와 자료를 교환하여 라이센스를 인증한다[6,7].

LVL의 기본적인 인증 방식은 안드로이드 마켓 어플리케이션을 통하여 사용자의 기본 정보와 어플리케이션의 정보 그리고 기타 요구된 정보를 포함하여 마켓 라이센스 서버에 보내고 마켓 라이센스 서버는 해당하는 정보를 검증하여 라이센스의 정보를 PKI 기반 개인키로 서명하여 안드로이드 마켓 어플리케이션으로 송신, 어플리케이션 사용자를 인증한다[6]. 이 과정에서 라이센스 상태에 대한 정보는 단말 내에 기록될 수 있으며, 이에 대한 내용은 정책에 의해 결정된다[6]. 이 방식은 기본적으로 네트워크가 제공되는 환경에서의 인증이 주가 된다. 그리고 네트워크가 안 되는 환경에서는 기록된 라이센스 상태 정보를 통해서 인증을 수행할 수 있으나, 이때 이 기록된 상태 정보를 획득 하여 공격하는 간단한 디컴파일과 같은 형태의 공격 방법이 가능하다[7].

3. 제안 방식의 설계

본 논문에서 제안하는 라이센스 관리 시스템은 크게 두 가지로 구분할 수 있다. 먼저 USIM의 기본적인 목적과 동작을 방해하지 않는 한도 내에서 파일 구조를 설계하고, 이 파일 구조 내에 라이센스를 안전하게 저장 및 관리 할 수 있도록 하는 부분과, 실제 라이센스의 내용 구조와 이를 안전하게 발급하고 인증하기 위한 인증 기법에 관한 부분이다.

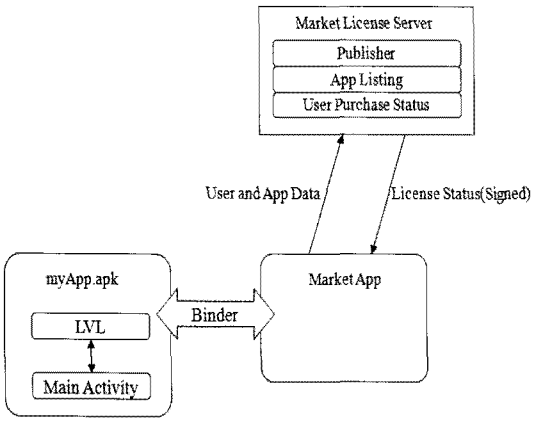


그림 3. LVL을 이용한 라이센스 관리 방법

3.1 제안 방식의 구조.

본 논문에서 제안하는 방식은 그림 4와 같다. 어플리케이션의 배포 경로는 크게 2가지로, 그림 4에서 보는 것과 같이 (1)의 방식과 (1-1)의 방식이 있다. (1)의 방식은 마켓 또는 어플리케이션을 제공해 주는 곳으로부터 다운로드 하는 방법이고, (1-1)의 방식은 스마트 폰으로부터 다른 스마트 폰으로 직접 송신하는 방법이다. 라이센스에 대한 발급과 인증은 이 두 가지 어플리케이션 탑재에 대해 모두 고려하여야 한다.

그리고 라이센스 데이터에 대한 요청은 (2)와 (2-1)과 같은 방식으로 이루어지는데 이는 어플리케이션을 탑재한 모든 안드로이드 폰에서 가능하여야 한다. 그리고 라이센스를 다운로드 하는 방법은 (3)과 (3-1)이 있다. (3)의 경우에는 라이센스 발급 서버로부터 사용자 정보를 등록하고 직접 발급받아 USIM에 저장하는 것이고 (3-1)의 경우에는 라이센스가 저장된 USIM을 다른 안드로이드 폰에 탑재하는 방법으로 사용자의 단말기 이동성을 보장하는 방법이다. 본 논문에서는 이와 같은 모든 방식들에 대한 것이 보장되고, 안전한 라이센스 발급 및 관리가 가능하도록 설계하였다.

3.2 License File 구조 설계

본 논문에서 제안하는 라이센스 파일 구조는 그림 5와 같다. 먼저 MF 아래 USIM ADF가 존재하고, 기존의 USIM 환경에 문제를 발생시키지 않기 위하여 USIM ADF 아래 LICENSE DF를 설계 하위에 관련된 파일을 저장하는 구조를 설계하였다.

