앱스토어, 디바이스, 제 3 자 개발자간의 인증을 위한 Web 기반의 사설 인증 시스템 구현

강동민*, 장성수*, 최영현*, 박선호*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신공학부
e-mail : {dmkang, ssjang, yhchoi, shpark}@imt.skku.ac.kr, tmchung@ece.skku.ac.kr

A Web Based Private Authentication System for Authentication among App Store, Devices and Third Party Developers

Dong-Min Kang*, Seongsoo Jang*, Young-Hyun Choi*, Seon-Ho Park*, Tai-Myoung Chung**
*Dept. of Electrical and Computer Engineering, Sungkyunkwan University
**School of Information Communication Engineering, Sungkyunkwan University

요 약

스마트폰의 사용자 증가로 인한 스마트폰의 성장과 더불어 제 3 자 개발자들이 개발한 애플리케이션이 오픈 마켓에 등록되면서, 오픈 마켓의 시장 규모가 급속히 증가하고 있다. 하지만 제 3 자 개발자는 보안 관점에서 신뢰 할 수 없는 개체이며, 이들이 개발한 애플리케이션은 악의적 또는 개발자 실수에 의해 보안 문제를 일으킬 수 있다. 따라서 스마트폰과 같은 디바이스와, 앱스토어, 제 3 자 개발자 간에 인증이 요구되고 있다. 이를 위해 본 논문에서는 기존의 사설 인증 시스템에 인증서 소유 검증 모듈과 인증서 유효성 검증 모듈을 추가하고, 시스템의 확장성과 관리를 용이하게 만들기 위해 Web 기반의 사설 인증 시스템을 설계하고 구현한다.

1. 서론

스마트폰의 사용자 증가로 인한 스마트폰의 성장과 더불어 오픈 마켓의 시장 규모가 급속히 증가하고 있다. 2010 년 3 월 구글은 안드로이드 마켓에 등록된 애플리케이션이 최근 3 만개를 돌파 했다고 보고 했으며, 2010 년 4 월 애플은 앱스토어에 22 만개의 애플리케이션이 등록 되어 있다고 전했다. 이처럼 오픈 마켓 시장이 급속하게 증가하는 된 이유는 스마트폰 사용자의 증가뿐만 아니라 많은 애플리케이션이 제 3 자 개발자에 의해 등록되고 있기 때문이다. 그러나 제 3 자 개발자들이 개발한 애플리케이션을 스마트폰에서 실행할 경우, 발생할 수 있는 가장 큰 문제는 보안 문제이다. 제 3 자 개발자는 보안 관점에서 신뢰할 수 없는 개체이며, 이들이 개발한 애플리케이션은 악의적 또는 개발자 실수에 의해 보안 문제를 일으킬 수 있다. 이에 따라 사업자들은 애플리케이션이 앱스토어에 등록되어 구매자에게 전달되기까지 유통자 증명을 제공하는 기술로써, 코드 사이닝(Code Signing) 기술과 자가 서명(Self Signing) 기술을 적용하고 있다. 하지만 자가 서명이나 코드 사이닝을 적용하지 않는 앱스토어가 대부분이다. 그리고 제 3 자 개발자의 애플리케이션 검증을 위해서는 코드에 싸인 하는 키가 필요한데, 대칭키를 사용 할 경우, 애플리케이션을 다운로드 받는 모든 사용자와 1:1 로 대칭키를 설정해야 하므로, 기술을 적용하는데 어려움이 따른다.

이와 같은 문제점으로 스마트폰과 같은 디바이스와 앱스토어, 제 3 자 개발자들에 의해 개발된 애플리케이션 간의 인증이 요구되고 있다.

