QPOS SDK Integration Guide

Document Version：<2.7>

Version	Date	Author	Description
1.0	2013.07.12	Longhui.xiao	Initial added
1.1	2013.12.05	Junan Zhang	
2.0	2014.01.05	Longhui.xiao	
2.1	2014.02.12	Wang Xu	Minor bug fix
2.2	2014.03.11	Longhui Xiao	Add icc apdu
2.3	2014.03.26	Longhui Xiao	Add set the sleep time interface
2.4	2014.04.08	Longhui Xiao	Add update work key interface
2.5	2015.02.13	Qingfu Zeng	Perfect the documents
2.6	2015.03.16	Longhui Xiao	Review modify the document
2.7	2015.09.17	Longhui Xiao	Added resetPosStatus interface. Perfect the documents
2.8	2022.08.23	Ziwei Wang	Add an Example

Copyright notification

notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent - or other industrial or intellectual property rights.

Table

1 INTRODUCTION......8

1.1 SUMMARY......8

1.2 CONNECTION MAP FOR QPOS......9
1.3 CONNECTION MAP FOR QPOS READER......10
1.4 PURPOSE AND SCOPE......10
1.5 GLOSSARY AND DEFINITIONS......10
1.6 TRANSACTION KEY......11
1.7 EMVCO TERMINAL TYPE APPROVAL......11
1.8 MESSAGE FORMAT......11
1.9 EMV STANDARD TAGS......11
1.10 PROPRIETARY TAGS DESCRIPTION......11
1.11 BLUETOOTH MODE......12
1.11.1 HOW TO GET QPOS/SPOS BLUETOOTH ID......12
1.11.2 HOW TO CONNECT QPOS/SPOS THROUGH BLUETOOTH......12
1.12 MANAGEMENT KEYS FOR QPOS......13
1.13 FIRMWARE UPGRADE FOR QPOS......13

2 IOS SDK API......14

2.1 SYSTEM REQUIREMENT......14

2.2 SDK FRAMEWORK......15

2.3 XCODEPROJECT SETTINGS......16

2.4 SDK METHODS......16

2.4.1 COMMON METHODS......16
2.4.2 TRANSACTION RELATED METHODS......17
2.4.3 DELEGATE METHODS......19

2.5 TRANSACTION FLOW......21

2.5.1 EMV ICC TRANSACTION FLOW FOR QPOS......22
2.5.2 EMV ICC TRANSACTION FLOW FOR QPOS READER......23
2.5.3 EMV ICC TRANSACTION DESCRIPTION......24
2.5.4 MAGNETIC TRANSACTION FLOW......27
2.5.5 MAGNETIC TRANSACTION DESCRIPTION......27

2.6 API METHODS REFERENCE......29

2.6.1 GETSDKVERSION......29
2.6.2 RESETPOS......29
2.6.3 STARTAUDIO......29
2.6.4 STOPAUDIO......30
2.6.5 CONNECTBT......30
2.6.6 DISCONNECTBT......30

2.6.7 ISQPOSPRESENT 31
2.6.8 GETQPOSID 31
2.6.9 GETQPOSINFO 31
2.6.10 SETAMOUNT 32
2.6.11 CANCELSETAMOUNT 32
2.6.12 DOTRADE 33
2.6.13 DOCHECKCARD 33
2.6.14 DOEMVAPP 34
2.6.15 SETAMOUNT 34
2.6.16 SELECTEMVAPP 35
2.6.17 CANCELSELECTEMVAPP 35
2.6.18 FINALCONFIRM 35
2.6.19 SENDONLINEPROCESSRESULT 35
2.6.20 ISSERVERCONNECTED 36
2.6.21 SENDTIME 36
2.6.22 SETAMOUNTICON 37
2.6.23 GETPIN 37
2.6.24 POWERONICC 38
2.6.25 POWEROFFICC 38
2.6.26 SENDAPDU 38
2.6.27 SETPOSSLEEPTIME 38
2.6.28 UPDATEEMVCONFIG 39
2.6.29 READEMVAPPCONFIG 39
2.6.30 READEMVCAPKCONFIG 39
2.6.31 UDPATEWORKKEY 40
2.6.32 MACKEYENCRYPT 41
2.6.33 ISQPOSPRESENT 41
2.6.34 SETMASTERKEY 41
2.6.35 CALCMACSINGLE 42
2.6.36 CALCMACDOUBLE 42
2.6.37 CALCMACSINGLENOCHECK 42
2.6.38 CALCMACDOUBLENOCHECK 43
2.6.39 DOWNLOADRSAPUBLICKEY 43
2.6.40 UPDATEMASTERKEYRANDOM 44
2.6.41 UPDATEMASTERKEY 44
2.6.42 PINKEY_TDES 45
2.6.43 PINKEY_TDESNOCHECK 45
2.6.44 SETSYSTEMDATETIME 46
2.6.45 SETMERCHANTID 46
2.6.46 SETTERMINALID 46
2.6.47 GETMAGNETICTRACKPLAINTEXT 47
2.6.48 GETCARDNO 47
2.6.49 GETICCCARDNO 47
2.6.50 POWEROFFNFC 48

2.6.51 sendApduByNFC 48
2.6.52 powerOnNFC 48
2.6.53 cbc_mac 49
2.6.54 cbc_macNoCheck 49
2.6.55 inquireECQAmount 50
2.6.56 isIdle 50
2.6.57 anlysEmvIccData 50
2.6.58 VIPOSBatchSendAPDU 51
2.6.59 synVIPOSBatchSendAPDU 51
2.6.60 anlysEmvIccData_qf 51
2.6.61 iccCashBack 52
2.6.62 setPosPresent 52
2.6.63 setCardTradeMode 52
2.6.64 setPinPadFlag 53
2.6.65 qposStatus 53
2.6.66 readBusinessCard 53
2.6.67 writeBusinessCard 54
2.6.68 syncReadBusinessCard 55
2.6.69 syncWriteBusinessCard 55
2.6.70 confirmAmount 56
2.6.71 setAmount 56
2.6.72 getPin 57
2.6.73 doUpdateIPEKOperation 57
2.6.74 updateIPEKOperationByKeyType 58
2.6.75 doSetBuzzerOperation 59
2.6.76 updateEMVRID 59
2.6.77 updateAID 60
2.6.78 setAIDwithBool 60
2.6.79 setOnlineTime 60
2.6.80 getUpdateCheckValueBlock 61
2.6.81 doSetShutDownTime 61
2.6.82 getBluetoothState 61
2.6.83 setBTAutoDetecting 62
2.6.84 getConnectedPeripheral 62
2.6.85 connectBluetoothByCBPeripheral 62
2.6.86 connectBluetoothNoScan 62
2.6.87 getConnectedDevices 63
2.6.88 cancelTrade 63
2.6.89 asynresetPosStatus 63
2.6.90 setDoTradeMode 64
2.6.91 setFormatID 64
2.6.92 setAmountPoint 64
2.6.93 doTrade_QF 64
2.6.94 setIsQuickEMV 65