LICENSE DF의 FID는 기존 USIM 내에 존재하는 같은 Level의 다른 DF의 FID와 동일해선 안 된다[8,11,12]. 그러므로 USIM 표준 스펙에 따른 FID의 확인 후 동일하지 않은 '5FF0'로 FID를 설계하였고, 이 LICENSE DF 하위에는 APP_MAN EF를 제외한 EF에는 실제 안드로이드 플랫폼에 탑재될 어플리케

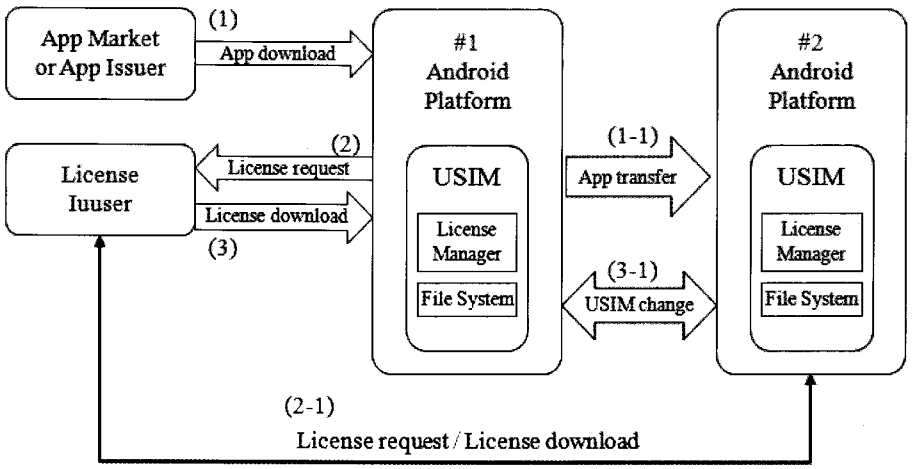


그림 4. 제안 방식의 설계

이션들의 라이센스 데이터가 저장 및 관리된다. 각각의 파일들은 기본적으로 Linear Fixed Record 타입으로 설계하였다. 그리고 이 시스템의 핵심적인 부분은, 각각의 어플리케이션은 파일의 FID를 고정적으로 가지고 유지하지 할 수 없다는 것이다. 왜냐하면, 마켓에서 구매하는 모든 어플리케이션에 고유한 FID를 할당하는 것은 불필요한 낭비이기 때문이다. 사용자가 원하는 어플리케이션을 다운로드 하고, 해당하는 어플리케이션의 라이센스만 USIM 내에 저장하고 관리하면 되는 것이다. 그러므로 유동적인 FID를

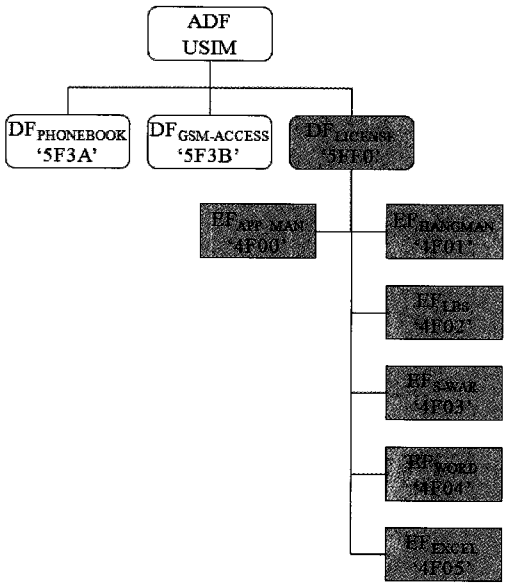


그림 5. 제안 시스템의 License File 구조

유지 및 관리하기 위해서는 해당하는 FID 정보를 동적으로 유지 관리할 필요성이 있다. 이러한 운영이 가능하도록 하는 것이 바로 APP_MAN EF 이다. 이 파일의 경우에는 TLV(Tag, Length, Value) 구조로 설계되어 있다. 이 EF 내의 정보는 Tag의 경우에는 A0로 각각의 파일을 구분하고, Length의 경우에는 뒤에 따라오는 Value의 길이를 지정하며, Value에는 파일명과 FID가 저장된다. FID는 16진수 4자리로 구성되어 있어 해당되는 길이의 하위 2byte만 취하면 바로 해당하는 파일의 FID가 되고, 나머지 상위 byte가 어플리케이션의 이름이 된다. 이와 같이 어플리케이션이 추가되고, 라이센스를 발급받게 되면, 먼저 APP_MAN EF내에 추가될 파일의 정보를 기록하고, 이 정보와 동일한 EF 파일을 생성하게 된다. 그리고 이후, 라이센스 발급 절차에 의해 정해진 대로, 라이센스를 발급 받아 저장 및 관리하게 된다.