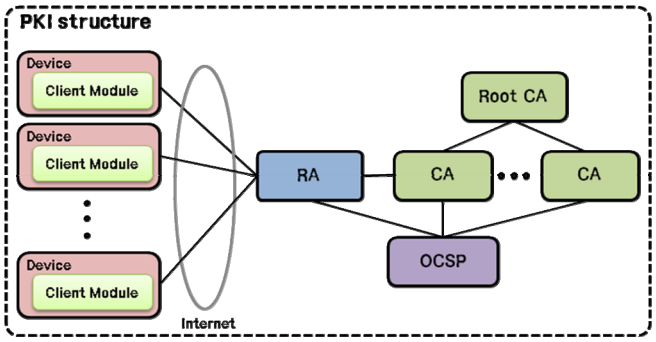
본 논문에서는 디바이스와 앱스토어의 기능을 하는 다운로드 서버, 제 3 자 개발자들을 인증하기 위한 Web 기반의 사설 인증 시스템을 설계하고 구현한다.

본 논문의 구조는 다음과 같다. 2 장에서는 관련연구를 통해 선행 연구를 분석하고, 3 장에서는 제안하는 Web 기반의 사설 인증 시스템의 구조와 기능들을 소개하며, 4 장에서는 제안하는 사설 인증 시스템의 구현물을 소개한다. 마지막으로 5 장에서는 본 논문의 결론과 함께 향후 연구 방향에 대해 기술한다.

2. 관련 연구

2.1. 선행 연구

[강동민 et al., 2010][1] 에서는 디바이스들과 앱스토어 간의 인증을 제공하기 위해 X.509 인증서를 이용하는 공개키 사설 인증 시스템을 설계하고 구현했다. [1] 에서는 공개키 사설 인증 시스템으로 모든 디바이스에게 X.509 인증서를 발급하고, 인증서를 통해 디바이스들과 앱스토어(RA 서버) 간의 인증이 용이하도록 구현했다. [1] 에서 구현한 공개키 사설 인증 시스템의 구성도는 (그림 1)과 같다.



(그림 1) 사설 인증 시스템의 구성도

[1] 에서 구현한 공개키 사설 인증 시스템은 Client Module 을 탑재하고 있는 디바이스, RA, CA, OCSP 로 구성 되며 각각의 기능은 다음과 같다.

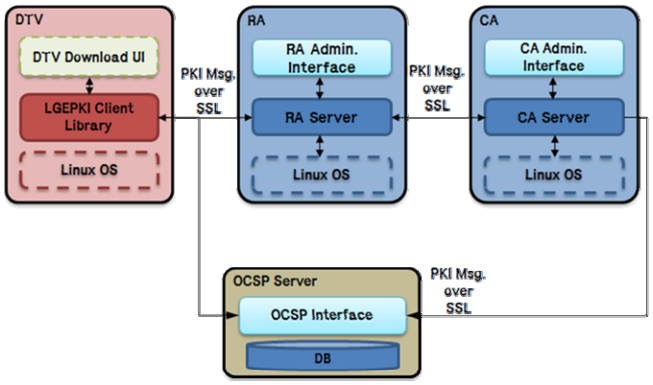
- Client Module : Client Module 은 리눅스 OS 를 기반으로 RA 서버와 통신을 하기 위해 디바이스에 내장 된 시스템이며, 인증서를 생성하기 위해 공개키와 개인키 생성, CSR 생성을 한다. 공개키 인증서 요청 시 디바이스 관련 정보인 UDN(시리얼 번호과 모델 정보)을 SSL 통신을 통해 전송하여 RA 로부터 디바이스를 인증 받는다.
- RA : RA 서버는 RA 인터페이스를 이용하여 디바이스로부터 SSL 통신으로 전송된 CSR 의 유효성과 디바이스 관련 정보인 UDN 과 모델 정보 등을 확인하고, CA 서버에게 인증서 요청, 인증서 연장, 인증서 폐기를 요청하고 처리 결과를 Client Module 에게 전송하는 역할을 수행한다
- CA : CA 서버는 CA 인터페이스를 이용하여 RA 서버로부터 SSL 통신으로 전송된 데이터를 수신받아 헤더를 분석하여 이벤트를 추출하고, 각 이벤트의 요청을 처리한다. CA 서버는 인증서 생성과 인증서 폐기가 주된 기능이며, 인증서 생성 기능은 RA 서버로부터 디바이스나 RA 의 CSR 을 받아 인증서를 생성해주고, 인증서 폐기 기능은 RA 서버로부터 디바이스나 RA 의 인증서를 전송받아, 인증서 확인 후 인증서를 폐기하고, OCSP 서버에게 SSL 통신으로 CRL 을 전송하여 인증서가 폐기 되었음을 알리는 역할을 수행한다.
- OCSP : OCSP 서버는 인증서의 유효성을 실시간으로 알려주기 위해 존재하며, CRL(Certificate Revoke List)과 인증서의 유효성을 저장하는 데이터 베이스(OpenSSL 에서는 index.txt)를 관리하는 역할을 한다. OCSP 서버는 CA 서버들로부터 CRL 과 데이터 베이스를 받고 OCSP 가 가지고 있는 데이터 베이스를 업데이트 한다.

