情 報 保 護 學 會 誌
第21卷 第1號, 2011. 2

스마트폰 응용 프로그램 개발 및 코드 서명

조 태 남*, 문 남 미**

요 약

최근 스마트폰의 사용자가 국내외적으로 급증하고 있으며 국내 사용자 수도 수백만에 이르고 있다. 스마트폰의 종류는 매우 다양하며 이 장치들이 기반하고 있는 플랫폼도 다양하다. 스마트폰 플랫폼 개발자들은 해당 플랫폼 장치에서 실행될 수 있는 일관된 응용 프로그램 개발을 지원하기 위해 개발 툴을 제공하고, 안전성 제공을 위해 응용 프로그램의 배포 과정을 지원한다. 응용 프로그램이 배포되기 위해서는 사전에 코드 서명이 이루어져야 하는데, 이 코드 서명을 통하여 사용자는 개발자 및 프로그램의 무결성 등을 확인할 수 있다. 본 고에서는 스마트폰 시장에서 주류를 이루고 있는 플랫폼별로 개발 지원 툴과 이를 통한 코드 서명에 대하여 살펴본다.

Ⅰ. 서 론

스마트폰은 전화 기능뿐 아니라 이동성과 함께 인터넷과 멀티미디어를 지원하며, 사용의 편리성으로 인하여 사용자들의 수가 급증하고 있다. 여러 이동통신사와 장치 개발 업체에서 여러 가지 스마트폰이 시판되고 있지만, 이들 장치들의 플랫폼(운영체제)은 몇 가지로 분류된다. 플랫폼 공급자들은 제 3의 개발자들이 개발하는 응용 프로그램들이 일관성 있고 안정적으로 개발되어 장치에서 실행되도록 하기 위해서, 정책을 수립하고 응용 프로그램들이 준수해야 할 기준을 제시하며 통합 개발 툴을 제공한다. 제공되는 툴을 이용하여 프로그램이 개발되고 나면, 이 프로그램은 배포되기 전에 플랫폼 공급자들이 규정한 기준을 준수하는지 검증 받고 서명 받아야 한다. 서명된 응용 프로그램을 다운로드하는 사용자는 프로그램이 적정 수준의 기준을 만족하는지 확인하거나 프로그램 개발자를 확인하거나 혹은 다운로드 받는 프로그램이 배포 당시의 프로그램과 동일하다는 것을 확인할 수 있게 된다.

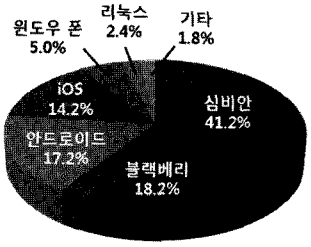
본 고에서는 국내외 스마트폰 마켓을 점유하고 있는 플랫폼들에 대해 알아보고, 각 플랫폼 공급자들이 제공하고 있는 코드 서명에 대해 살펴본다.

Ⅱ. 플랫폼의 시장 점유율

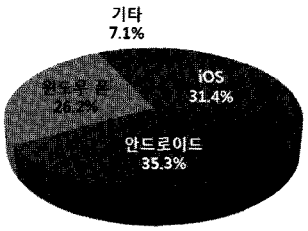
스마트폰 플랫폼의 시장 점유율은 조사 기관에 따라 다른 분석결과를 보이며, 시간에 따라 그 분포도가 변하고 있다. 또한 국내 시장에서의 점유율과 세계 시장에서의 점유율은 크게 다르다. 가트너(Gartner)에서 2010년 9월에 조사한 바에 의하면, 세계 시장에서는 [그림 1]과 같이 심비안(Symbian)이 41.2%로서 압도적인 비율로 1위를 차지하고 있고, 블랙베리(BlackBerry)가 18.2%, 안드로이드(Android)가 17.2%, iOS(iPhone OS)가 14.2%를 차지하고 있다[1]. 한편 애틀러스(Atlas) 리서치앤컨설팅이 조사한 바에 의하면, 2010년 5월 국내 스마트폰 시장 점유율은 [그림 2]와 같이 안드로이드가 35.3%로 1위를 차지하였고, iOS가 31.4%, 그 뒤로 윈도우 폰(Windows Phone)이 26.2%을 차지하고 있다[2].

