iOS 시스템의 데이터 보호 기법 분석

홍경태, 류재철
충남대학교 컴퓨터공학과
e-mail : kkokdae@gmail.com, jcryou@home.cnu.ac.kr

Analysis of iOS data protection techniques

Gyung-Tae Hong, Jae-Cheol Ryou
Dept of Computer Science and Engineering, Chung-Nam National University

요 약

iOS 시스템의 데이터 보호 기법은 하드웨어에 위치한 암호화 엔진과 계층 키 구조를 사용하여 안전하게 데이터를 보호 하고 있다. 또한, 4가지 클래스로 데이터를 분류하여 데이터를 안전하게 보호 한다. 하지만, 계층 키가 위치한 플래시 메모리를 분석할 경우 데이터 보호 기법이 취약해 질 수 있다는 문제점이 존재한다.

1. 서론

전 세계적으로 스마트폰의 점유율은 기존 피쳐폰(전화, 문자 수신 위주의 폰)의 점유율 보다 높아졌으며, 그중 iOS를 사용하는 스마트폰 및 타블렛은 스마트 디바이스를 판매하는 단일 회사로써 높은 점유율을 보이고 있다.

이러한 모바일 장비 사용자들의 증가에 따라 모바일 환경에서 구동되는 다양한 어플리케이션들이 출시되고 있다. 또한, 그 중에는 사용자의 편의를 위해 중요한 개인 정보를 저장하는 어플리케이션이 점차 늘고 있다. 금융 어플리케이션 같은 경우 공인인증서와 같은 민감한 정보를 저장하고 있고, 사용자의 인터넷 계정 정보와 신용카드 정보를 저장해 관리해주는 어플리케이션도 점차 늘고 있다. 이에 따라 모바일 시스템의 기본 데이터뿐만 아니라 응용 프로그램이 저장하고 있는 정보들도 안전하게 보호해야 하는 필요성이 높아지고 있다.

iOS는 안드로이드OS에 비해 최신 버전 점유율이 높은 것이 특징이다. 2013년 10월에 발표된 안드로이드의 최신 버전 '4.4 킷캣'은 2014년 2월 기준 점유율이 아직 2.5%에 불과하다.[1] 반면 iOS의 2014년 9월에 발표된 최신 버전인 'iOS 7'의 경우 2014년 2월 기준 점유율이 벌써 83%를 넘어선 것으로 발표되었다.[2]

이처럼, 안드로이드에서 발생되는 보안 문제는 최신 버전보다 이전 버전의 보안 기술을 연구하고 문제점을 파악하는 것이 중요하지만, iOS의 경우 최신 버전을 채택하는 속도가 매우 빨라 최신 iOS버전에서의 보안 기술을 빠르게 연구하고 문제점을 파악하는 것이 중요하다.

2. iOS의 데이터 보안

iOS 시스템은 보안 부팅 체인, 코드 서명 및 런타임 프로세스 등 어플리케이션의 안전을 제공하는 다양한 기능이 제공된다. 최신 버전 'iOS 7'을 출시하면서 사용자 데이터를 보호하기 위해 추가로 암호화 및 데이터 보호 기능을 이전 보다 더 강화하였다. 데이터 보호 기능은 개인 및 기업 정보를 안전하게 보호하고 원격에서도 데이터를 제거하기 위한 방법도 제공한다. 본 논문에서는 데이터 보호 기능의 기본이 되는 하드웨어 보안 기능을 알아보고, 최신 데이터 보호 기능을 살펴본다.

가. 하드웨어 보안 기능

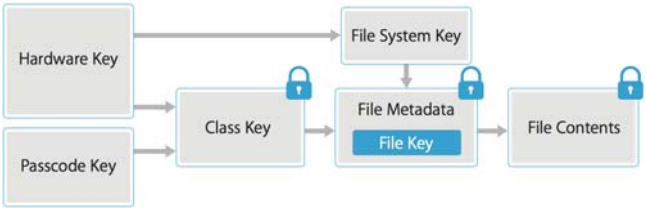
iOS 장치는 하드웨어에 내장된 AES-256 암호화 엔진과 SHA-1 해시 함수 엔진을 사용하여 빠르고 안전하게 데이터 암호화를 제공한다. 이때 사용되는 AES-256 키는 하드웨어에 내장된 UID와 GID를 조합해서 사용한다. UID는 각 장치마다 고유한 랜덤 값이며, Apple이 기록하여 저장하고 있지 않다. GID는 같은 Application Processor를 사용하는 장치에서 공통으로 사용되는 값이다. 이 두 값은 외부에서 접근 할 수 없도록 설계 되었으며 오로지 하드웨어 AES 엔진만이 접근 할 수 있다.

나. 파일 데이터 보호

iOS 시스템에 내장된 하드웨어 암호화 기능 이외에도 iOS 장치의 플래시 메모리에 저장되는 데이터를 보호하는 데이터 보호 기능이 사용된다. 데이터 보호를 위해 키들을 계층 구조로 관리하고, 이런 키들은 하드웨어 암호화 엔진을 통해 생성된다. 그리고, 각 파일들을 클래스로 분류하여 파일 단위로 암호화를 수행한다.