3.3 인증 프로토콜 설계

본 논문에서 제안하는 방식에서의 라이센스 인증은 크게 두 가지로 이루어져 있다. 외부와의 통신이 불가능한 환경에서의 내부 유심(USIM)과 사용자의 정보만으로 인증이 가능한 기본인증과 라이센스 인증 서버와의 통신이 가능한 환경에서 인증서버와의 통신에 의해 인증을 수행하는 외부 인증 방식이다. 그리고 이장에서는 라이센스 관리의 효율성을 위하여 효율적인 라이센스 데이터 구조를 정의하고 인증 방식에 대해 기술한다.

3.3.1 라이센스 데이터의 구조

모바일 환경에서의 어플리케이션의 특징은 데스크 탑보다 어플리케이션의 생명 주기가 짧다. 보통 2년 정도면 휴대폰을 교체하거나, 새로운 모바일 어플리케이션을 다운받거나 하기 때문이다. 그러므로 어플리케이션을 영구적으로 구매하는 것 보다, 기간 단위로 어플리케이션을 구매하는 것이 더욱더 효율적일 수 있다. 이러한 기간 단위의 구매 정보는 라이센스 데이터 내에 저장되어 있어야 한다. 그리고 휴대폰을 분실할 경우도 고려하여야 한다. 휴대폰을 분실할 경우, 라이센스 데이터 역시도 함께 잃어버리기 때문에, 해당하는 라이센스를 다시 발급 받아야 한다. 이와 같이 재발급 받을 횟수 역시도 구매 시 고려하여 구매하고, 이 정보가 라이센스 데이터에 저장되어야 한다. 그리고 라이센스는 구매자에 종속되어야 하므로, 구매자의 정보가 라이센스 데이터에 저장되어야 한다. 마지막으로, 구매되는 어플리케이션에 라이센스 데이터는 종속되므로, 어플리케이션의 정보 역시 라이센스 데이터에 저장되어야 한다. 이러한 기본 조건들을 만족시키는 구조로 라이센스 데이터인 LDATA와 LADATA를 아래 와 같이 설계 되었다.

표 1에서 보는 것과 같이 라이센스 발급을 위해서는 라이센스 만료 기간과, 재발급 횟수, 그리고 사용자의 패스워드 마지막으로 발급 받고자 하는 어플리케이션의 ID가 필수적이다. 라이센스 데이터는 LDATA와 LADATA가 있다. 각각의 데이터의 내용은 아래와 같다.

- $LIC\_exp=h(exp||x)\oplus exp$
- $LIC\_cou=h(counter||x)\oplus counter$
- $ILDATA=h(LIC\_exp||LIC\_cou||PWi||AppID)$
- $LDATA=ILDATA\oplus h(IDi||PWi)$
- $LADATA=ILDATA\oplus h(AppID)$

표 1. 라이센스 발급 항목에 대한 설명

항 목	설 명
exp	라이센스 만료기간
counter	라이센스 재발급횟수
x	LRC의 고유비밀 정보
IDi	i번째 사용자의 라이센스 발급서버의 ID
PWi	i번째 사용자의 라이센스 발급서버의 패스워드
AppID	어플리케이션의 ID

이 LDATA와 LADATA를 통해서 해당하는 사용자의 라이센스인지, 그리고 발급되는 라이센스가 어떤 어플리케이션의 것인지를 확인 및 검증 가능하도록 설계 되었다.

3.3.2 라이센스 발급 Phase

라이센스의 발급 절차에서는 어떻게 라이센스 내에 포함되는 정보가 노출되지 않을 것인가가 중요하다. 이를 위해 라이센스 발급 및 등록 센터인 LRC의 공개키로 사용자의 IDi, PWi를 암호화 하여 송신한다. 자세한 통신방식은 다음과 같다.

Step 1 : 안드로이드 플랫폼에서는 USIM에 IDi와 PWi를 보내면 USIM은 난수 Ni를 생성하여 LRC의 공개키로 $E_{pukLRC}(IDi,PWi,Ni)$를 수행하여 LRC로 송신한다.

Step 2 : LRC는 수신된 정보를 개인키로 복호화 하여 IDi와 PWi 그리고 Ni를 획득한다.