이와 같이 스마트폰 플랫폼의 국내외 시장 점유율이 다르고, 조사 기관에 따라 점유율이 다르지만, RIM (Research In Motion)에서 개발한 블랙베리, 애플(Apple)에서 개발한 iOS, 구글(Google)에서 개발한 안드로이드 및 노키아(Nokia)에서 개발한 심비안, 마이크로소프트(MicroSoft)에서 개발한 윈도우즈 폰이 주요 플랫폼임을 알 수 있다.

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임 (No.2010-0000487).
* 우석대학교 정보보안학과 (tncho@ws.ac.kr)
** 호서대학교 벤처전문대학원 IT응용기술학과 (mnm@hoseo.edu)



〔그림 1〕 세계 스마트폰 시장 점유율



〔그림 2〕 국내 스마트폰 시장 점유율

Ⅲ. 플랫폼별 코드 서명 기법

심비안, 블랙베리, iOS, 안드로이드, 윈도우 폰 7 등으로 대표되는 스마트폰 플랫폼 공급자들은 각 플랫폼에서 실행될 응용 프로그램 개발을 지원하기 위한 개발 툴을 제공하고, 개발된 응용 프로그램이 마켓에 공개되기 전에 필요한 요구사항이나 기준을 만족하는지 검사하고 인증한다. 또한, 각 플랫폼 응용 개발자를 위해서 개발자를 위한 웹페이지 및 포럼을 운영하고 있다. 웹페이지에서는 개발자 가이드 문서들과 필요한 툴들을 다운로드 받을 수 있으며, 포럼에서는 개발자들이 경험한 문제점 및 해결방법들에 대한 의견 교환이 이루어지고 있다.

각 플랫폼에서는 응용 프로그램에 대한 인증 방법으로서 코드 서명 기법을 이용하고 있다. 각 플랫폼에서는 개발자의 지적 재산권 보호를 위해 코드 서명을 사용하기도 하고, 응용 프로그램이 최소한의 퀄리티를 만족함을 인증하기 위해 사용하거나 특정 API의 사용권한 제어 혹은 응용간의 원활한 통신을 위한 방법으로 사용하기도 하며, 응용 프로그램의 무결성을 검증하기 위한 방법으로 사용하기도 한다. 응용 프로그램이 만족해야 할 기준이나 코드 서명 기법은 응용 프로그램이 기반하고 있는 플랫폼에 따라 다르지만, 코드 서명 과정은 유사하다. 대부분의 플랫폼에서는 개발자가 개발한 응용프로그램에 개발자의 개인키를 이용하여 서명하도록 요구한다. 이를 위해 개발자는 인증서를 요청하고 발급받게 된다. 이 인증서 발급 및 서명은 개발 단계와 배포 단계용으로 구분하도록 하고 있다. 개략적인 개발 단계는 다음과 같다.

1. 개발자 등록
2. 개발 툴 다운로드
3. 개발 환경 설정
4. 공개키-개인키 쌍 생성
5. 인증서 요청 혹은 생성
6. 인증서 다운로드 및 설치
7. 응용 프로그램 개발
8. 응용 프로그램 서명
9. 응용 프로그램 테스트

응용 프로그램의 개발 및 테스트가 완료되면 다음 단계를 거쳐 배포된다.

1. 응용 프로그램을 배포용으로 구축
2. 배포용 인증서 요청 혹은 생성
3. 인증서 다운로드 및 설치
4. 응용 프로그램 서명
5. 응용 프로그램 등록
6. 응용 프로그램 서명 검증
7. 응용 프로그램 배포

위에서 기술한 개발 및 배포 절차는 개략적인 절차로서, 각 플랫폼에서 동일하게 적용되는 것은 아니며 플랫폼에 따라 절차가 약간 다르거나 일부 단계가 지원되지 않기도 한다. 다음 절에서는 각 플랫폼 별로 코드 서명을 위한 세부 단계와 지원 방법에 대해 살펴본다.