2.6.95 getQuickEMV 65
2.6.96 getNFCBatchData 65
2.6.97 powerOffNFC 66
2.6.98 sendPinEntryResult 66
2.6.99 cancelPinEntry 66
2.6.100 bypassPinEntry 66
2.6.101 updateEMVConfigByXml 67
2.6.102 updateEmvAPPByTlv 67
2.6.103 updateEmvCAPKByTlv 68
2.6.104 updateEmvAPP 69
2.6.105 updateEmvCAPK 69
2.6.106 lcdShowCustomDisplay 70
2.6.107 lcdShowCloseDisplay 70
2.6.108 getUpdateProgress 70
2.6.109 updatePosFirmware 70
2.6.110 syncGetPin 71
2.6.111 getICCTag 71
2.6.112 customInputDisplay 72
2.6.113 isCardExist 72
2.6.114 isCardExistInOnlineProcess 72
2.6.115 cbc_mac_cn_all 73
2.6.116 getKsn 73
2.6.117 getKeyCheckValue 73
2.6.118 setBuzzerStatus 74
2.6.119 setAESKey 74
2.6.120 getAESTransmissionKey 74
2.6.121 getShutDownTime 75
2.6.122 setPanStatus 75
2.6.123 getDevicePublicKey 75
2.6.124 setShutDownTimeOnConnected 76
2.6.125 getShutDownTimeOnConnected 76
2.6.126 getCvmPinTryLimit 76
2.6.127 updateKeyByTR_31 77
2.6.128 generateSessionKeys 77
2.6.129 updateRSA 77
2.6.130 getEncryptData 78
2.6.131 getMPUCardInfo 78
2.6.132 getMIccCardData 78
2.6.133 updateKeyByTR_31VersionD 79
2.6.134 powerOnFelica 79
2.6.135 powerOffFelica 79
2.6.136 sendApduByFelica 80
2.6.137 generateTransportKey 80
2.6.138 sendCVV 80

2.6.139 GETENCRYPTEDDATABLOCK 80
2.6.140 SETISSAVELOG 81
2.6.141 DOTRADELOGOPERATION 81
2.6.142 OPERATELEDBYTYPE 81
2.6.143 PLAYBUZZERBYTYPE 82
2.6.144 POLLONMIFARECARD 83
2.6.145 AUTHENTICATEMIFARECARD 83
2.6.146 OPERATEMIFARECARDDATA 83
2.6.147 READMIFARECARD 84
2.6.148 WRITEMIFARECARD 84
2.6.149 FINISHCARD 85
2.6.150 SETISOPERATEMIFARE 85
2.6.151 SETISSUPPORTCLSSELECTEMVAPP 85
2.6.152 GETENCRYPTDATADICT 86
2.6.153 SENDCVMPIN 86
2.6.154 GETCVMKEYLIST 86

2.7 DELEGATE METHODS REFERENCE 87

2.7.1 ONREQUESTWAITINGUSER 87
2.7.2 ONQPOSIDRESULT 87
2.7.3 ONQPOSINFORESULT 87
2.7.4 ONDOTRADERESULT 88
2.7.5 ONREQUESTSETAMOUNT 88
2.7.6 ONREQUESTSELECTEMVAPP 88
2.7.7 ONREQUESTISSERVERCONNECTED 89
2.7.8 ONREQUESTFINALCONFIRM 89
2.7.9 ONREQUESTONLINEPROCESS 89
2.7.10 ONREQUESTTIME 90
2.7.11 ONREQUESTTRANSACTIONRESULT 90
2.7.12 ONREQUESTTRANSACTIONLOG 90
2.7.13 ONREQUESTBATCHDATA 91
2.7.14 ONREQUESTQPOSCONNECTED 91
2.7.15 ONREQUESTQPOSDISCONNECTED 91
2.7.16 ONREQUESTNOQPOSDETECTED 92
2.7.17 ONERROR 92
2.7.18 ONREQUESTDISPLAY 92
2.7.19 ONREQUESTUPDATEWORKKEYRESULT 93
2.7.20 ONGETCARDNORESULT 93
2.7.21 ONRETURNREVERSALDATA 93
2.7.22 ONRETURNGETPINRESULT 93
2.7.23 ONRETURNPOWERONICCRESULT 94
2.7.24 ONRETURNPOWEROFFICCRESULT 94
2.7.25 ONRETURNAPDURESULT 95
2.7.26 ONRETURNSETSLEEPTIMERESULT 95
2.7.27 ONREQUESTCALCULATEMAC 96

2.7.28 onReturnCustomConfigResult 96
2.7.29 onReturnSetMasterKeyResult 96
2.7.30 onReturnBatchSendAPDUResult 97
2.7.31 onReturniccCashBack 97
2.7.32 onUpdatePosFirmwareResult 97
2.7.33 onReturnDownloadRsaPublicKey 98
2.7.34 onPinKey_TDES_Result 98
2.7.35 onUpdateMasterKeyResult 98
2.7.36 onEmvICCExceptionData 99
2.7.37 onReturnUpdateEMVRIDResult 99
2.7.38 onLcdShowCustomDisplay 99
2.7.39 onGetKeyCheckValue 100
2.7.40 onGetShutDownTime 100
2.7.41 onGetDevicePublicKey 100
2.7.42 onReturnUpdateKeyByTR_31Result 100
2.7.43 onQposGenerateSessionKeysResult 101
2.7.44 onReturnGetEncryptDataResult 101
2.7.45 onReturnPowerOnFelicaResult 101
2.7.46 onReturnPowerOffFelicaResult 102
2.7.47 onReturnSendApduFelicaResult 102
2.7.48 onAsyncResetPosStatus 102
2.7.49 onReturnUpdateEMVResult 103
2.7.50 onReturnGetEMVListResult 103
2.7.51 onReturnBuzzerStatusResult 103
2.7.52 onReturnSetAESResult 103
2.7.53 onReturnAESTransmissonKeyResult 104
2.7.54 onReturnSetConnectedShutDownTimeResult 104
2.7.55 onReturnGetConnectedShutDownTimeResult 104
2.7.56 onDoSetRsaPublicKey 105

1 Introduction

1.1 Summary

QPOS is a mobile payment card reader device with pinpad that works with mobile devices such as smart phone. It provides merchants and consumers a safe and convenient way to make mobile payments. QPOS can communicate with the mobile device through many methods, such as: audio jack, Bluetooth and USB cable.

