ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor & UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 9, September 2017

An Innovative Approach for Host Card Emulation

V. M. Joshi[1], Dhanashri Shinde[2]

Assistant Professor, Dept. of E&TC, PVPIT Engineering College, Pune, Maharashtra, India[1]

PG Student [VLSI& Embedded], Dept. of E&TC, PVPIT Engineering College, Pune, Maharashtra, India[2]

ABSTRACT: With Android version 5.1, UICC can be treated as Secure Element. Android OS version 5.1 introduces a mechanism with which UICC can be accessed with special privileged APIs. Storage on UICC is compatible with 'Global Platform Secure Element Access Control Specifications**'.**

KEYWORDS: NFC, HCE, SE, AID, RFID, UICC Carrier Privileges.

I.INTRODUCTION

Android operating system (OS) v4.4 (KitKat) supports Host Card Emulation. Host Based Card Emulation can co exists with Secure Element. This approach implements Universal Integrated Circuit Card (UICC) usage for secured data storage. This eliminates embedded secure element need. Also provides more secure way for data storage.
Android 5.1 introduced a mechanism to grant special privileges for APIs relevant to the UICCowner's apps. The Android platform loads certificates stored on a UICC and grants permission to apps signed by these certificates to make calls to a handful of special APIs.
Storage on the UICC is compatible with the Global Platform. The application identifier (AID) on card is A00000015141434C00, and the standard GET DATA command is used to fetch rules stored on the card.

II.MOTIVATION

Security is the major concern for Host based Card Emulation approach. Hybrid HCE approach provides more security for storing secure data. Hybrid HCE approach eliminates the need of physical embedded secure element in Smartphone. This saves cost as well as removes dependency of special hardware. With Hybrid HCE, secured data is stored within the Smartphone; this does not require internet connectivity.

III.OBJECTIVE

The objective of paper is to present 'Hybrid HCE' concept. With Hybrid HCE solution Mobile Network Operators (MNO) and issuers can work together. The hybrid solution leverages the assets under the control of both the MNO and issuers.

IV.OVERVIEW

A. UICC Carrier Privilege

Android 5.1 introduced a mechanism to grant special privileges for APIs relevant to the Universal Integrated Circuit Card (UICC) owner's apps. The Android platform loads certificates stored on a UICC and grants permission to apps signed by these certificates to make calls to a handful of special APIs. Since carriers have full control of the UICC, this mechanism provides a secure and flexible way to manage apps from the Mobile Network Operator.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor & UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 9, September 2017

Storage on the UICC is compatible with the GlobalPlatform Secure Element Access Control specification. The application identifier (AID) on card is A00000015141434C00, and the standard GET DATA command is used to fetch rules stored on the card.UICC rules use the following data hierarchy (the two-character letter and number combination in parentheses is the object tag).
Each rule is a REF-AR-DO (E2) and consists of a concatenation of a REF-DO and an AR-DO:

- REF-DO (E1) contains a DeviceAppID-REF-DO or a concatenation of a DeviceAppID-REF-DO and a PKG-REF-DO.
- DeviceAppID-REF-DO (C1) stores the SHA-1 (20 bytes) or SHA-256 (32 bytes) signature of the certificate.
- PKG-REF-DO (CA) is the full package name string defined in manifest, ASCII encoded, max length 127 bytes.
- AR-DO (E3) is extended to include PERM-AR-DO (DB), which is an 8-byte bit mask representing 64 separate permissions.

If PKG-REF-DO is not present, any app signed by the certificate is granted access; otherwise both certificate and package name need to match. [REF 1]