3.1 심비안(Symbian)

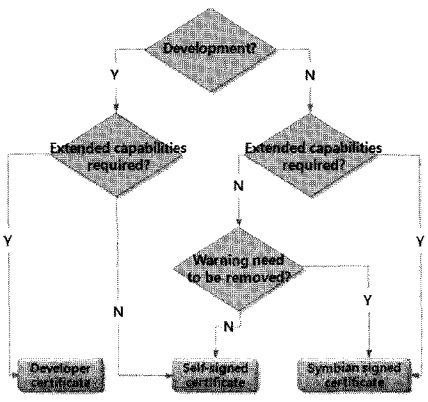
심비안 응용 프로그램을 개발하기 위해서는 노키아가 제공하는 Nokia Qt SDK가 필요하다. 이러한 툴 및 가이드 문서가 심비안 개발자 사이트를[3] 통해 제공되고 있다.

모든 설치파일 .SIS은 심비안 내에서 서명되어야 하는데, 심비안의 SW installer가 이를 점검한다. 심비안에서도 개발용 서명과 배포용 서명이 구분된다. 배포용

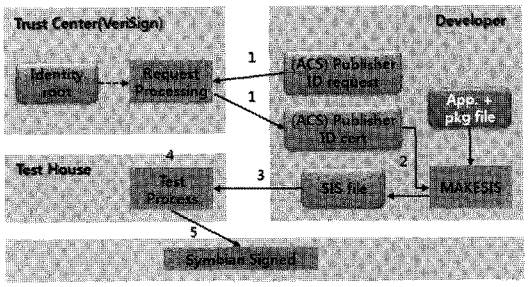
서명의 경우, 배포될 응용 프로그램에 필요한 사용권(capability)에 따라 요구되는 서명의 레벨이 다르다. 사용권에는 단계별로 사용자 사용권(user capabilities), 시스템 사용권(system capabilities), 인증된 서명/배포자 ID(certified signed/publisher ID)와 장치 제조사 사용권(Device manufacturer capabilities)이 있는데, 응용 프로그램이 사용자가 승인할 수 있는 기능(예: 사용자 데이터 접근)만을 사용한 경우, 자가 서명(self-signed)만으로 배포할 수 있다. 하지만 마켓에서 이 프로그램은 비신뢰 응용으로 취급되거나 벤더가 검증하지 않은 응용으로 취급된다. 만약 응용 프로그램이 민감한 시스템 기능을 사용할 경우에는(system capabilities) express 서명을 사용할 수 있다. 이 경우에는 응용이 개발자에 의해 테스트된 경우이며 비신뢰 응용으로 간주된다. 심비안이 인증한 서명(certified signed)을 사용할 경우에는 시스템 기능 뿐 아니라 심비안이 서명한 계정과 배포자 ID(publisher ID)를 사용하는 경우로서 인증된 테스팅 하우스에서 테스트된 것을 의미한다. 비신뢰 응용인 경우에는 장치에서 설치될 때 사용자에게 경고가 안내된다[4][5][6]. 심비안이 인증한 서명을 사용하게 되면 응용 프로그램에 "for Symbian OS" 로고 사용권이 주어진다([그림 3] 참조)[6][7].

심비안이 인증한 서명을 하기 위해서는, 개발자가 계정을 등록하고 배포자 ID를 구입해야 하며 키를 생성하여 인증서를 받아야 한다. 이 단계는 심비안이 허용한 인증기관인 Trust Center를 통하여 이루어진다. 그러나 베리사인(VeriSign)이 발행한 기존의 ACS 배포자 ID도 사용할 수 있다[8].