Step 3 : 사용자는 라이센스의 만료시간 LIC_exp와 라이센스의 재발급 횟수 LIC_cou를 선택한다. 이후 LRC는 ILDATA와 LDATA,LADATA를 생성하고, (LDATA||LADATA)⊕Ni를 수행하여 USIM에 송신, USIM은 해당하는 파일의 정보를 통해 FID를 생성하고 LDATA와 LADATA를 저장한다.

3.3.3 기본인증 Phase

기본인증은 무선인터넷이 동작하지 않는 환경에서의 인증 방법으로, 사용자의 정보와 어플리케이션의 정보가 USIM에 넘겨지고 USIM 내에서 정보를 노출시키지 않고, 라이센스 데이터를 사용하여 인증을 수행하는 것이다. 자세한 통신은 다음과 같다.

Step 1 : 사용자는 IDi, PWi, AppID를 USIM에 송신한다.

Step 2 : USIM은 ILDATA'=h(IDi||PWi)⊕LDATA, ILDATA''=h(AppID)⊕LADATA를 생성한다. 그리고 ILDATA'=?ILDATA''을 비교하여 동일한지를 검증한다.

Setp 3 : 비교 검증 결과를 AppID와 함께 안드로이드에 송신한다.

3.3.4 외부인증 Phase

외부인증은 무선 인터넷이 되는 경우에 라이센스

에 대한 검증을 수행하는 것으로, 강한 인증이라고 할 수 있다. 이때 통신상에 인증과 관련된 정보가 노출되어서는 안 된다. 자세한 통신은 다음과 같다.

Step 1 : 사용자는 IDi, PWi, AppID를 USIM에 송신한다.

Step 2 : USIM은 ILDATA'=h(IDi||PWi)⊕LDATA, ILDATA''=h(AppID)⊕LADATA를 생성한다. 그리고 ILDATA'=?ILDATA''을 비교하여 동일한지를 검증한다.

Step 3 : USIM은 난수 Ni를 생성한다. 이 난수를 기본으로 하여 SLDi=ILDATA⊕Ni를 생성한다.

Step 4 : LRC의 공개키로 사용자의 IDi, PWi 그리고 SLDi를 암호화 하여 LRC로 송신한다.

Step 5 : LRC는 개인키로 복호화 하여 수신한 IDi와 PWi로 ILDATA를 생성한다.

Step 6 : Ni'=SLDi⊕ILDATA를 통해 Ni'을 획득하고 Ni'⊕h(AppID)를 USIM으로 송신한다.

Step 7 : USIM에서는 검증할 AppID를 해시함수에 넣고 (Ni'⊕h(AppID))⊕h(AppID)를 수행하여 Ni'을 획득한다. 그리고 Ni'=?Ni를 확인하여 안드로이드에 AppID와 함께 검증결과를 송신한다.

4. 구현 및 테스트

제안 방식의 구현은 크게 USIM 부분과 안드로이드, 그리고 라이센스를 발급하는 LRC로 구분할 수 있다. USIM 부분에서는 파일 시스템과 인증을 위한 인증 API, 그리고 애플릿을 구현하였고, 안드로이드 부분에서는 네트워크 통신을 수행하고, 정상적인 인증이 완료되면 이에 대한 정보를 표시하도록 구현하였으며, LRC의 경우에는 사용자로부터 넘겨오는 정보에 따라 라이센스를 발급하고, 인증에 따른 정보를 생성하도록 구현하였다. 그리고 테스트 환경으로는 1대의 호스트 내에서 루프백 주소를 이용하여 LRC 서버와 어플리케이션 그리고 USIM과 통신하는 APDU 클라이언트로 구성하여 테스트 하였으며, 유·무선지역에 대한 판단은 변수 값에 의해 조절하도록 구현하였다.

4.1 USIM File System 구현

USIM 파일 시스템은 라이센스 데이터를 저장하고 관리 할 수 있도록 하여야 한다. 더불어 기존의 USIM의 역할을 방해해서도 안 되고, 데이터의 교환에 의한 손실도 있어서도 안 된다. 이러한 조건에 따라 USIM 표준문서에 정의되어 있는 FID를 제외한 FID를 기본으로 하여 작성하여야 한다[8,11]. 이는 그림 6에서는 보는 것과 같이 정의하였다.