QPOS Reader is another mobile payment card reader device without pinpad. QPOS Reader can communicate with the mobile device through audio jack or USB cable.

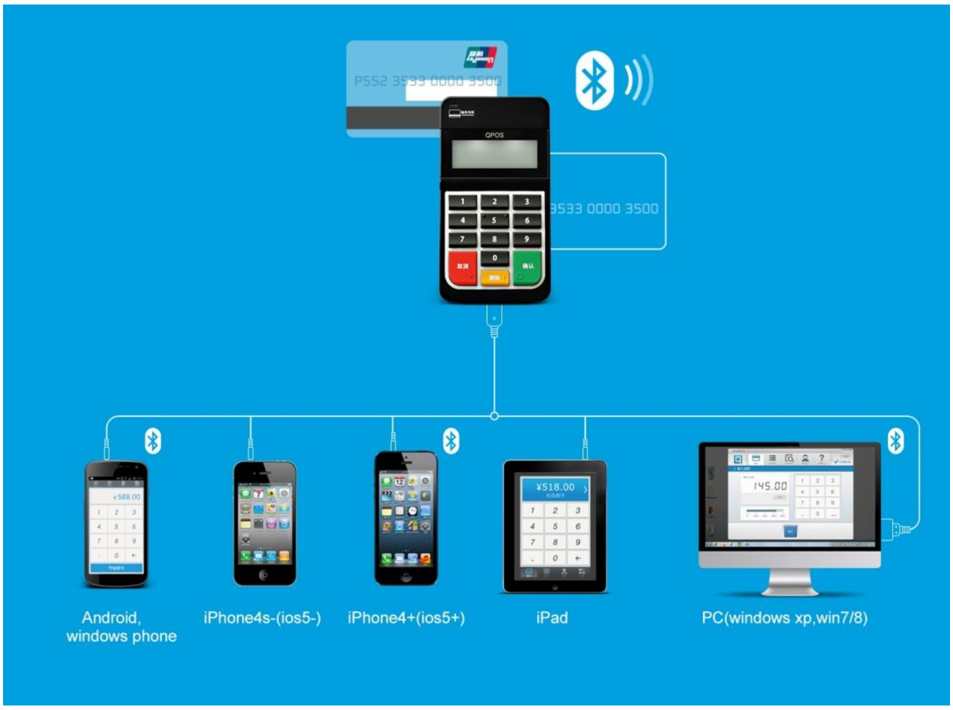
SPOS is a yet another mobile payment card reader device with pinpad and touch screen, SPOS can communicate with the mobile device through many methods, such as: audio jack, Bluetooth and USB cable. The touch screen can be used to capture the signature of consumer in an electronic way.
QPOS share a lot in common, the SDK API is almost same for QPOS. In the following chapters.

Features:

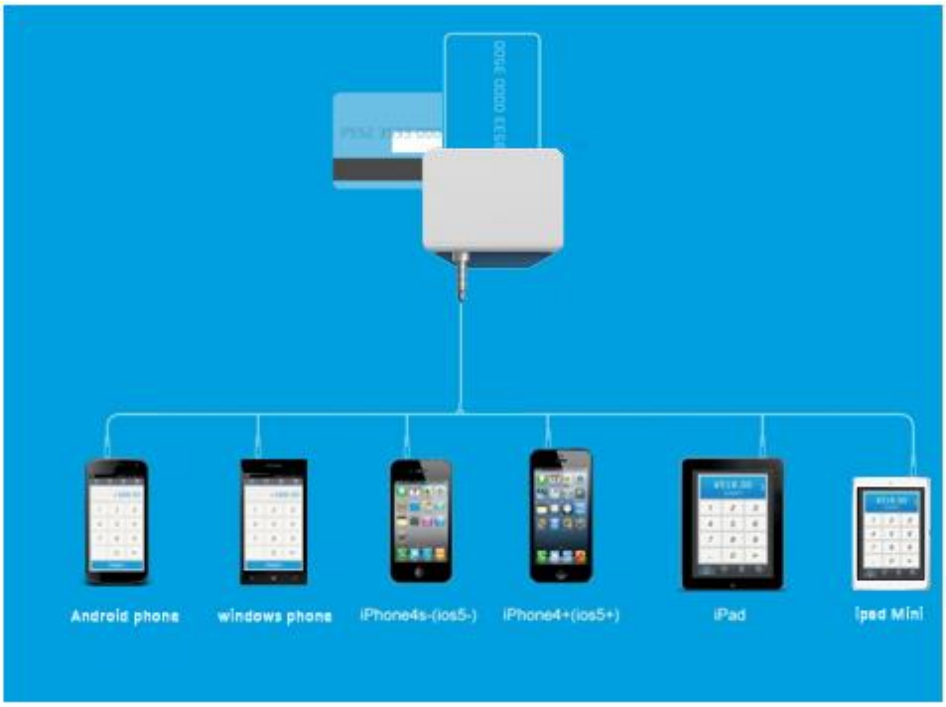
- Ensure secure transactions: integrated keyboard and multiple encryption algorithms to ensure secure transactions.
- Accept all types of bank card: supports magnetic stripe card, contact EMV IC card and contactless EMV IC card.
- Adapts to more smart devices through audio jack: Supports over 2,000 smart devices through audio jack.
- Fulfill global standards: EMV L1&L2, PCI PTS and more.
- Supports all types of mobile systems: Such as iOS, Android, Windows phone and PC OS.

This document is to help readers to integrate QPOS SDK into their mobile payment APPs.

1.2 Connection map For QPOS



1.3 Connection map For QPOS Reader



1.4 Purpose and Scope

This document is to describe the APIs of QPOS iOS SDK . The goal of SDK is to communicate between the smart device and QPOS.
The readers of the document are those who plans to use QPOS SDK in their application.

