GOTrust Technology Inc.

HCE microSD Demo Package User's Guide

Revision 1.1

September, 24, 2014

Table of Contents

Table of Figures

1 Introduction

This document is a manual of GO-Trust Host-based Card Emulation (HCE) microSD demo package. This document introduces a new technique of integrating Android HCE and GO-Trust microSD solution.

2 NFC Mobile Application

Recently mobile devices such as smart phones and tablets are spread over a wide area. A variety of mobile applications on mobile devices also bring significant convenience to people's daily life. However, due to the consideration of security and compatibility of existing ecosystem, the development of mobile commerce is relatively slower than other mobile application areas, especially on the area of mobile payment. People still used to make transactions via the legacy practice, which is taking IC cards as verification front-end and commit payment with traditional bank card payment architecture via NFC readers. Up to now there exists no suitable and powerful mobile payment solution that can successfully and widely meet the requirements both of general people and of bank card ecosystem.

The most crucial and essential problem of driving mobile payment system is on the ownership of secure element since the owner of secure element can dominate both customer services and ecosystem cash flow. Over the past decade many companies have tried to integrate Secure Element (SE) into different components of mobile devices or other peripheral accessories and then construct proprietary mobile payment services. These solutions can be categorized by the ownership of secure element, please refer to Figure 1.

1. **Telecom Operators:** telecom operators propose NFC mobile payment system based on their issued USIMs. Telecom operators' proposal is essentially incompatible with existing bank card ecosystem, and even worse the proposal is conflict to banks' interest, hence this system is not practical for general purpose payment system.
2. **Mobile Devices Provider:** The mobile devices manufacturers embed secure element into their devices and provide NFC services. This solution lacks of cooperation of payment application providers and cannot be a successful proposal, even though the mobile devices providers decide the functionalities and features that end-users can access.
3. **NFC Application Operators:** Application operators integrate secure element into sleeves and provides proprietary NFC applications. This technique separates

hardware limitation and operators integration. However, this proposal can only be deployed on specific use scenario with a few mobile devices.

4. **Banks:** For reasons banks tend to own a secure element that not only is separated from USIM controlling by Telecom operators, but also compatible with user devices, secure microSD is an acceptable alternative. This proposal encounters two problems, firstly not all mobile devices provide microSD interface. And secondly only few mobile devices can support single-wire protocol (SWP) directly communicating with microSD. Even though secure microSD solution fitted in with the expectation of bank card industry; it is not easy to widely deploy secure microSD solution to their customers.

Figure 1. Secure element embedded in different form factor

3 Host-based Card Emulation (HCE)

Recently Google releases a new Android Distribution 4.4, Code name Kitkat, providing a new NFC services called Host-based Card Emulation (HCE). HCE technique implements NFC protocol stack on system level and provide service interface for developers to interact with NFC readers on application layer, which dramatically decrease the cost of NFC service development and deployment. This chapter aims to summarize several important features provided by HCE.

3.1 HCE Protocol Stack implementation

Android Kitkat implements standard NFC protocol stack for application extended from HCE service to communicate with external NFC reader. Please refer to Figure 2. The HCE protocol stack is implemented based on ISO / IEC 14443 Type A while the transmission protocol is based on ISO / IEC 14443-4. Application Protocol Data Unit (APDU) follows the specification of ISO / IEC 7816-4. The HCE protocol stack implementation provide sufficient NFC features for emulating behaviors of most CPU-based secure element.

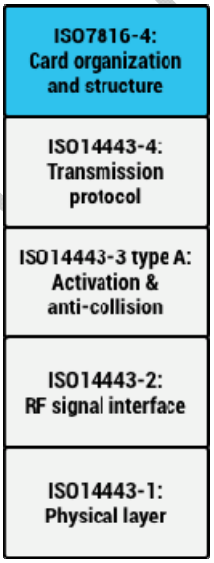

Figure 2. Android's Protocol Stack

3.2 HCE NFC Routing Table and Binding Technique

HCE technique makes NFC Controller relay NFC signal to target location, either host application or secure element, in the state of receiving SELECT AID commands. The goal is achieved by maintaining a routing table in HCE service, which records the mapping of specific AID with its target location. Take Figure 3. For example, if NFC controller receives NFC request SELECT Y from NFC reader where AID Y is located in secure element, NFC controller will look up the routing table and transfer the request to secure element, vice versa.