이후 추가되는 라이센스에 대해서는 기본 4F00의 APP_MAN EF에 기록한 뒤에 해당하는 파일의 정보를 저장 할 수 있도록 애플릿에 구현하였다.

4.2 인증 프로토콜의 동작 방식

인증 프로토콜의 경우에는 라이센스 데이터의 발급, 기본인증과 외부인증까지의 포함한 내용에 대해 개별적으로 동작을 수행할 수 있도록 설계 및 구현하였다. 이에 대한 USIM 상의 흐름도는 그림 7에서 보는 것과 같다.

먼저 APDU프로토콜을 통해 수신된 IDi와 PWi 그리고 AppID를 수신하고 이 AppID가 현재 존재하는파일인지를 검증하여 기본인증과 외부인증 중에 선택하여 인증을 수행하도록 한다. 이때 인증의 성공은 인증 프로토콜에서 설계한 것처럼 AppID와 '9000'을 연접하여 송신하고 '6FF0'의 경우에는 기본인증의 실패를 '6FF1'의 경우에는 외부인증의 실패를, 그리고 마지막으로 '6FF2'의 송신은 어플리케이

```
//License Management File System FID List - Yun-Seok Lee //100926
private final static short LIC_FILE          = (short)0x5FF0;
private final static short APP_MAN           = (short)0x4F00;
private final static short HANGMAN           = (short)0x4F01;
private final static short LBS               = (short)0x4F02;
private final static short L_WAR             = (short)0x4F03;
private final static short WORD              = (short)0x4F04;
private final static short EXCEL             = (short)0x4F05;
```

그림 6. USIM 내의 애플릿에 정의된 라이센스 전용 파일의 FID

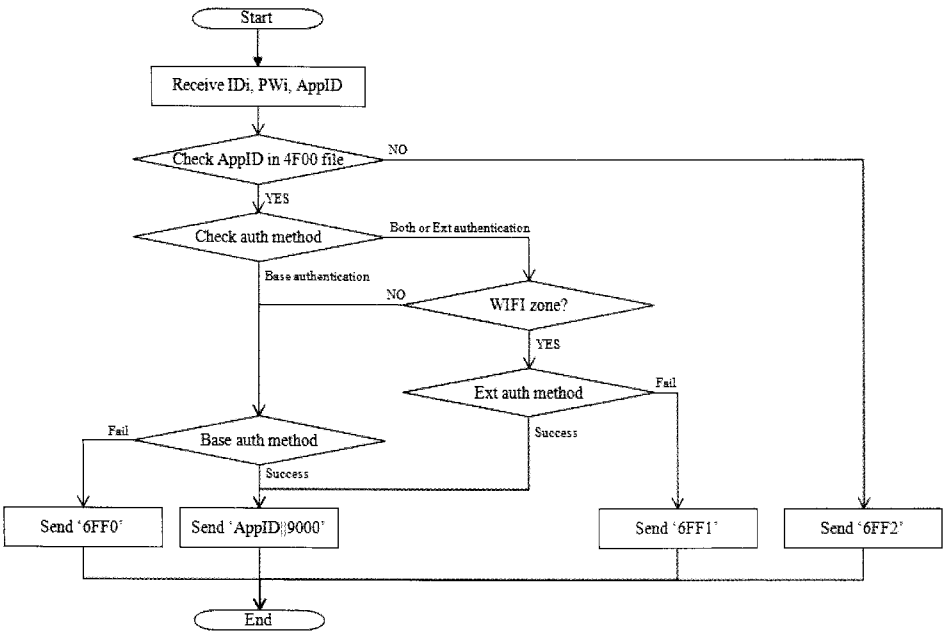


그림 7. USIM 내의 라이센스 에플릿의 동작

션 매니저 파일 내에 인증하려는 어플리케이션의 ID인 AppID가 존재하지 않을 경우 송신하도록 구현하였다.

4.3 안드로이드 부분

안드로이드 부분의 경우에는 USIM 내에서 넘겨받은 인증 정보에 따라 라이센스를 넘겨줄 것인지에 대한 결정을 수행한다. 이때 테스트를 위해, 사용자로부터 IDi와 PWi를 입력 받도록 하였으며, 해당하는 AppID의 경우에는 테스트 어플리케이션에 하드코딩 해 놓았다. 이와 같은 방식으로 인증 결과에 대한 정보가 넘겨오면 그에 따라 인증완료와 다음화면으로 넘어가는 형태로 간단히 구현하였다.