1.5 Glossary and Definitions

DUKPT	DERIVED UNIQUE KEY PER TRANSACTION.
PK_Q	QPOS Public Key
SK_Q	QPOS Private Key
PK_P	Payment Operator Public Key
SK_P	Payment Operator Private Key
PK_T	Terminal Manufacturer Public Key
SK_T	Terminal Manufacturer Private Key
KSN	Key Serial Number
BDK	Base Derivation Key
IPEK	Initial PIN Encryption Key
DATA-key	The data key to be generated by KSN and IPEK or by KSN and BDK
PIN-key	The PIN key to be generated by KSN and IPEK or by KSN and BDK

1.6 Transaction Key

Unless otherwise specified, Triple DES encryption with EBC and DUKPT key management are assumed. DUKPT is specified in ANSI X9.24 part 1.
Refer to
http://en.wikipedia.org/wiki/DUKPT#Key_Register_.2832_hexadecimal_digits.29

In the Demo, the default transaction keys refer to the following table.

Key Name	Default	Length(Bytes)
KSN	00000332100300e00001	20
BDK	0123456789ABCDEFFEDCBA9876543210	32

1.7 EMVCo Terminal Type Approval

EMVCo has approved QPOS application EMVCo type for Terminal level 2. QPOS application is based on the requirements stated in the EMV 4.3 specification.

1.8 Message Format

Messages within data communication protocols between the mobile payment application and QPOS EMV kernel are encoded as a BER-TLV (Basic Encoding Rules-Tag-Length-Value) which is defined in EMV 4.3 book3 Annex B.

1.9 EMV Standard Tags

EMV Standard Tags are defined in EMV 4.3 book3 Annex A.

1.10 Proprietary Tags Description

Tag	Description	Length(Bytes)	Key	Algorithm
0xC0	KSN of online message	10	No	No
0xC3	KSN of Batch	10	No	No
0xC4	Masked PAN	0-10	No	No
0xC5	Batch message[1]	Var	DATA-key	Triple-Des
0xC2	Online message[3]	Var	DATA-key	Triple-Des
0x70	Online EMV data message[2]	Var	No	No

Note:

1. The **Batch message** is the Triple-Des encrypted result with Data-key. For using, first Triple-Des decrypted the **batch message** with **DATA-key**, the decrypted result is encoded as a

BER-TLV which is defined in EMV 4.3 book3 Annex B.

2. The **Online EMV data message** is encoded as a BER-TLV which is defined in EMV 4.3 book3 Annex B.

3. The **Online message** is the Triple-Des encrypted result with Data-key. For using, first Triple-Des decrypted the **Online message** with **DATA-key,** the decrypted result is encoded as a BER-TLV which is defined in EMV 4.3 book3 Annex B and Proprietary Tags.

1.11 Bluetooth Mode

1.11.1 How to get QPOS/SPOS Bluetooth ID

QPOS Bluetooth ID is combined by ‘QPOS’ strings and the last-10 numbers of the label.

For example:

QPOS_ID := 12070002000200100151

QPOS_BT_ID := QPOS0200100151

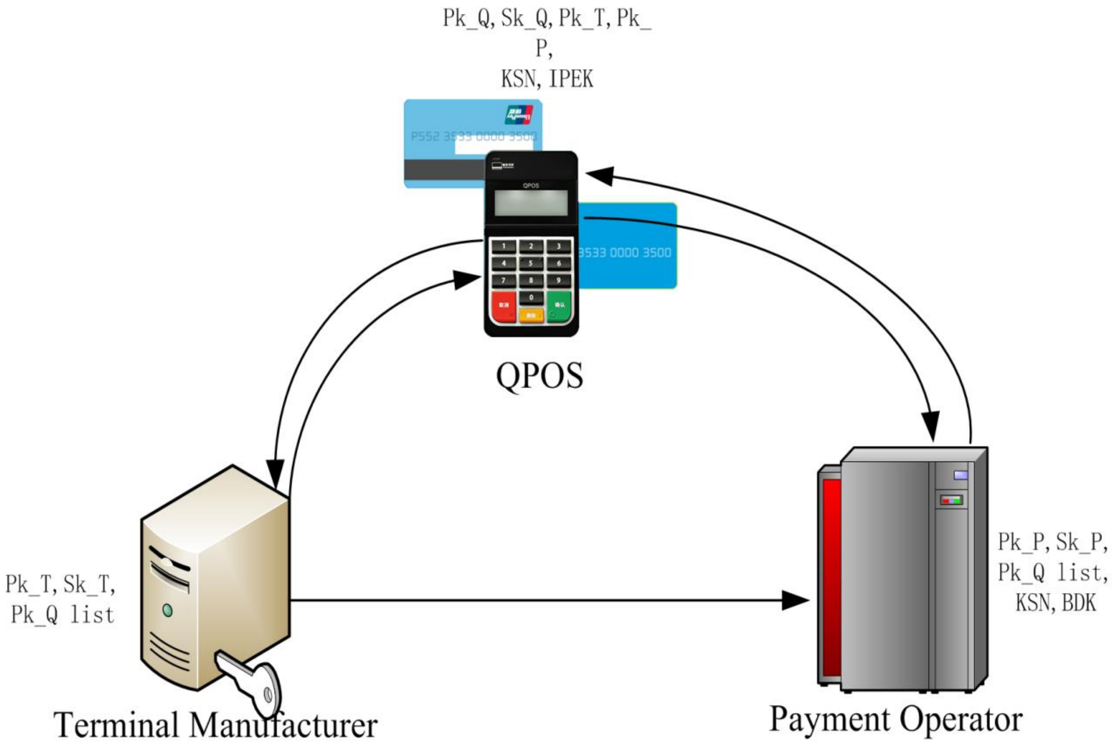
(The label is sticked on QPOS back cover)

SPOS Bluetooth ID is similar to QPOS' but starting with SPOS.