HCE technique provides single logical channel for NFC transaction. While SELECT command is processed correctly and successfully, Android HCE service ensure the NFC transaction by binding this NFC transaction with target applet located in the Host CPU or secure element. The binding relationship will stay active until NFC transaction is done or NFC controller receives another SELECT AID command.

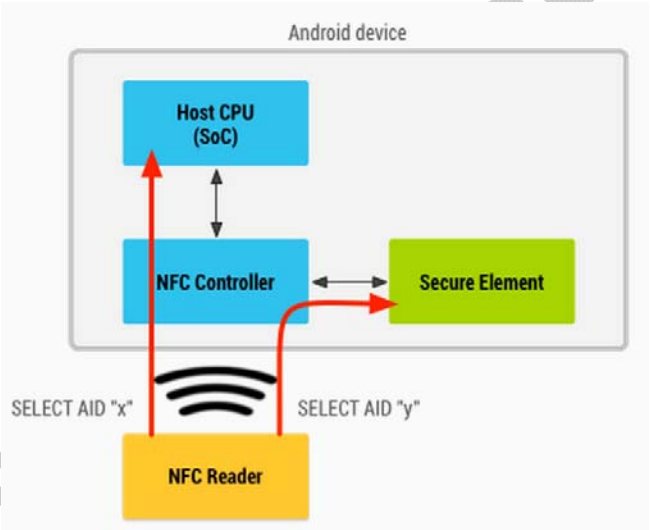

Figure 3. HCE Basic Behavior

3.3 HCE Card Emulation with Secure Element

Android HCE technique provides NFC transaction and binding with target applet either in Host-based applications or secure elements. The NFC signal communication of host-based application is emulated by android HCE service, while the most popular way to make NFC reader communicate with secure element is done by physical wire via Single-wire protocol (SWP). The majority of implementation of mobile devices makes SWP physical wire connecting to USIM and few relays the SWP wire to secure microSD interface.

In the implementation of Android Kitkat, the receiving SELECT AID command is sent to host applications in default setting. If host contains no application to deal with this AID, the signal will then be sent to secure element. Or NFC service providers can register the AIDs as OffHostApduService in the secure element to NFC routing table, in this case NFC controller can identifies the registered AIDs and conducts correct secure element relay

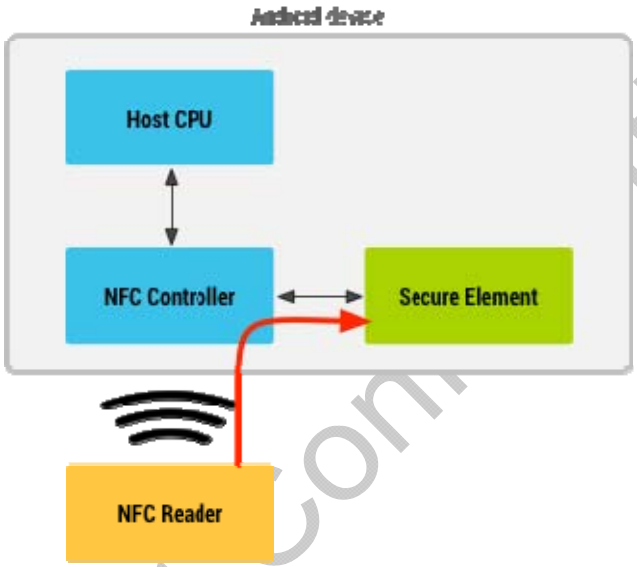

Figure 4. HCE Behavior with Secure Element via SWP

3.4 Host-based Card Emulation

The major feature provided by Android HCE is to provide convenient way for service provider to develop their own NFC applications; moreover, the application can be fast and widely deployed to end users' mobile devices. Please refer to Google Developers Website for detailed programming techniques.