인증된 서명을 위한 절차는 다음과 같다([그림 4 참조]). (1) 개발자는 TC Trust Center 등록하여 심비안이 서명한 계정을 얻고 배포자 ID를 구입한다. (2) SignSIS 툴을 이용하여 배포자 ID를 가지고 개발한 응용 프로그램에 서명한다. (3) 배포자 ID로 서명한 프로그램의 SIS 파일을 테스트 하우스(Test House)에 제출한다. (4) 테스트 하우스는 제출한 응용 프로그램이 테스트 기준을 만족하는지 점검한다. 이 테스트는 응용 프로그램이 만족해야 할 최소한의 요구 조건을 만족하는지를 검사한다. (5) 테스트에 통과하면 심비안이 응용 프로그램에 서명한다[8][9].



[그림 3] 심비안 사용권별 코드 서명



[그림 4] 심비안 코드 서명 절차

3.2 블랙베리(BlackBerry)

블랙베리 플랫폼에서의 개발자를 위해서도 웹사이트가[10] 있어서 개발툴과 문서 다운로드가 지원되고 개발자 포럼이 운영된다. 블랙베리 기반의 스마트폰 응용 프로그램 개발툴로는 프로그램 통합 개발 툴인 이클립스(Eclipse)용 자바 플러그인(JDE: Java Development Environment)과 시뮬레이터를 제공한다. 블랙베리 개발 회사인 RIM은 보안과 수출 제어를 위해 주요 API들의 사용을 관리하고 모니터링한다. 관리 대상으로는 런타임 API, 블랙베리 응용 API와 블랙베리 암호 API들이다. 만약 개발 응용 프로그램에서 이러한 API나 메소드들을 사용할 경우, 이 프로그램이 스마트폰에 로드되기 전에 RIM이 제공한 서명키로 서명되어 있어야 한다. 여기에 사용되는 암호학적 클래스는 Certicom의 기술을 사용하고 있다[11].

안전한 코드 서명을 위한 툴로서는 SAT(Signing Authority Tools)를 제공한다. 이를 통하여 개발자들의 공개키-개인키 쌍을 생성하고, 개발자들의 데이터베이스를 관리하며 코드 서명 요청을 평가하고 허가한다. API 개발자들의 공개키는 .key 파일로 개발자에게 전

송되고, 개인키는 패스워드로 암호화되어 SAT 안에 저장된다. 응용 프로그램 개발자는 이 .key 파일을 개발하는 응용 프로그램의 프로젝트에 추가한다. 이 서명키는 특정 API에 대한 접근을 제어하는데 사용되기도 하지만, 응용 프로그램이 실행될 때 특정 데이터에 대한 접근을 제어하는데 사용될 수도 있다.

응용 프로그램 개발자가 개발한 코드에 대한 서명을 얻는 절차는 다음과 같다. 개발자는 SAT의 구성 요소인 ST(Signing Tool)을 이용하여 (1) SAT에 등록하고 (2) 공개키-개인키 쌍을 생성한다. (3) 이 때 개발자가 적용할 수 있는 서명에 대한 정보 파일인 .csi가 생성된다. (4) 개발자가 ST를 이용하여 SAT에게 RIM에 의해 관리되고 있는 API 사용을 위한 서명을 요청하면 ST는 .csi를 검증하고 (5) SAT에 서명을 요청한다. 이 요청에는 컴파일된 응용 프로그램에 대한 해쉬값과 개발자의 식별자가 포함되며 응용 프로그램에 대한 어떤 정보도 포함하지 않는다. 이 요청은 개발자의 개인키로 서명되어 전송된다. (6) 요청을 수신한 SAT는 개발자의 서명 등을 검증하고, (7) 해쉬 적용 후 서명을 생성하여 ST에게 전송한다. (8) ST는 수신한 서명을 원본 .cod 파일에 첨부한다. (9) 개발자는 .cod 파일을 장치에 업로드한다. (10) 장치의 VM은 API 라이브러리에 대해 요구되는 서명이 존재하는지 검증한다.