B. Java Card Technology Overview

Java Card refers to a software technology that allows Java-based applications (applets) to be run securely on smart cards and similar small memory footprint devices. Java Card is the tiniest of Java platforms targeted for embedded devices. Java Card gives the user the ability to program the devices and make them application specific. It is widely used in SIM cards (used in GSM mobile phones) and ATM cards. Java card products rely on the Global Platform specifications for the secure management of applications on the card (download, installation, personalization, deletion).
Java Card aims at defining a standard smart card computing environment allowing the same Java Card applet to run on different smart cards, much like a Java applet runs on different computers. As in Java, this is accomplished using the combination of a virtual machine (the Java Card Virtual Machine), and a well-defined runtime library, which largely abstracts the applet from differences between smartcards. Portability remains mitigated by issues of memory size, performance, and runtime support (e.g. for communication protocols or cryptographic algorithms).
At the language level, Java Card is a precise subset of Java: all language constructs of Java Card exist in Java and behave identically. This goes to the point that as part of a standard build cycle, a Java Card program is compiled into a Java class file by a Java compiler; the class file is post-processed by tools specific to the Java Card platform.
However, many Java language features are not supported by Java Card (in particular types char, double, float and long; the transient qualifier; enums; arrays of more than one dimension; finalization; object cloning; threads). Further, some common features of Java are not provided at runtime by many actual smart cards (in particular type int, which is the default type of a Java expression; and garbage collection of objects). [REF 2]

C. Implementation Algorithm

At power on Android OS detects if UICC is present. If yes, it requests data e.g. file system which includes network parameters, Unique Identifiers, Language etc.
Then it checks in ARA-M applet is available on UICC. If yes, it sends Get Data Command. In Response, it receives SHA-1 of the certificate by which Android application is signed. It stores this response.
When NFC reader, sends request for selecting HCE application and asks for token, this application is invoked. SHA-1 of this application and stored SHA-1 is verified. On successful verification, access to UICC is granted.
Whenever Select AID and token request is sent to UICC, data is encrypted and sent back to NFC Reader. This data encryption matches with the data on backend server and transaction is verified OK.

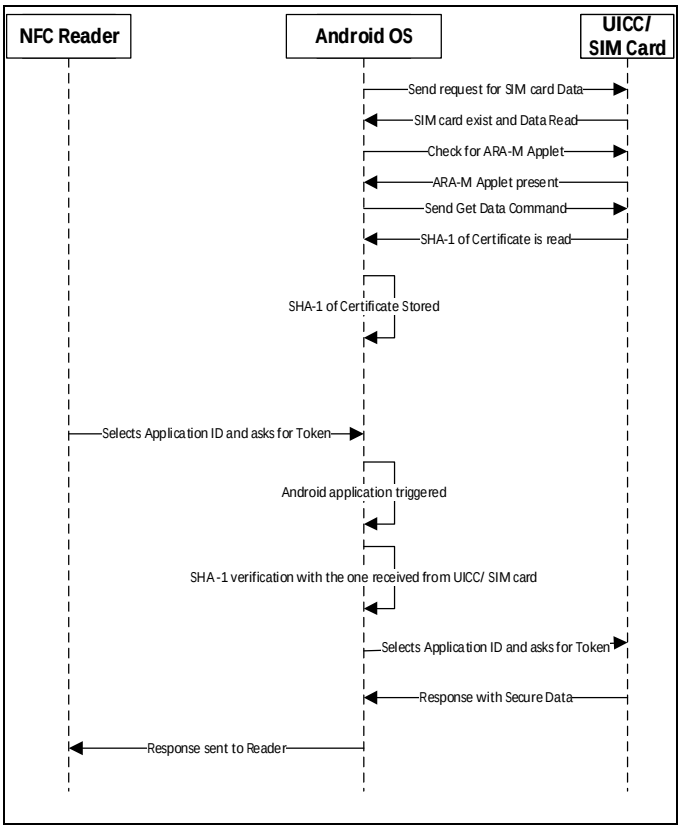
ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor & UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 9, September 2017



V. RESULTS

A. Simulation of Business Logic

1. When Select Applet is received
   >> /send 00A4040007000000000001
   >> 00 A4 04 00 07 00 00 00 00 00 00 01
   << 90 00

2. Example Token Formation
   >> /send 00FE00001011223344556677889900112233445566
   >> 00 FE 00 00 10 11 22 33 44 55 66 77 88 99 00 11 22 33 44 55 66
   << 42 9A 06 1C 5E 34 7B 39 63 0E 2C 2E CA 07 14 BB 90 00

B. Simulation of UICC/ SIM Card

1. When Command for Select applet is received, positive response is sent
   >> /send 00A4040009A00000015141434C00
   >> 00 A4 04 00 09 A0 00 00 01 51 41 43 4C 00
   << 90 00

   - Header: '00 A4 04 00 09'
   - AID: 'A0 00 00 01 51 41 43 4C 00'