2.2. 선행 연구의 제한점

[1] 에서 구현한 사설 인증 시스템은 X.509 인증서를 사용하므로 디바이스들과 앱스토어 간의 인증이 용이하며, 인증서의 확장 필드에 정책 설정이 가능하므로 관리가 용이하다. 하지만 제 3 자 개발자가 앱스토어에 애플리케이션을 등록할 때, CA 가 발급한 인증서 인지의 여부만을 확인 하고 애플리케이션 등록이 가능하다. 이에 따라 악의적인 제 3 자가 다른 사용자의 인증서로 애플리케이션을 등록할 수 있으므로, 제 3 자 개발자가 개발한 애플리케이션의 검증이 불가능하게 된다. 또한 CUI 환경으로 구현되어 관리자가 관리하기 어려우며, 시스템을 이전하거나 확장할 경우 이식성 및 확장성이 고려되지 않은 문제점을 가진다.

3. Web 기반의 사설 인증 시스템 설계

본 장에서는 선행 연구의 문제점을 해결하기 위해 인증서 소유 검증 모듈과 인증서 유효성 검증 모듈을 추가하고, 시스템의 확장성과 관리를 용이하게 만들기 위해 Web 기반의 공개키 사설 인증 시스템을 구현했다. (그림 2)는 Web 기반의 공개키 사설 인증 시스템의 구조를 보여준다.

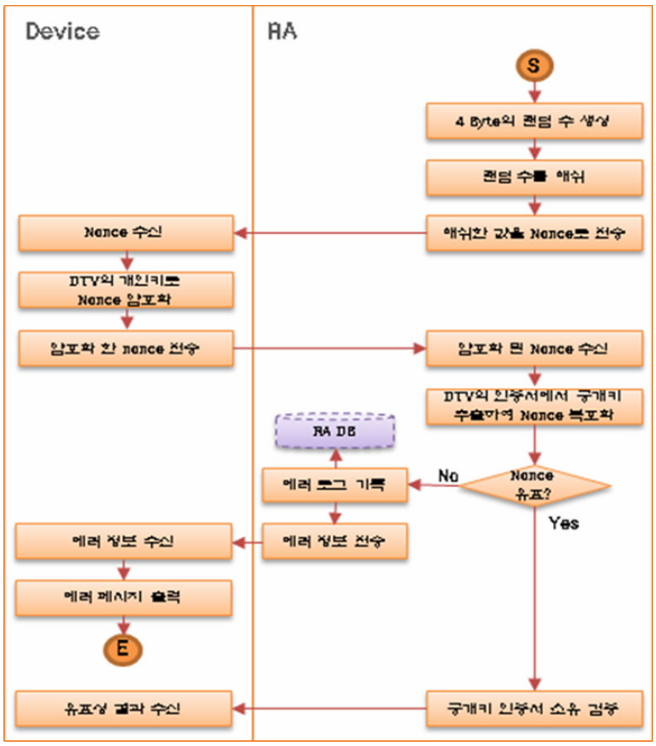


(그림 2) Web 기반의 사설 인증 시스템 구조

3.1. 인증서 소유 검증

인증서 소유 검증 모듈은 제 3 자 개발자가 앱스토어에서 애플리케이션을 등록하거나, 디바이스가 앱스토어에서 애플리케이션을 다운로드 받을 경우, 인증서를 소유하고 있는지 검증하는 모듈이다. (그림 3)은 인증서 소유 검증 모듈의 동작과정을 보여준다.