데이터 보호 기능 (그림 1)과 같은 구조를 가진다. 클래스는 4가지로 분류되고, 각 클래스에 해당하는 '클래스 키'가 존재한다. 그리고 파일 시스템마다 존재하는 '파일 시스템

키'와 조합하여 '파일 키'를 생성한다. 이 파일 키는 각 파일의 내용을 암호화 하는데 사용되고, 파일의 메타 데이터 정보에 저장된다.



(그림 1) Architecture overview

클래스는 'Complete Protection', 'Protected Unless Open', 'Protected Until First User Authentication', 'No Protection'와 같이 4가지로 분류된다. 모든 클래스는 기본적으로 암호화를 사용하여 저장하며 데이터 복호화에 사용되는 클래스 키를 유지하고 있는지 혹은 삭제하는지에 따라 클래스를 분류한다.

'Complete Protection' 클래스는 장치가 잠금 상태가 되면 클래스 키를 삭제하여 데이터를 복호화 할 수 없도록 하고, 장치 잠금 상태가 해제되면 클래스 키를 생성하여 데이터에 접근이 가능한 클래스이다. 이에 속하는 데이터는 메일 어플리케이션의 데이터와 어플리케이션 실행 이미지 및 위치 데이터가 있다.

'Protected Unless Open' 클래스는 장치가 잠금 되어 있는 동안에 클래스 키를 유지하여 데이터에 접근 할 수 있는 클래스 이다. 대표적인 예로 백그라운드에서 동작하는 어플리케이션의 데이터가 이에 속한다. 사용자 설정에 의해서 백그라운드에서 구동하도록 어플리케이션을 선택 할 수 있는데, 선택된 어플리케이션의 데이터는 해당 클래스에 속하게 된다.

'Protected Until First User Authentication' 클래스는 장치가 부팅된 이후 처음 잠금 해제를 하기 전 까지는 'Complete Protection' 클래스와 동일하다. 처음 잠금이 해제되어서 클래스 키가 생성되면 다시 장치가 잠금 상태로 전환 되더라도 클래스 키가 유지돼서 데이터에 접근이 가능하다. 기본적으로 타사 어플리케이션의 데이터들이 이에 속한다.

'No Protection' 클래스는 부팅된 이후부터 장치 잠금 여부에 상관없이 계속 클래스 키를 유지하고 있어서 언제든지 데이터 접근이 가능한 클래스이다.

iOS 시스템에서 데이터 보호 기능은 위와 같이 클래스를 분류하여 파일을 보호 하고, 계층 키 구조를 선택하여 파일 데이터를 여러 개의 키를 거쳐 암호화 되어 저장된다.

3. 문제점

앞서 설명한 보호 기능에는 몇 가지 문제점이 존재 한다. 이를 살펴보고 대응 방안을 제시하여 본다.

가. 상위 계층의 키 유출

계층 구조의 키를 사용하여 데이터를 보호하고 있기 때문에 상위 계층의 키가 유출 될 경우 하위 키와 관련된 모든 정보가 유출되는 문제점이 발생할 가능성이 있다.

iOS 시스템의 데이터 보호 기능에서 가장 중요한 2가지 키는 '파일 시스템 키'와 4가지 클래스 마다 존재하는 '클래스 키' 이다. 파일 시스템 키와 클래스 키는 (그림 1)과 같이 하드웨어 키를 사용하여 만들어 진다. 하드웨어 키는 물리적으로 외부에서 접근하기 힘들도록 설계되어 있지만, 이를 사용해 생성되는 2가지 키는 플래시 메모리의 'Effaceable Storage' 라는 특정한 장소에 저장되기 때문에 하드웨어 키를 알아내지 못하더라도, 상대적으로 접근이 쉬운 플래시 메모리를 분석하여 'Complete Protection' 클래스에 해당하는 데이터를 제외한 나머지 클래스 키가 유출될 가능성이 존재한다.

나. 장치 잠금 해제 우회

'iOS 7'이 발표된 이후 많은 취약점이 발생했지만 그 중 장치 잠금 해제를 사용자 암호 없이 가능하게 하는 취약점이 많이 발생하였다.

장치가 잠금 해제된 상태일 경우 모든 클래스 키는 플래시 메모리의 Effaceable Storage에 저장되어 있는 상태이다. 이 경우 앞서 제시된 문제점으로 인해 모든 정보가 유출될 가능성이 존재한다.

4. 결론

위에서 언급한 문제점 중 하나인 잠금 해제 우회 취약점의 경우 보안 패치가 바로 나오고 있지만, 근본적으로 계층형 키 구조의 문제점을 보안하기 위해서는 클래스 키와 같이 중요한 키를 플래시 메모리 내에 저장하는 것이 아니라, 물리적으로 접근하기 힘든 하드웨어 암호화 엔진만 읽기/쓰기 권한을 가진 저장 공간을 확보해서 저장한다면 키 구조의 안전성이 증가할 것이다.

"이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업(No.2010-0020726)과 한국전자통신연구원-차세대통신네트워크산업원천기술사업(No.10043380)의 지원을 받아 수행된 연구임"

참고문헌

[1] Google, Android Developers, http://developer.android.com/about/dashboards
[2] Apple, App Store Distribution, https://developer.apple.com/support/appstore
[3] Apple, "iOS Security", Feb 2014.