5. 분 석

본 논문에서 제안하는 안드로이드 플랫폼에서의 라이센스 관리 방법에 대한 분석은 안전성과 효율성 측면에서 분석하였다. 안전성 측면의 경우에는 공격방법에 따라 해당하는 안전성을 증명하였고, 효율성 측면에서는 기존 LVL 방식과 OMA DRM 방식에 비해 라이센스 관리 및 인증 방법의 효율성을 표 2에 비교 분석하였다.

5.1 안전성 분석

안전성 분석에서는 발급 단계와 외부 인증을 수행하는 경우에서의 노출된 정보에 대한 재연공격이 가능한지를 검증하고, 사용자의 정보와 어플리케이션의 정보를 조합하여 라이센스 데이터를 직접 수정 또는 생성할 수 있는지를 검증하였으며, 마지막으로 발급단계에서의 정보를 토대로 하여 라이센스 데이터에 대한 복제가 가능한지를 검증한다.

5.1.1 재연공격

제안 하는 방식에서 정보의 노출은 발급 단계에서의 (LDATA||LADATA)⊕Ni와 발급 및 인증 단계에서 사용되는 $E_{pukLRC}()$에 의해 암호화된 정보뿐이다. 이때 발급단계에서 공격자가 LRC의 공개키에 의한 암호화된 정보를 획득하더라도, 이후 라이센스를 발급 받을 때, IDi와 PWi를 알 수 없으므로 안전하고, 이 IDi와 PWi를 포함하는 발급 단계에서의 정보는 난수 Ni에 의해 매번 다른 값이 되므로, 발급단계에서의 노출된 정보는 재연공격에 강하다. 그리고 라이센스 데이터 역시 Ni 값을 알지 못하면 LDATA와 LADATA를 알 수 없으므로 안전하고, 발급 단계의 이 데이터 역시 난수 Ni 값에 의해 매번 다르므로 재연공격에 안전하다. 그리고 외부 인증단계에서의

노출되는 정보는 LRC의 공개키에 의해 암호화되는 정보와 그 결과에 해당하는 Ni⊕h(AppID) 이다. 이 때 LRC의 공개키에 의해 암호화 되는 정보의 경우에는 SLDi를 포함하는데, SLDi는 SLDi=ILDATA⊕Ni 이므로 매 세션마다 다른 인증정보가 송신되므로 재연공격에 강하며, 결과의 송신 역시도 난수 Ni에 의해 매 세션마다 다른 값을 가지게 되므로 재연공격에 안전하다.

5.1.2 라이센스 데이터에 대한 직접 수정

라이센스 데이터 LDATA와 LADATA의 기본 구성은 ILDATA이다. ILDATA의 경우에는 h(LIC_exp||LIC_cou||PWi||AppID)이다. 이때 LIC_exp는 LIC_exp=h(exp||x)⊕exp 이고, LIC_cou=h(counter||x)⊕counter이다. 이때 x는 LRC의 노출되지 않는 비밀 고유 값으로 이 x 값을 알지 못하고서는 ILDATA를 생산할 수 없다. 그리고 ILDATA에 대한 획득을 위해서는 LDATA와 LADATA에서 사용자의 ID와 PW 그리고 어플리케이션의 AppID를 획득하여야 하는데, 이때 공격자가 임의의 ID와 PW 그리고 AppID를 이용하여 ILDATA를 생성하려하여도, 사용자의 PWi와 AppID가 ILDATA 내에 함께 해시되어 저장되어 있으므로 동일한 형태의 정보를 생성할 수가 없다.

5.1.3 라이센스 데이터 복제

라이센스 데이터가 노출되는 시점은 등록단계에서의 사용자가 생성한 난수 Ni에 의해 XOR 연산을 거친 결과 단 한번 뿐이다. 이후의 과정에서는 라이센스 데이터를 생성하기 위한 어떠한 정보도 노출되지 않으며, 라이센스 데이터 자체도 노출되지 않는다. 한번 저장된 라이센스 데이터는 USIM 내의 파일 시스템에 안전하게 저장 및 관리되며, 이 정보에 대한 연산은 모두 USIM내에서만 이루어지므로 안드로이드 플랫폼 까지 정보를 교환하지 않아 라이센스를 노출시키지 않는다. 그러므로 라이센스 데이터를 복제하기 위해서는 초기 등록 단계에서의 정보를 획득하여야 한다. 하지만 사용자가 생성한 난수 Ni를 알고 있어야 하고, 이 난수 Ni는 초기 사용자의 등록 단계에서 LRC의 공개키에 의해 안전하게 암호화 되어 LRC에 송신되므로, 공격자가 LRC의 개인키를 알고 있지 않는 한 복제가 불가능하다.