이러한 코드 서명은 저작권 보호를 위한 것일 뿐, RIM은 개발자의 API 사용을 승인하거나 보증하기 위한 것은 아니며 개발자의 사용으로 인한 모든 문제에 대하여 책임지지 않는다.

3.3 안드로이드(Android)

안드로이드 기반의 응용 프로그램 개발자를 위한 웹 사이트가 운영되고 있으며, 이를 통해 개발에 필요한 안내와 툴들을 다운로드 받을 수 있다[12]. 개발 툴로는 ADT(Android Development Tool)을 제공하는데 이것은 프로그램 개발 통합 환경이 이클립스(Eclipse)용 플러그인이다[13]. 안드로이드 기반의 장치에서도 개발자가 서명한 응용 프로그램만이 설치될 수 있다.

응용 프로그램에 개발 및 서명 단계는 [그림 5]와 같다. (1) 포털에서 개발자를 등록하고 (2) 응용 프로그램 개발 환경으로서 자바 JDK, 이클립스, 그리고 ADT를 설치한다. (3) JDK의 표준 툴인 keytool을 이용하여 공

[그림 5] 안드로이드 코드 서명 절차

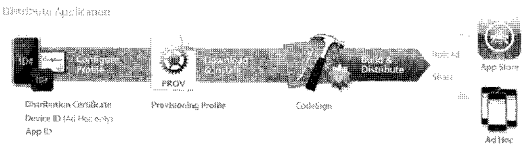
개키-개인키를 생성한다. 키의 크기는 1,024비트가 기본이나 2,048비트를 권장한다. 키 생성 알고리즘으로는 DSA와 RSA 중에서 선택할 수 있으며 기본은 DSA이다. 이 때 공개키에 대한 X.509 v1 형식의 자가 서명 인증서(self-signed certificate)가 생성된다(따라서 CA (Certificate Authority)를 필요로 하지 않는다). (4) Keytool에서는 보호된 형식으로 개인키를 keystore라는 파일 안에 저장한다. 각 개인키는 각 패스워드로 보호되며, keystore도 패스워드로 보호된다. (5) 응용 프로그램을 릴리즈 모드로 컴파일한다. (6) 릴리즈 모드로 컴파일된 응용 프로그램인 .apk 파일에 개인키로 서명한다. 이 때 JDK 표준 툴인 Jarsigner를 사용할 수 있으며 ADT에서는 디버그 모드와 릴리즈 모드의 서명을 지원한다. 만약 키생성에 사용한 알고리즘이 DSA였다면 서명에는 SHA1과 DSA가 사용되며, RSA였다면 MD5와 RSA가 사용된다[14].

안드로이드 응용 프로그램의 코드 서명의 목적은 프로그램의 검증이나 제어에 있지 않다. 응용 프로그램을 일관되게 업그레이드할 수 있도록 하고, 동일한 인증서로 서명된 응용 프로그램들이 동일한 프로세스로 실행될 수 있도록 하며 코드와 데이터를 공유할 수 있도록 하기 위한 것을 목적으로 한다[14].

3.4 iOS(iPhone OS)

iOS를 이용하여 응용 프로그램을 개발하기 위해서는 먼저 iOS 개발자 센터에[15] 방문하여 사용자 등록을 한다. 이미 아이튠즈나 앱스토어를 가지고 있다면 이 아이디를 이용할 수 있다. 개발자 센터에서는 Xcode, iPhone Simulator, Instrument 및 Interface Builder와 같은 여러 가지 개발용 프로그램을 지원한다[16]. 이 중 Xcode는 C, C++, Java 등을 지원하는 응용 프로그램 통합 개발 환경(IDE)으로서 코드 서명에 관련된 작업도 이 프로그램에서 지원한다.