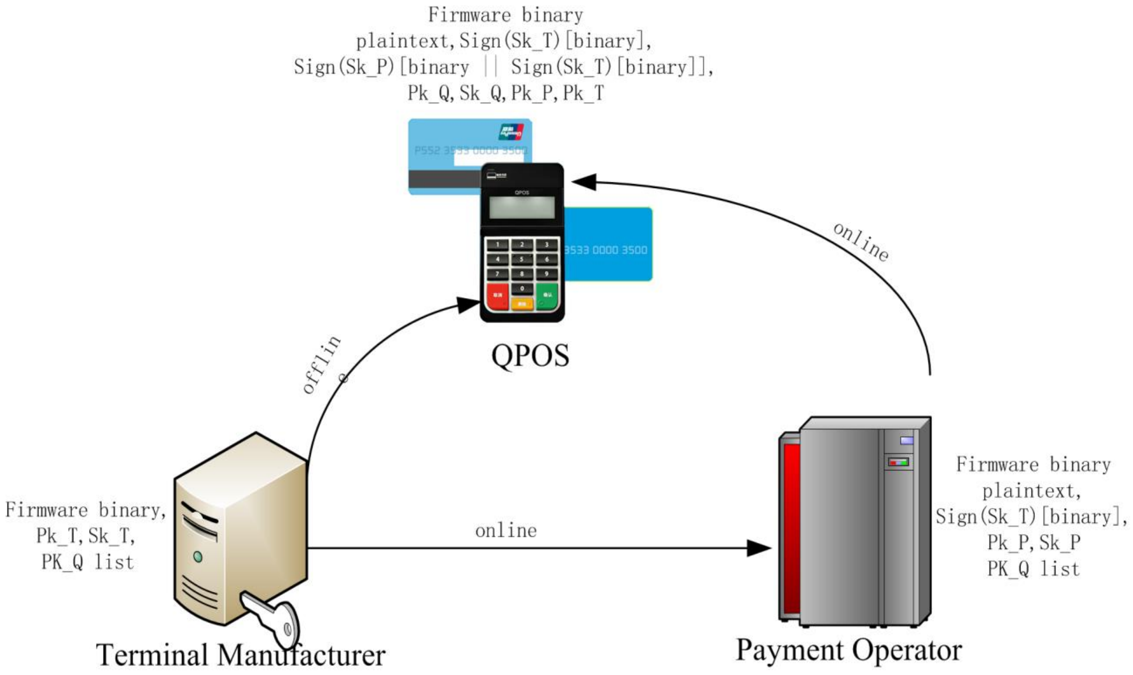
1.11.2 How to connect QPOS/SPOS through Bluetooth

CONNECT METHOD	Automatic
PASSWORD	NA

1.12 Management Keys For QPOS



1.13 Firmware Upgrade For QPOS



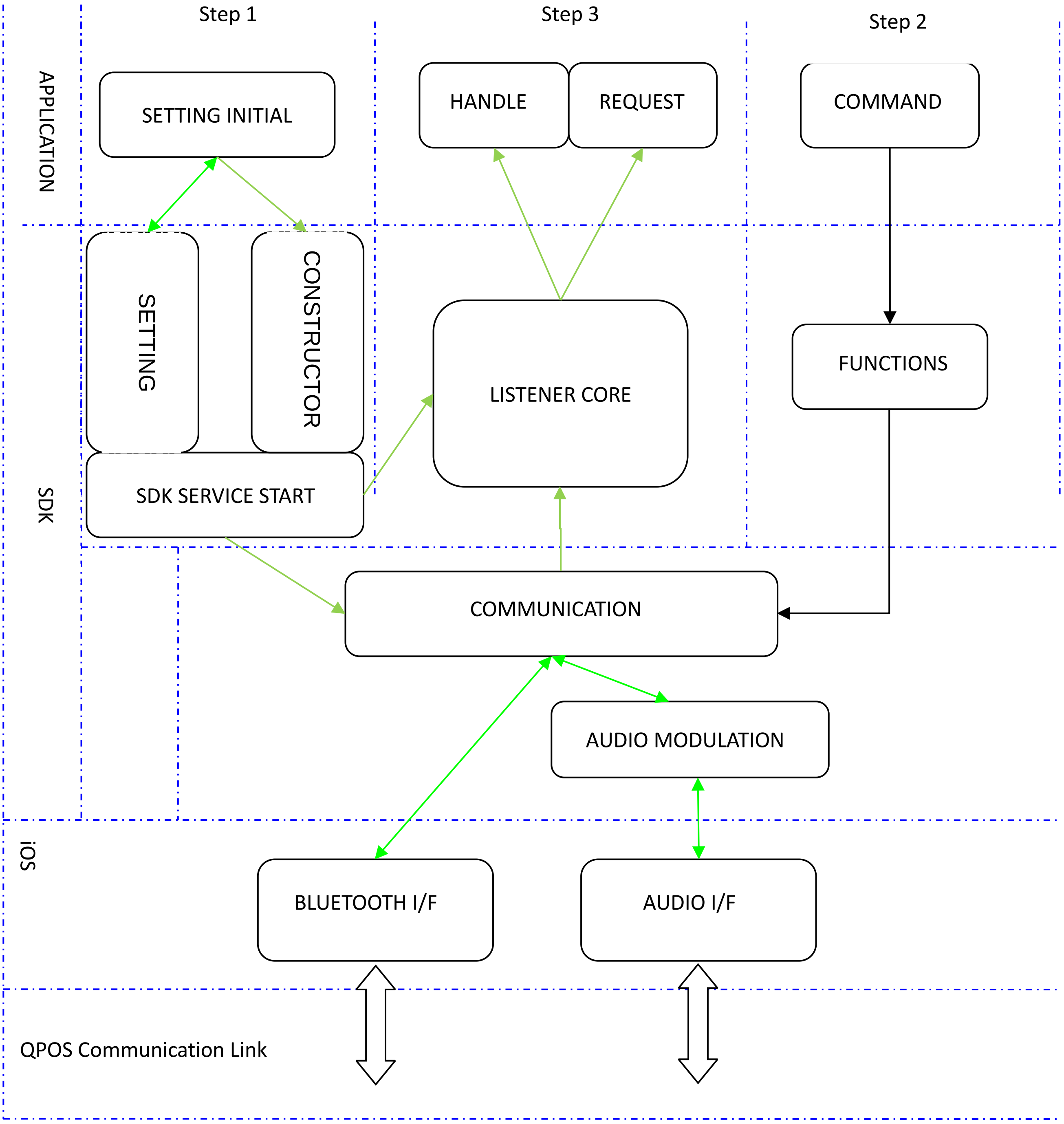
2 iOS SDK API

2.1 System Requirement

Target System (Audiojack):
iOS 3.2 or above (iOS 4 or above is recommended)
iPhone 3GS or above (iPhone 4 or above is recommended)
iPod Touch 3
Generation or above
iPad 1 or above

Target System (Bluetooth):
iPhone 4S / new Pad or above

2.2 SDK Framework



2.3 XCode Project Settings

The library needs several frameworks to be included into the XCode project for a successful compilation:

AudioToolbox.framework
CoreBluetooth.framework
AVFoundation.framework

2.4 SDK Methods

2.4.1 Common methods

With iOS SDK, developer has two ways to communicate with QPOS, audiojack or Bluetooth.