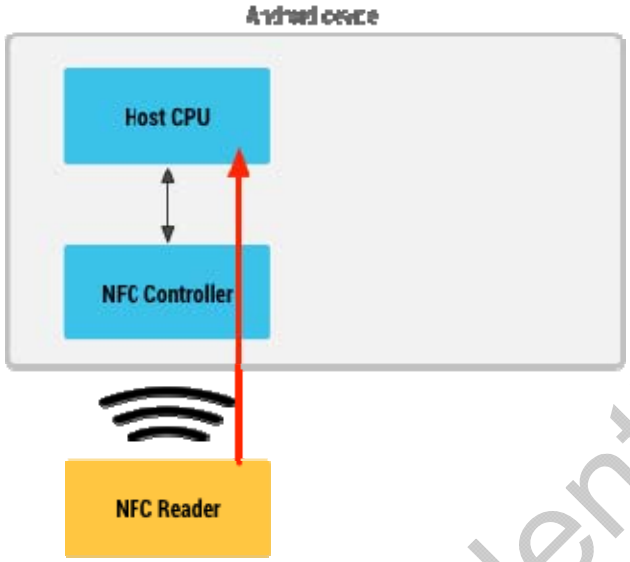

Figure 5. HCE Behavior with Host Application

3.5 Summary

Android HCE Technique possesses significant advantage both in NFC applications development and in services deployment. However, host-based card applications are vulnerable to system-wise attacks while comparing to hardware based solution such as secure elements. Taking payment system for example, lacking of secure storage for secure keys restricts the deployment of HCE technique.

4 GO-Trust HCE with Secure microSD

Although HCE shows some security flaws that make it hard to practical deployment, the new software technique of processing NFC signal via the mobile application brings new a new thought for secure microSD to conduct NFC transaction without the help of SWP.

In traditional practice hardware SWP interface is usually connected to USIM of mobile devices, instead of secure microSD. This hardware limitation dramatically keeps secure microSD from widely deployment, even though secure microSD solution possesses the advantage of being compatible of on-going bank card ecosystem. The appearance of HCE technique exactly provides a gateway for the communication between secure microSD and NFC reader.

This kind of new technique brings four advantages to NFC payment system. Firstly, the limitation of mobile device without SWP connection to secure microSD can still use secure microSD as secure element, which increases the number of target devices that can be used for mobile payment. Secondly, legacy android devices can also support HCE with secure microSD by upgrading their operation system. Thirdly, HCE with secure microSD solution provides a solution to original HCE technique without secure storage for secure keys. Lastly and most important, HCE with secure microSD provides an excellent platform for bank industry to deploy mobile payment application based on existing bank card system architecture.

This chapter aims to introduce GO-Trust HCE secure microSD solution, including security policy, HCE application design and secure microSD design.

4.1 Security Features

The security features of GO-Trust HCE secure microSD are listed as follows.

1. HCE application must provide a unique binding relation between secure microSD and NFC reader.
2. HCE application must not store or cache any information about security parameters (secret keys) protected by secure microSD.
3. HCE application must not store transaction information outside the allocated application memory spaces.
4. HCE application must ensure the integrity of NFC transaction.
5. HCE application must not process any cryptographic computation relating to security parameters protected by secure microSD.

4.2 HCE - Secure microSD System Architecture

Figure 6. describes system architecture of GO-Trust HCE secure microSD. To conduct communication between secure microSD and NFC reader, one special tailored HCE application is required. GO-Trust HCE application is constructed based on two components; one is HostApduService provided by HCE service and the other one is the proprietary GO-Trust SDIO service. HostApduService is responsible for the transaction binding between NFC reader and HCE application, while GO-Trust SDIO service ensures the communication between HCE application and secure microSD. In this way GO-Trust HCE application construct a unique binding between NFC reader and target applet in the secure microSD. Moreover, SWP is not required in this system design.