코드 서명은 프로그램 개발과 배포 시에 다른 서명키



〔그림 6〕 iOS 코드 서명 절차

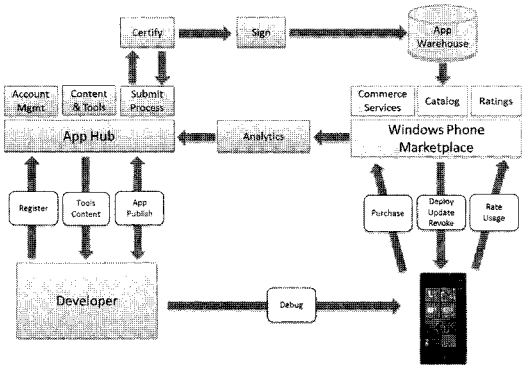
를 사용하여 이루어진다. 코드 서명을 위한 단계는 다음과 같다. (1) 먼저 개발자 프로그램 포털 사이트에서 개발자를 등록한다. (2) 개발 PC의 Xcode에서 KeyChain Access 프로그램 이용하여 개발 서명용 공개키-개인키 쌍 생성한다. 이 때 키의 크기는 2,048비트이며 알고리즘은 RSA이다. 생성된 개인키는 KeyChain에 저장된다. (3) 이 서명키에 대한 인증서를 포털에 요청(업로드)한다(CSR: Certificate Signing Request). (4) 포털에 인증서가 발행되어 게시되면 (5) 인증서를 다운로드하여 KeyChain에 저장한다. 이 인증서를 WWDR Intermediate Certificate라고 하며 신뢰된 인증서로서 개발자 인증서 서명용이다. (6) 포털에서 개발에 사용할 장치를 등록하고 응용 프로그램 식별자 App ID를 생성한 후 DPP(Development Provisioning Profile)를 생성한다. 여기에는 인증서, App ID, 장치 등이 포함된다.

시뮬레이터나 테스트 툴을 이용하여 개발된 응용 프로그램에 대한 테스트가 완료하면, 배포를 위한 별도의 코드 서명이 이루어져야 한다. 이 과정은 개발을 위한 코드 서명 절차와 동일하다. 단, 배포용으로 별도의 공개키-개인키 쌍을 새로 생성하여 인증서를 받고, 장치를 등록하지 않으며 배포용 DPP를 생성해야 하고, 배포용으로 응용 프로그램을 컴파일해야 한다. 응용 프로그램을 배포용으로 컴파일하고 나면, .app 파일을 배포용 서명키로 서명하여 마켓에 등록한다([그림 6] 참조)[17].

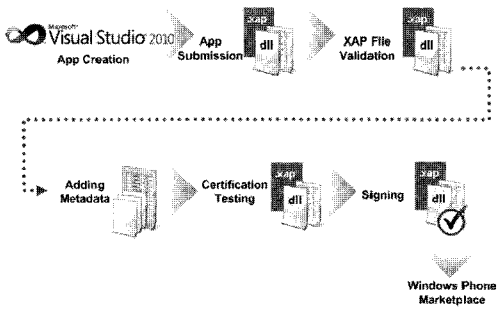
3.5 윈도우즈 폰 7(Windows Phone 7)

윈도우 폰 7 개발자를 위한 사이트로서 App Hub가 [18] 운영된다. 이 사이트를 통하여 개발자는 등록하고 개발 툴을 다운로드하며 개발한 응용 프로그램을 제출한다. 여기서 개발자들은 Visual Studio 통합 개발 환경 툴이나 윈도우 폰 시뮬레이터를 포함하는 툴인 RTW를 [19] 제공받을 수 있다([그림 7] 참조)[20].

제출된 응용 프로그램은 검증되고 서명되어야만 장치에서 수행될 수 있다. 응용 프로그램에 서명받기 위해



〔그림 7〕 Windows Phone 7 응용 프로그램 개발 라이프 사이클



〔그림 8〕 Windows Phone 7 코드 서명 절차

서는 (1) 개발자 포털에서 계정으로 로그인한다. (2) 개발한 .xap 파일을 업로드하면 (3) .xap 파일 검증이 이루어진다. (4) .xap 파일를 repack하고 (5) 메타데이터를 추가한다. (6) 인증서 테스트를 위해 장치에 repacked xap을 설치하여 요구사항을 만족하는지 테스트한다. (7) 테스트를 통과하면 repacked xap 파일들에 서명한다([그림 8] 참조)[21].