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor & UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 9, September 2017

2. When command for Get Data is received, SHA-1 of certificate is sent
   >> /send 00CAFF402A
   >> 00 CA FF 40 2A
   << FF 40 26 E2 24 E1 16 C1 14 1C D7 B6 2B A2 69 05 12 C0 16 91 DA F3 FD 10 A3 B4 31 56 FB E3 0A DB 08 00 00 00 00 00 00 00 00 01 90 00

- Tag: Response-ALL-AR-DO
  FF 40
- Length:
  0x26
- Tag: REF-AR-DO
  0xE2
- Length
  0x24
- Tag: REF-DO
  0xE1
- Length
  0x16
- Tag: DeviceAppID-REF-DO
  0xC1
- Length
  0x14
- Hash of APK signing certificate
  0x1C, 0xD7, 0xB6, 0x2B, 0xA2, 0x69, 0x05, 0x12,0xC0, 0x16, 0x91, 0xDA, 0xF3, 0xFD, 0x10, 0xA3, 0xB4, 0x31, 0x56, 0xFB,
- Tag: AR-DO
  0xE3
- Length
  0x0A
- Tag: PERM-AR-DO
  0xDB
- Length
  0x08
- Bit mask for permissions
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01

VI.CONCLUSION

This demonstration simulates Hybrid HCE with Universal Integrated Circuit Card (UICC). With Java Card applet, it is demonstrated that UICC/ SIM card can be used to store secured data. This has advantage that physical secure element is not required. Advanced Encryption Standard (AES) is used to encrypt the secure data so that it can be verified at server's end.

REFERENCES

[1] UICC Carrier Privileges, [Online], Available: https://source.android.com/devices/tech/config/uicc
[2] GlobalPlatform Secure Element Access Control specification, [Online], https://www.globalplatform.org/specificationsdevice.asp
[3] Mainetti, Patrono, Vergallo, "IDA-Pay: an innovative micro-payment system based on NFC technology for Android mobile devices", Publication Year: 2012, Publication Number: 978-1-4673-2710-7, Publication: IEEE
[4] HCE and SIM Secure Element: It's not black and white, A Discussion Paper from Consult Hyperion Date: June 2014
[5] Wikipedia, [Online], Available: http://en.wikipedia.org/wiki/Near_field_communication

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor & UGC Approved Journal)

Website: www.ijareeie.com

Vol. 6, Issue 9, September 2017

[6] NFC primer for Developers (2013, Sep)[Online], Available : https://supportforums.blackberry.com/t5/Java-Development/NFC-Primer-for-Developers/ta-p/1334857
[7] API Guides, Host-based Card Emulation, [Online], Available: https://developer.android.com/guide/topics/connectivity/nfc/hce.html
[8] ISO 7816 Part 4: Interindustry Commands for Interchange, 2015, Jan, [Online], Available: http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816-4.aspx
[9] NFC applications, [Online], Available: http://www.nfcworld.com/list-of-nfc-trials-pilots-tests-and-commercial-services-around-the-world/
[10] "Google Wallet ends support for physical secure elements," NFC World, Mar. 17, 2014, [Online], Available: http://www.nfcworld.com/2014/03/17/328326/google-wallet-ends-support-physical-secure-elements
[11] "Tim Hortons launches NFC payments service using Host Card Emulation," NFC World, December 13, 2013,[Online], Available: http://www.nfcworld.com/2013/12/13/327339/tim-hortons-launches-nfc-payments-service-using-host-card-emulation
[12] "Visa to Enable Secure, Cloud-Based Mobile Payments," Visa press release, February 19, 2014,[Online], Available: http://investor.visa.com/news/news-details/2014/Visa-to-Enable-Secure-Cloud-Based-Mobile-Payments/default.aspx.
[13] "MasterCard to Use Host Card Emulation (HCE) for NFC-Based Mobile Payments," MasterCard press release, Feb. 19, 2014, [Online], Available: http://newsroom.mastercard.com/press-releases/mastercard-to-use-host-card-emulation-hce-for-nfc-based-mobile-payments/
[14] "BBVA introduces HCE-based mobile payments," Finextra, June 30, 2014, [Online], Available: http://www.finextra.com/news/fullstory.aspx?NewsItemID=26217