5.2 효율성 분석

본 논문에서 제안하는 라이센스 관리 기법의 효율성 분석은 크게 라이센스 인증에 소요되는 처리 시간의 분석과 제안 라이센스 구조가 가지는 특징에 의한 효율성을 분석한다.

먼저 라이센스 인증에 소요되는 처리시간의 경우에는 기본인증의 경우에는 $2T_{Hash}+2T_{Xor}$ 의 시간이 소요되고, 외부 인증의 경우에는 $T_{RSA}+2T_{Hash}+4T_{Xor}+T_{Random}$ 의 시간이 소요된다. 저전력, 저자원 환경인 USIM과 스마트폰에서는 해당하는 알고리즘의 동작 횟수를 줄이고, 상대적으로 빠른 형태의 암호 알고리즘을 운용하는 것이 효율적이다. 기존의 모바일 환경에서의 3G 인증 방식은 AES를 기반으로 하는 f1,f2,f3,f4함수를 사용하는데 이때 걸리는 시간은 500ms 이내이다[13]. f1,f2,f3,f4함수의 경우에는 OPc를 생산하는 것을 제외하면 $5T_{AES}+4T_{Rot}+13T_{Xor}+T_{Random}$이 걸린다[13]. 이때 T_{AES}는 AES의 동작시간이고 T_{Rot}는 rotate의 동작시간, T_{Xor}은 Xor 연산의 동작시간, 마지막으로 T_{Random}은 난수를 발생하는데 소요되는 시간이다. 그러므로, 제안 인증방식의 인증시간은 3G 환경의 인증 방식에 비해 적은 암호 알고리즘의 운용과, 상대적으로 빠른 해시 함수의 운용으로 실제 USIM내에 탑재되어 운용되어도 500ms 이내에 동작이 완료 될 것이다. 그리고 LVL의 경우에는 마켓 라이센스 서버가 해당하는 라이센스 데이터를 서명하여 송신하고 안드로이드 플랫폼에서는 해당 라이센스 데이터의 서명을 검증하도록 되어 있으므로, T_{RSA} 만큼의 동작시간이 소요된다. 이는 제안 방식에 비해 빠른 처리가 가능하지만, 저장된 라이센스 정보가 노출 될 수 있으므로 안전성에 문제가 있다[7]. 그리고 OMA-DRM의 경우에는 $2T_{RSA}+4T_{AES}+T_{KDF}+T_{MAC}+T_{Random}$ 이 소요된다[14]. 이때 키 생성과 C값 생성에 소요되는 시간은 제외하였다. 이는 제안 방식에 비해 많은 알고리즘 운용 횟수로 인하여, 제안 방식에 비해 느리다.

그리고 제안 라이센스 구조가 가지는 특징에 의한 효율성은 라이센스 데이터에 대한 기간단위의 구매가 가능하여, 자신의 프로그램 활용 패턴에 맞게 적절하게 구매할 수가 있다는 장점이 있다. 그리고 단말기의 분실, 도난 그리고 교체 등에 의해 라이센스 데이터의 손실에 대비하여, 재발급 가능 횟수를 구매 요소로 정의하여 그에 맞게 대처할 수 있게 하였다.

또한 제안방식에서는 LVL 방식에서의 네트워크 환경에서의 인증뿐만 아니라, 상황과, 프로그램의 특성에 따라 기본 인증과 외부 인증을 수행할 수 있도록 설계하여 언제 어느 장소이든 상관없이, 정당하게 구매한 라이센스 데이터로 어플리케이션을 인증 후 사용할 수 있게 하였다. 그리고, 단말기 교체에 따른 어플리케이션의 라이센스 데이터의 재발급에 관해서는 LVL 방식의 경우에는 존재하는 라이센스 데이터를 이동할 방법이 없으므로, 재발급을 받아야 하고, OMA-DRM의 경우에는 동일 도메인으로 정의된 단말기에는 컨텐츠의 이동이 가능하지만, 이후 다른 디바이스의 구매일 경우에는 재발급 받아야 한다[10]. 하지만 제안 방식의 경우에는 라이센스 데이터를 재발급 받을 필요 없이, 사용자의 USIM을 교체함으로서 기존의 발급된 라이센스를 활용할 수 있는 장점이 있다.