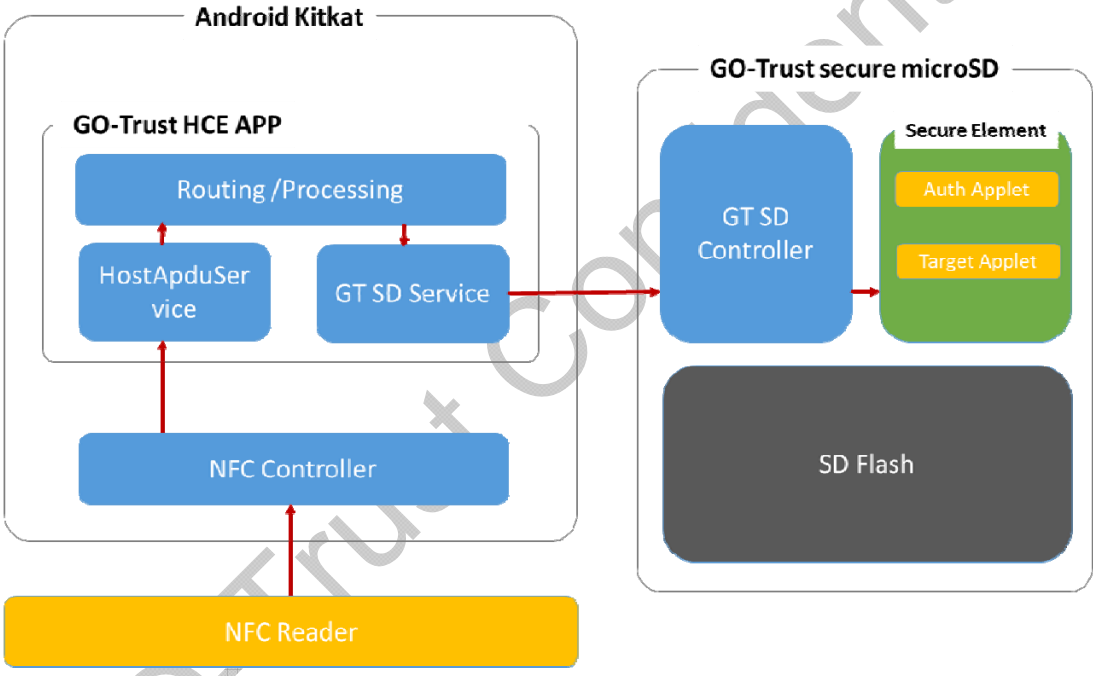

Figure 6. HCE - Secure microSD System Architecture

5 GO-Trust HCE Demo App

GO-Trust HCE App with source code is included in demo package. Developer can take use of this demo app to customize HCE microSD applications.

GO-Trust provides an easy-to-use library for developer to implement HCE microSD apps. Figure 7 shows the library components and the relationship between secure microSD, NFC reader, and HCE app.

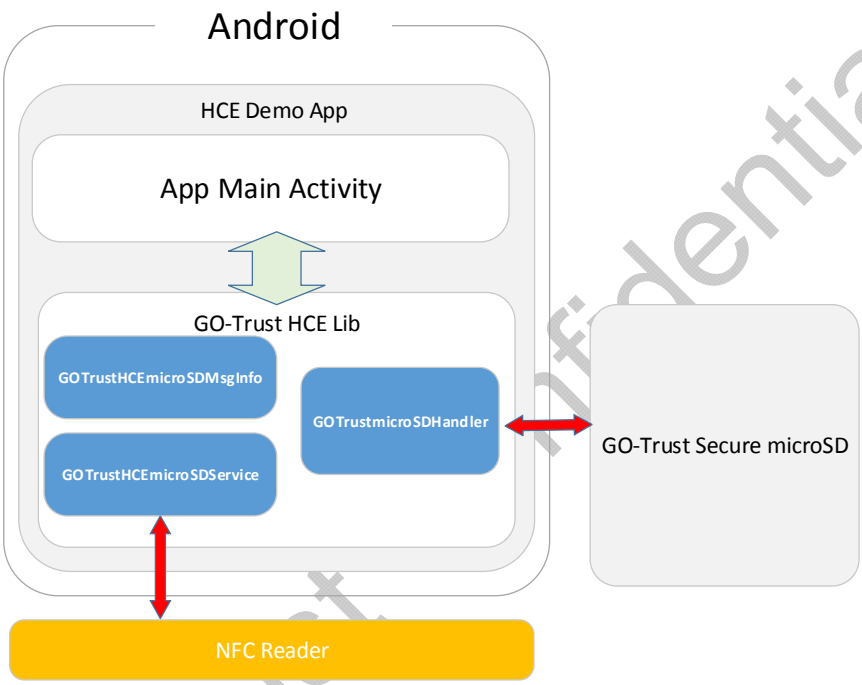

Figure 7. GO-Trust HCE Library Components

Figure 8 shows the outlook of HCE demo app. There are 5 components. Functions of each component is listed as following.