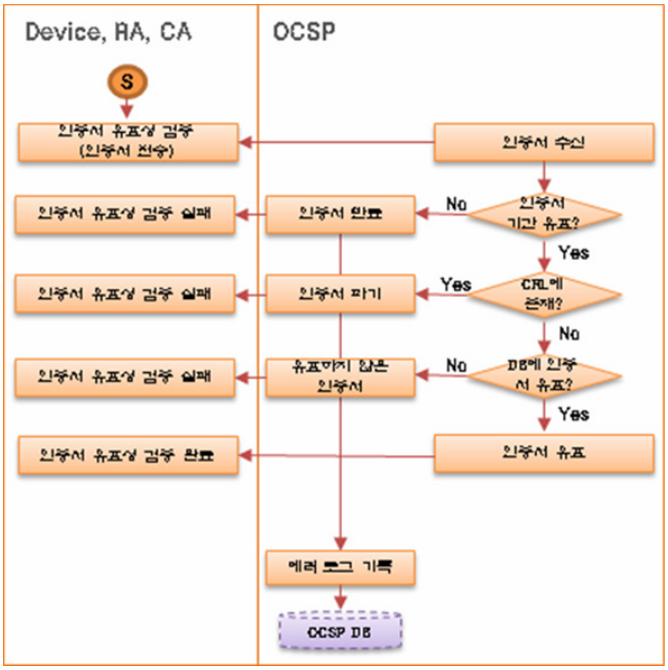
인증서 소유 검증을 위해 앱스토어는 4 Bytes 의 랜덤 한 수를 생성하고, 생성한 값을 해쉬하여, 인증서 소유 검증을 받아야 하는 디바이스나 제 3 자 개발자에게 nonce 로 전송한다. 만약 nonce 를 받은 디바이스나 제 3 자 개발자가 인증서를 소유한 주체가 아니라면 앱스토어로부터 받은 nonce 를 복호화 하지 못하기 때문에 앱스토어에게 받은 4 Bytes 의 랜덤 수를 알아낼 수 없으며, 인증서 소유 검증이 실패하게 된다. 그러므로 제 3 자 개발자가 다른 사용자의 인증서로 악의적인 애플리케이션을 등록 할 수 없으며, 자신의 인증서로 악성코드가 포함된 애플리케이션을 앱스토어에 등록했을 경우에도 탐지 된다.



(그림 3) 인증서 소유 검증 동작과정

3.2. 인증서 유효성 검증

인증서 유효성 검증 모듈은 다수의 CA 가 존재 할 경우 다른 CA 가 생성한 인증서까지도 실시간으로 유효한지 검증하기 위한 모듈이다. (그림 4)는 인증서 유효성 검증 동작과정을 보여준다.