6. 결 론

사용자에게 다양한 어플리케이션을 제공하는 모바일 환경에서, 이 어플리케이션에 대한 라이센스는 시장의 보호라는 측면에서 필수적인 부분이다[5,9]. 기존의 어플리케이션에 대한 보호 방식은 OMA-DRM 1.0부터 시작되어 2.0 그리고 구글에서 제공하는 LVL 까지 여러 방식이 고려되고 있으나, 라이센스 데이터의 정보가 단말기에 저장되어 있어 해당하는 정보에 대한 직접적인 공격에 약하고, 사용자 단말의 도난, 분실, 낡은 단말기 등에 따른 단말기의 교체와 같은 것에서는 도메인의 재설정, 라이센스의 재발급 등의 절차를 추가적으로 수행하여야한다. 이와 같은 문제를 해결하기 위하여 본 논문에서는 단말내에 존재하는 USIM 칩 내부에 라이센스 데이터를 안전하게 저장 및 관리하는 파일 시스템을 설계 및 구현하였고, 온·오프라인에 상관없이 안전한 라이센스 인증이 가능하도록 2가지의 인증 프로토콜을 제안하여, 라이센스 데이터에 대한 직접공격을 방어하였다. 그리고 사용자의 단말 교체시 USIM 칩 교환으로 간단하게 라이센스 데이터를 사용할 수 있으며, 기간단위 그리고 재발급 횟수를 라이센스 데이터 정보에 추가 하므로 해서 효율성을 높였다. 이와 같은 이유로 사용자의 권리와 어플리케이션의 시장을 안전하게 보호 할 수 있다.

참 고 문 헌

[1] http://developer.android.com/guide/index.html

[2] http://source.android.com/

[3] Y. Hashimi and S. Komatineni, "Pro Android," BERKELEY, 2009.

[4] F. Ableson, C. Collins, and R. Sen, "Unblocking Android," MANNING, 2009.

[5] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android Security," *IEEE Security & Privacy*, Vol.7, pp. 50-57, 2009

[6] http://developer.android.com/guide/publishing/licensing.html

[7] http://www.readwriteweb.com/archives/android_drm_cracked_pirating_apps_is_easy.php

[8] TTAT.3G-31.102, "IMT-2000 3GPP - Characteristics of the USIM Application(R7)," 3GPP, pp. 6-117

[9] C.Y. Chuang, Y.C. Wang and Y.B. Lin, "Digital Right Management and Software Protection on Android Phones," VTC 2010, pp. 1-5, 2010.

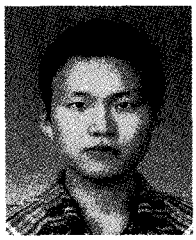
[10] W. Buhse, and J.V.D Meer, "The Open Mobile Alliance Digital Rights Management," *IEEE Signal Processing Magazing*, Vol.24, pp. 140-143, 2007

[11] W. Rankl and W. Effing, "Smart Card Handbook," JOHNWILEY & SONS, 2003.

[12] 이윤석, 전하용, 정민수, "File Cache 및 Direct Access 기능을 추가한 Java Card File Systemd에 관한 연구," 한국멀티미디어학회논문지, 제11권, 제3호, pp. 404-413, 2008.

[13] V.Neimi and K. Nyberg, "UMTS SECURITY," JOHNWILEY & SONS, 2003.

[14] E.J Yoon, J.S Kim, B.H. Cho and K.Y. Yoo, "Efficient OMA-DRM v2.0 ROAP for Protecting a Rights Object for a Device," FGCNS2008, pp. 9-13, 2008.



이 윤 석

2006년 경남대학교 컴퓨터공학부 졸업(공학사)
2008년 경남대학교 컴퓨터공학과 졸업(공학석사)
2010년~현재 경남대학교 컴퓨터공학과 박사 수료

관심분야 : Java Card, Mobile Security, Home Network Security

김 은

2009년 경남대학교 컴퓨터공학부 졸업(공학사)
2010년~현재 경남대학교 첨단공학과 석사과정
관심분야 : Java Technology, Home Network Security, Network Security

정 민 수

1986년 서울대학교 컴퓨터공학과 학사
1988년 한국과학기술원 전산학과 석사
1994년 학국과학기술원 전산학과 박사

1990년~현재 경남대학교 컴퓨터공학부 교수
관심분야 : Java Technology, JavaMachine, Home Networking