Component 1) Enable GO-Trust HCE microSD:

a、 Enable or disable APDU log (by checking component 5) to method
void com.gotrust.hce.GOTrustmicroSDHandler.setLogEnable(...)

b、 Connect GO-Trust HCE microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler. connectHCEmicroSD ("SMART_IO.CRD")

c、 Enable GO-Trust microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler.setSDEnable(true)

d、 Enable polling microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler.enablemicroSDPolling(0)
Purpose of polling microSD is to prevent OS to power off microSD.

Component 2) Disable GO-Trust microSD:

a、 Disable polling microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler.disablemicroSDPolling()

b、 Disconnect HCE microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler.disconnectHCEmicroSD()

c、 Disable GO-Trust microSD via method
void com.gotrust.hce.GOTrustmicroSDHandler.setSDEnable(false)

Component 3) Create IO File for Kitkat: Android Kitkat version adds some restriction to access microSD. App has to create a large IO file to make GO-Trust working. Developer can refer more detail in Chap 2.4 of SESDAPI User's Guide

Component 4) Auto Activate microSD: When this checkbox is enabled, the app will set com.gotrust.hce.GOTrustmicroSDHandler. `setAutoActivateMicroSD(true)`. HCE microSD will be activated when NFC reader sending select Applet command with registered AID to mobile.

Component 5) Enable/Disable APDU Log: This checkbox is used in Component 1). To show APDU log and save it in app specific folder in SD card.

Component 6) APDU Log Component: Demo App inherits class **`GOTrustHCEmicroSDMsgInfo`** and implement method **`onMessage`** to print logs.

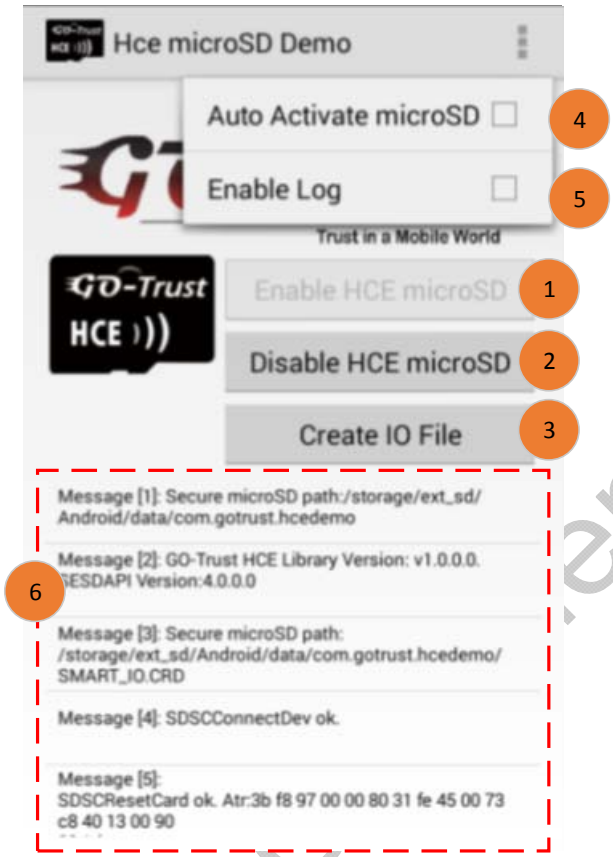

Figure 8. Outlook of HCE Demo App

5.1 Register AID

Each HCE app has to bind specific AID which is described in chapter 3.4 and 3.5. In HCE demo app, developer can add new binding AID in "res\xml\apduservice.xml