.xap 파일 검증 단계에서는 응용 프로그램이 신뢰성, 성능, 자원 관리, 기능, 보안 요구사항을 만족하는지 점검한다[21]. 응용 프로그램이 인증 테스트에 통과되면 자동적으로 응용에 베리사인 인증서를 이용한 코드 서명이 생성된다[22].

Ⅳ. 결 론

본 고에서는 스마트폰 응용 프로그램을 제어하고 인증하기 위한 코드 서명에 대하여 살펴보았다. 코드 서명은 스마트폰이 기반하고 있는 플랫폼의 공급자에 따라 목적과 방법에 차이가 있다.

심비안 응용 프로그램에서는 응용 프로그램이 사용할 수 있는 시스템 기능에 제약을 두기 위한 방법으로 코드 서명을 사용한다. 개발자는 응용 프로그램이 사용하는 시스템 기능 레벨에 따라 적절한 서명을 하고 테스팅 하우스에 제출한다. 테스팅 하우스에서는 제출한 응용 프로그램이 적절한 테스트 기준을 만족하는지 점검한다. 심비안이 인증한 서명을 사용하는 경우, 신뢰 응용으로 인정하는 의미를 갖는다. 개발자 서명에 사용되는 배포자 ID는 심비안이 허용한 인증기관인 TC Trust Center나 베리사인에서 발행한 ID이다.

블랙베리에서는 수출 제어를 위해 주요 API들의 사용을 관리하고 모니터링하고 특정 API에대한 접근 제어를 하며 개발자의 저작권 보호를 위해 코드 서명을 사용한다. 개발자의 서명키는 블랙베리 공급자인 RIM이 제공한 개인키이다.

안드로이드에서는 프로그램 검증이나 제어가 아니라 응용 프로그램의 코드와 데이터를 공유하거나 일관된 업데이트를 위해 코드 서명을 사용한다. 개발자는 JDT 표준 툴을 이용하여 코드 서명에 필요한 공개키와 개인키를 생성하고 서명한다. 따라서 별도의 CA는 존재하지 않는다.

iOS에서는 응용 프로그램의 출처 확인을 위해 코드 서명을 사용한다. 개발자는 개발자로 등록하고, 코드 서명에 필요한 키 쌍을 생성하여 CA인 애플로부터 인증서를 발행받는다. 애플에서는 개발자가 제출한 응용 프로그램이, 애플이 발행한 인증서를 사용하여 서명되어 있음을 검증한 후에 마켓에 업로드한다.

Window Phone 7에서는 개발자가 응용 프로그램을 포털에 업로드하면, 응용 프로그램이 신뢰성, 성능, 자원 관리, 기능, 보안 요구사항을 만족하는지 점검한 후, 베리사인 인증서를 이용하여 코드 서명이 이루어진다.

스마트폰에는 사용자의 개인 정보 등 매우 민감한 정보가 들어 있다. 이러한 정보의 유출은 심각한 보안사고를 유발할 수 있다. 코드 서명은 사용자에게 응용 프로그램의 안전성을 확인할 수 있는 근거를 마련해준다[23]. 비록 코드서명은 플랫폼 공급자의 정책에 따라 다르게 지원되고 있지만, 스마트폰 응용 프로그램들의 안전한 배포와 사용을 위해서는 인증된 인증서의 사용과 이에 대한 관리 및 검증 체계에 대한 연구가 필요하다.

참고문헌

[1] http://www.gartner.com/it/page.jsp?id=1421013, 2010.9.

[2] "국내 주간동향 브리핑-국내 스마트폰 시장, 안드로이드 반격 가시화...아이폰 4G가 변수," http://www.arg.co.kr/atlas/client/html/index.html, 2010.6.