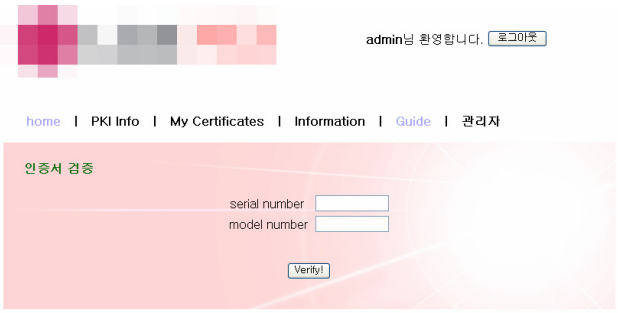
(그림 4) 인증서 유효성 검증 동작과정

4. Web 기반의 사설 인증 시스템 구현

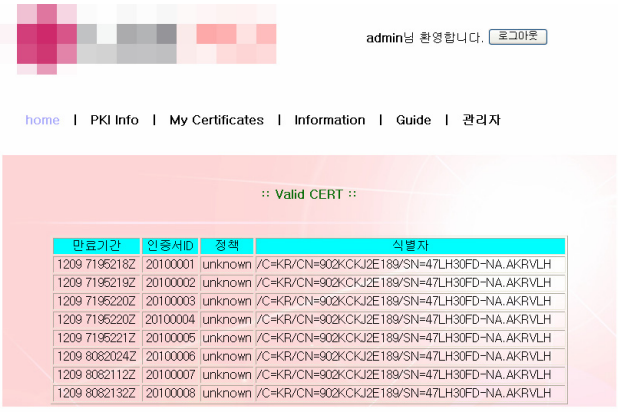
Web 기반의 사설 인증 시스템은 X.509 인증서 포멧, OpenSSL, LibPKI 라이브러리를 사용하여 동작하며, Web UI 는 Apache 로 동작하고, PHP 를 사용하여 구현하였다. Web UI 는 사설 인증 시스템을 이전하거나 확장 할 때, 쉽게 구성이 가능하며, PKI 의 초기 설정, 인증서 관리, 인증서 검증, 로그 관리 등 PKI 의 모든 관리를 UI 에서 할 수 있도록 설계되었다.

(그림 5), (그림 6), (그림 7), (그림 8) , (그림 9) , (그림 10)은 Web 기반의 사설 인증 시스템을 보여준다.

(그림 5) Web 기반의 사설 인증 시스템

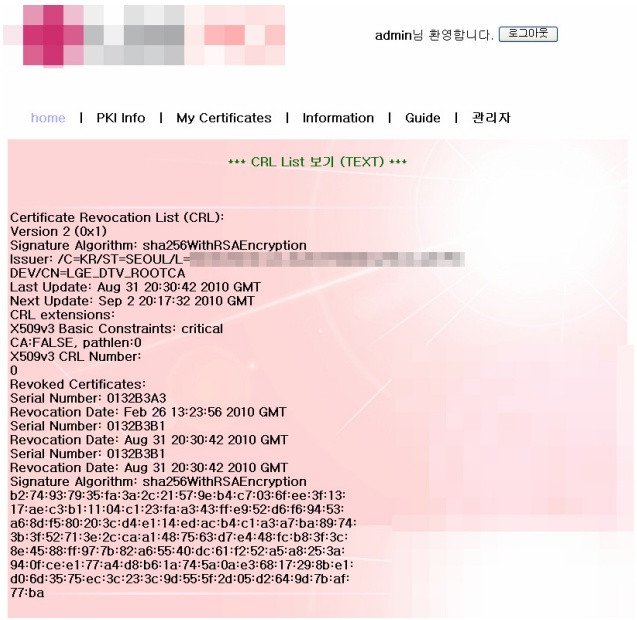


(그림 6) 인증서 검증 페이지



(그림 7) 인증서 관리 페이지

(그림 8) 디바이스의 인증서 생성 페이지



(그림 9) CRL 발행 페이지

(그림 10) 관리자 설정 페이지

5. 결론

본 논문에서는 디바이스와 앱스토어, 제 3 자 개발자간의 인증을 위한 Web 기반의 사설 인증 시스템을 설계하고 구현하였다. Web 기반의 사설 인증 시스템은 악의적인 제 3 자 개발자에 의해 개발된 애플리케이션을 인증하지 못하는 문제를 인증서 소유 검증 모듈과 인증서 유효성 검증 모듈을 구현함으로써 해결하였으며, Web UI 를 통해 PKI 의 확장성과 관리자의 관리를 용이하게 하였다.

향후 연구로는 디바이스들을 PKI 없이 인증하기 위한 UDN(Unique Device Number)을 연구하여, 실제 디바이스에 적용할 수 있도록 연구할 것이다.

참고문헌

[1] 강동민, 박민우, 박선호, 정태명, “앱스토어와 디바이스 간의 인증을 위한 사설 인증 시스템 구현”, 한국통신학회, 2010 년 7 월.

[2] 박선호, 정태명, “스마트 디지털 TV 에서의 제 3 자 개발 어플리케이션을 위한 보안 요구사항 분석”, 한국정보처리학회 논문집 제 17 권 1 호, 2010 년 4 월.

[3] C. Adams, S. Farrell, T. Kause and T. Monomen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol(CMP)”, RFC 4210, Sep. 2005.

[4] http://www.openssl.org

[5] http://www.openca.org