Initiation with audio mode:

```
QPOSService *pos = [QPOSService sharedInstance];
[pos setDelegate:self];
[pos setPosType:PosType_AUDIO];
[pos setQueue:nil];
```

Initiation with bluetooth Mode:

```
QPOSService *pos = [QPOSService sharedInstance];
[pos setDelegate:self];
[pos setPosType: PosType_BLUETOOTH];
[pos setQueue:nil];
```

Method Name	Description
sharedInstance	Get an QPOSService instance
getSdkVersion	Get this SDK version
resetQPOS	Reset and bring the QPOS back to a known initial state.
resetPosStatus	Reset and bring the QPOS back to a known initial state. Synchronized methods
startAudio	Start the audio instance for playing, recording and modulating
stopAudio	Stop the audio instance for playing, recording and

	modulating
connectBT	Connect to QPOS by Bluetooth, using for exchanging data between APP and QPOS
disconnectBT	Disconnect from QPOS with bluetooth

2.4.2 Transaction related methods

Method Name	Description		Handle Name
getQposId	Get the serial number about Qpos	→	onQposIdResult
getQposInfo	Get the config information from Qpos	→	onQposInfoResult
setAmount	Set the amount and transaction type required for EMV transaction.	←	onRequestSetAmount
cancelSetAmount	Cancel the process about setting amount	←	onRequestSetAmount
doTrade	Send the command as swiping/inserting card to QPOS.	→	onDoTradeResult
doEmvApp	Send the command as executing the EMV transaction flow to QPOS		no
selectEmvApp	Select one application of the application list returned from EMV kernel, then set the application ID to EMV kernel	←	onRequestSelectEmvApp

cancelSelectEmvApp	Cancel the process about setting one application	←	onRequestSelectEmvApp
finalConfirm	Send transaction confirmation command to Qpos	←	onRequestFinalConfirm
sendOnlineProcessResult	Send the connectivity status about network to Qpos		
isServerConnected	Send the connectivity status about network to Qpos		
sendTime	Set the date and time formatted as ‘YYMMDDHHMMSS’ to Qpos,based the smart terminal date and time	←	onRequestTime
sendPin	Set the pin to QPOS Reader	←	onRequestSetPin
emptyPin	Set the empty pin to QPOS Reader	←	onRequestSetPin
cancelPin	Set cancel to QPOS Reader	←	onRequestSetPin
powerOnIcc	Turn on the EMV card.	←	onReturnPowerOnIccResult
powerOffIcc	Turn off the EMV card.	←	onReturnPowerOffIccResult
sendApdu	Send data to EMV card in raw APDU formats. This is the EMV Level 1 protocol and developers can develop their only EMV Level 2 application	←	onReturnApduResult
setPosSleepTime	Set the pos sleep time	←	onReturnSetSleepTimeResult
udpateWorkKey	update the pos work key	←	onRequestUpdateWorkKeyResult

2.4.3 Delegate Methods

The controller has a public member delegate that must be set with an object that Implements the protocol **QPOSServiceListener**. This delegate handles different asynchronous events that occur during the operation of the QPOS.

The developer must design a class that implements the interface and assign it to the **delegate** member.

QPOSServiceListener Protocol Methods

Handle Name	Description			Method Name
onRequestWaitingUser	Qpos is ready and waiting for swiping or inserting a EMV card	o		no
onQposIdResult	Return the serial number about Qpos	o	←	getQposId
onQposInfoResult	Return the config information about Qpos	o	←	getQposInfo
onDoTradeResult	Return the action about swiping, inserting the ICC, canceling etc.	m	←	doTrade
onRequestSetAmount	Prompt to inputting the transaction amount to the application	m	→	setAmount
onRequestSelectEmvApp	Supply application list supported by EMV ICC to the application for selecting.	m	→	selectEmvApp

onRequestIsServerConnected		m	→	isServerConnected
onRequestFinalConfirm	Finally confirm before generating the AC by the ICC COS	m	→	finalConfirm
onRequestOnlineProcess	EMV kernel request the online handler	m	→	sendOnlineProcessResult
onRequestTime	Request setting the date and time from Qpos	o	→	sendTime
onRequestTransactionResult	After finishing this transaction, EMV kernel report this transaction result to the application	m		
onRequestTransactionLog	After finishing this transaction, EMV kernel report this transaction log to the application	o		
onRequestBatchData	EMV kernel send the batch data to the application	m		
onRequestPosConnected	SDK report to the application about the connected event between smart terminal and Qpos	m		
onRequestPosDisconnected	SDK report to the application about the disconnected event between smart terminal and Qpos	m		
onRequestNoPosDetected	SDK report to the application about the no connected event between	m		

	smart terminal and Qpos			
onError	SDK report the error ID to the application during transaction	m		
onRequestDisplay	SDK request display to the application	o		
onRequestSetPin	SDK request the application to set PIN for the EMV card. Note, this is only available for QPOS Reader since QPOS Reader doesn’t has PINPAD.	o	→	sendPin emptyPin cancelPin
onReturnPowerOnIccResult	Turn on the EMV card result	o	→	powerOnIcc
onReturnPowerOffIccResult	Turn off the EMV card result	o	→	powerOffIcc
onReturnApduResult		o	→	sendApdu
onReturnSetSleepTimeResult	Set the pos sleep time	o	→	setPosSleepTime
onRequestUpdateWorkKeyResult	update the pos work key	o	→	udpateWorkKey