[3] http://www.forum.nokia.com.

[4] Nokia, "Qt for Mobile Application Development for Education v1.1," http://www.forum.nokia.com/Develop/Qt/Documentation/.

[5] http://wiki.forum.nokia.com/index.php/MakeSIS.

[6] Nokia, "A guide to Symbian Signed 3rd edition," 2008.3.

[7] http://www.forum.nokia.com/Distribute/Packaging_and_signing.xhtml.

[8] http://www.trustcenter.de.

[9] Nokia, "How do I get my Symbian OS application signed? - A guide to Symbian Signed v2.3," 2005.

[10] http://na.blackberry.com/eng/developers/.

[11] http://us.blackberry.com/developers/javaappdev/codekeys.jsp.

[12] http://developer.android.com/index.html.

[13] http://developer.android.com/guide/publishing/preparing.html.

[14] http://developer.android.com/guide/publishing/app-signing.html.

[15] http://developer.apple.com/devcenter/ios/index.action.

[16] http://developer.apple.com/programs/ios/develop.html.

[17] Apple, "OS Developer Program - Standard Program User Guide for iOS 4 v2.7," http://developer.apple.com/, 2010.9.

[18] http://create.msdn.com/en-US/.

[19] http://www.microsoft.com/downloads/en/details.aspx?FamilyID=04704acf-a63a-4f97-952c-8b51b34b00ce&displaylang=en.

[20] http://msdn.microsoft.com/en-us/library/ff402531(v=VS.92).aspx.

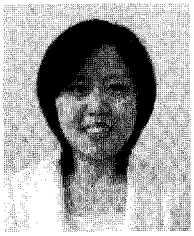
[21] Microsoft, "Windows Phone 7 Application Certification Requirements," http://create.msdn.com/

en-US/SearchResults.aspx?keyword=Application+Certification+Requirements,

[22] Microsoft, "Windows Phone 7 and certificates," http://create.msdn.com/.

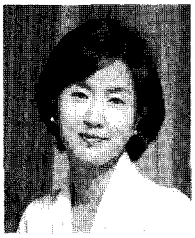
[23] 유재성, 김학현, 최동현, 원동호, 김승주, "스마트폰 어플리케이션의 코드서명," 한국컴퓨터정보학회 하계학술대회 논문집 제18권 제2호, 2010.7.

〈著者紹介〉



조 태 남 (Cho, Taenam)
종신회원

1986년 2월 : 이화여자대학교 전자계산학과 졸업
1988년 2월 : 이화여자대학교 대학원 전자계산학과 석사
1988년 3월~1996년 3월 : 한국전자통신연구원 선임연구원
2004년 2월 : 이화여자대학교 과학기술대학원 컴퓨터학과 박사
2004년 3월~2005년 8월 : 이화여자대학교 컴퓨터학과 전임강사
2006년 4월~2008년 5월 : 한국전자통신연구원 초빙연구원
2005년 9월~현재 : 우석대학교 정보보안학과 조교수
관심분야 : 스마트폰, IPTV, 키관리, 암호프로토콜



문 남 미 (Moon, Nammee)
정회원

1985년 2월 : 이화여자대학교 전자계산학과 졸업
1987년 2월 : 이화여자대학교 대학원 전자계산학과 석사
1998년 2월 : 이화여자대학교 대학원 컴퓨터학과 박사
1999년 ~ 2000년 : 아주대학교 미디어학과 교수
2000년 ~ 2003년 : 이화여자대학교 정보통신연구소 교수/인터넷멀티미디연구센터장
2003년 ~ 2008년 : 서울벤처정보대학원대학교 디지털미디어학과 교수
2008년 ~ 현재 : 호서대학교 벤처전문대학원 IT응용기술학과 교수
관심분야 : 메타데이터, 양방향미디어서비스분석, HCI, T-Commerce, 이러닝, 스마트폰 서비스 기술 모델 등