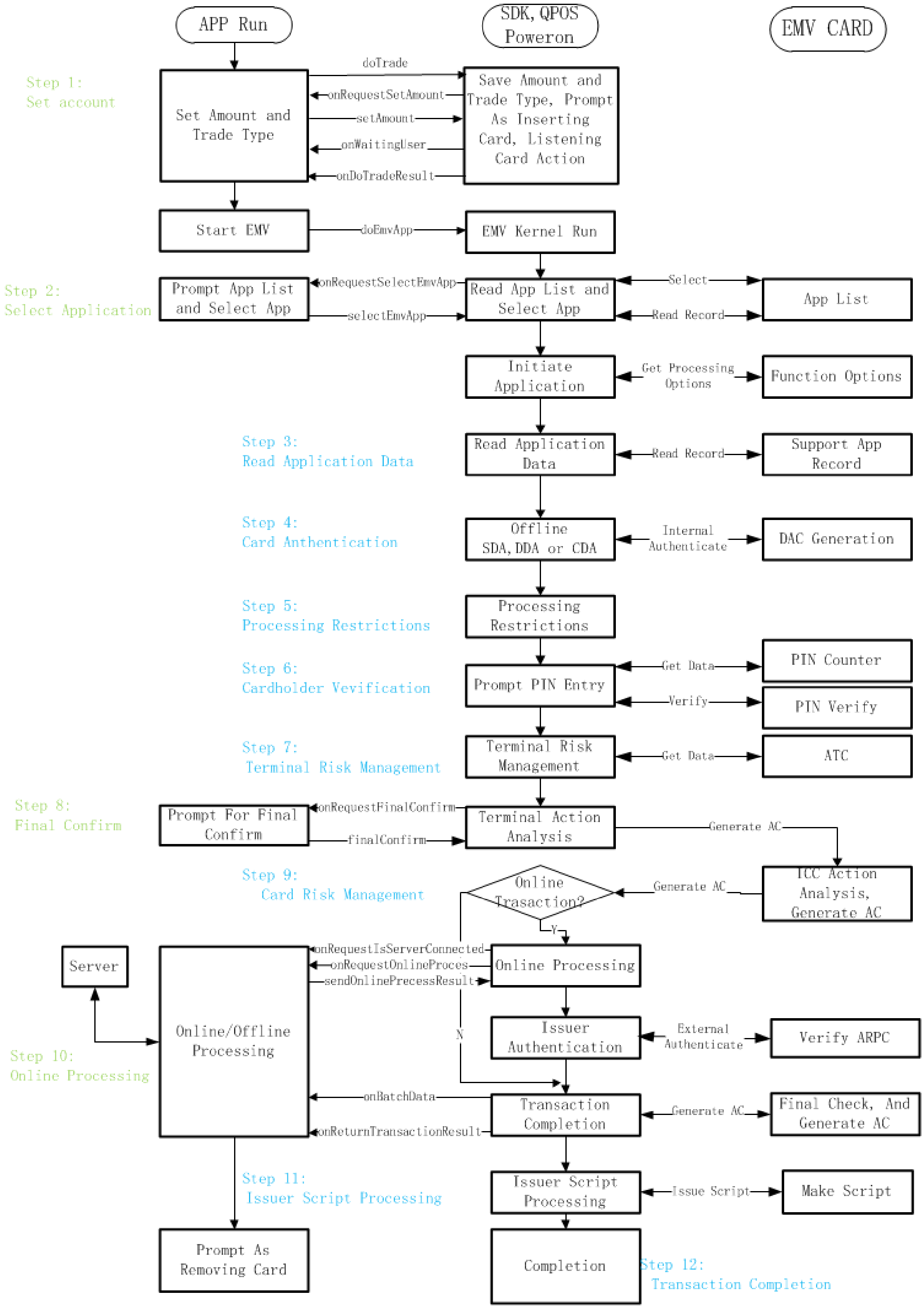
2.5 Transaction Flow

From the return of ‘**onDoTradeResult’**, the APP can find out whether consumer use an EMV ICC card or a magnetic card.

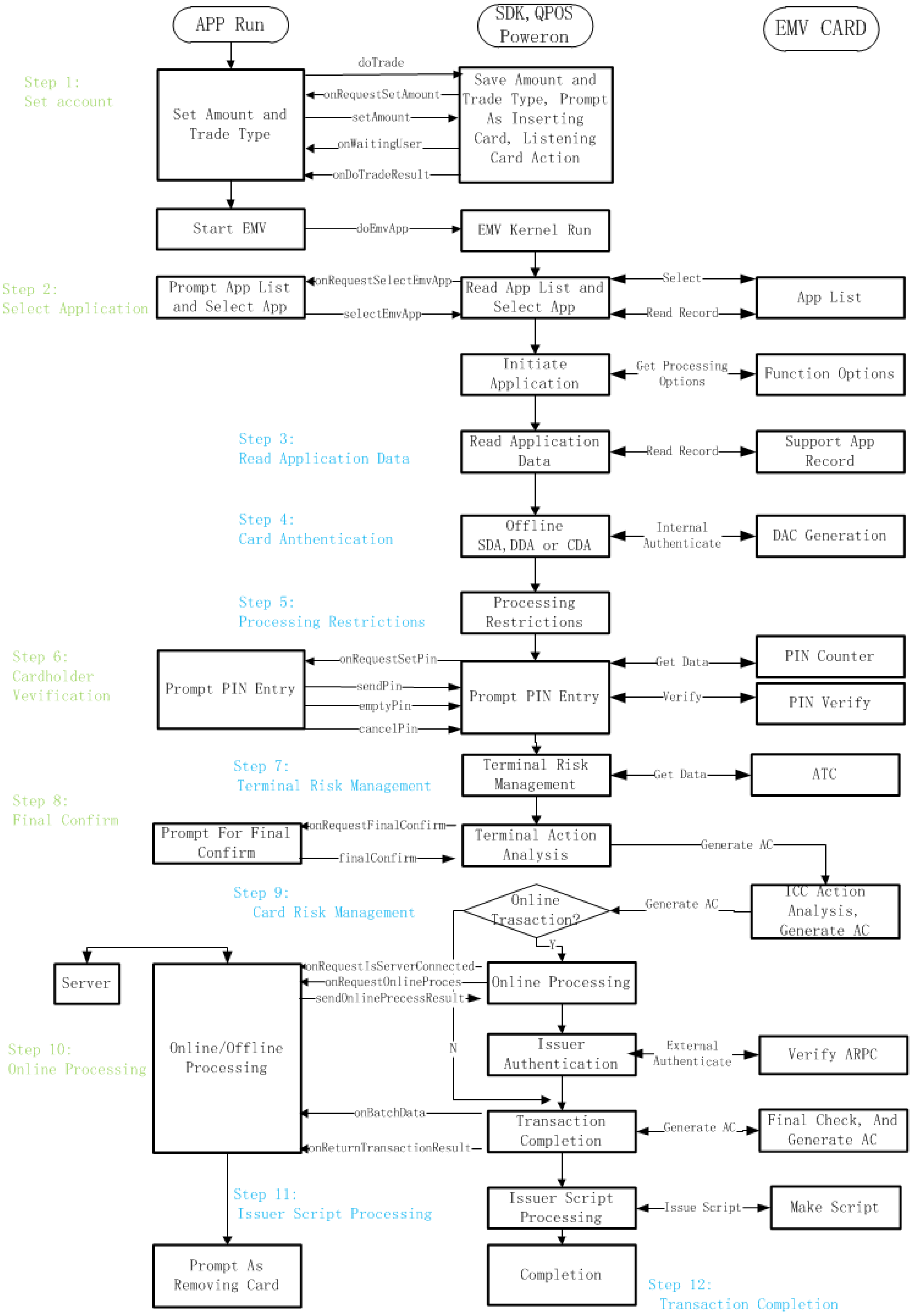
```
-(void) onDoTradeResult: (DoTradeResult)result DecodeData:(NSDictionary*)decodeData
typedef NS_ENUM(NSInteger, DoTradeResult)
{
        DoTradeResult_NONE,
        DoTradeResult_MCR,
        DoTradeResult_ICC,
        DoTradeResult_BAD_SWIPE,
        DoTradeResult_NO_RESPONSE,
        DoTradeResult_NOT_ICC
};
```

When the delegate method “**onDoTradeResult”** is triggered to return the “Enum DoTradeResult” result, if this result is **DoTradeResult_ICC**, the APP will trigger the EMV ICC transaction flow, or this result is **DoTradeResult_MCR**, the APP will trigger the magnetic transaction flow.

2.5.1 EMV ICC Transaction Flow For QPOS



2.5.2 EMV ICC Transaction Flow for QPOS Reader



2.5.3 EMV ICC Transaction Description

Step 1: A transaction amount is needed for a payment transaction and it to be entered by the operator (or calculated by the inventory system) regardless of whether magnetic stripe card or EMV card. The **onRequestSetAmount** delegate method is triggered.

Public void **onRequestSetAmount** ();

The APP should prompt the operator to enter the amount and then call the **setAmount** to send the data back to EMV kernel.

Public void **setAmount**(String amount,String cashbackAmount,String currency,TransactionType transactionType);//2 decimal places.e.g.234.87
The **transactionType** can be any of the following:

GOODS,
SERVICES,
CASHBACK,
INQUIRY,
TRANSFER,
PAYMENT

The amount has an upper limit of 1000000000.00.
The user can also select to abort the transaction.

Public void **cancelSetAmount**();

Step 2: Select Application
An EMV card may support multiple payment applications. The EMV kernel reads the list of applications supported by the EMV card and asks the customer/operator to select the desired application.
The delegate method **onRequestSelectApplication** is triggered to return an array of application Ids.

Public void **onRequestSelectEmvApp** (ArrayList<String>applist);

The APP should prompt the user to select one application and then call the **selectEmvApp** method.

Public void **selectEmvApp** (int index)

The user can also select to abort the transaction.

Public void **cancelSelectEmvApp** ()

In most cases, there is only one default application and this step is skipped.

Step 3: Read Application Data

In this step, EMV kernel reads the necessary data from the EMV card. The EMV kernel asks for the terminal time through the **onRequestTime** method.
This step is only done between the EMV kernel and EMV card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops.

Public void **onRequestTime** ();

The terminal time in YYMMDDHHmmss formats should be sent in response:

Public void **sendTime**(String terminalTime);

Step 4: Card Authentication

This step is only done between the EMV kernel and EMV ICC card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops.

Step 5: Processing Restrictions

This step is only done between the EMV kernel and EMV ICC card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops.

Step 6: Cardholder Verification

There are the different cardholder verification methods (CVMs) supported in an EMV transaction and some require the customer to enter a PIN (personal identifier number). If the EMV transaction requires PIN verification, the customer must enter his PIN by the keypad of QPOS. Where PIN is 4-12 digits. The PIN can be input via PINPAD for QPOS and SPOS, or can be input via mobile application for QPOS Reader.

Some applications (e.g. for small amount payment) does not require CVM (Cardholder Verification Method) and this step is skipped. But it is also possible that the EMV card decline a transaction without PIN. The customer/operator can press the **cancel** key on the keypad of QPOS to cancel the PIN entry and abort.

Step 7: Terminal Risk Management

This step is only done between the EMV kernel and EMV ICC card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops.

Step 8: Terminal Action Analysis

This step is only done between the EMV kernel and EMV ICC card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops. At the end of this step, a final confirmation will be needed to proceed via **onRequestFinalConfirm**.

Public void **onRequestFinalConfirm**();

The APP should prompt the user for a confirmation to proceed. This gives the user a chance to review the amount, the payment method, etc.
A final confirmation is sent to EMV kernel by calling this :

public void **finalConfirm**(boolean isConfirmed);

Step 9: Card Risk Management
This step is only done between the EMV kernel and EMV ICC card. If this step fails, **onRequestTransactionResult** will be returned and the EMV process stops.

Step 10: Online Processing
An EMV transaction can either be online or offline. If online processing is required, then the **onRequestOnlineProcess** delegate method is triggered.

public void **onRequestOnlineProcess**(string tlv);

The parameter TLV contains the tag-length-value data structure returned by the EMV kernel. After that, the client APP should send the data to the payment operator. When the processing results are returned from the payment operator, it should send the results back to EMV kernel by **sendOnlineProcessResult.**

public void **sendOnlineProcessResult**(String tlv);

The data elements that are required are payment operator and issuer dependent. See **chapter 1.9** and the **EVM Book 3 Annex A** for the full list of tags and the TLV structure.

The following tags are usually required but are ICC dependent:

Tags	Parameter
0089	Authorisation Code
008A	Authorisation Response Code
0091	Issuer Authentication Data
0071	Issuer Script Template 1 (needed for Step 11)
0072	Issuer Script Template 2 (needed for Step 11)

This step is skipped in offline processing.

Step 11: Issuer Scripts Processing
This step is handled transparently between the EMV kernel and EMV ICC card if issue scripts are present in the online processing results or skipped otherwise. This step is skipped in offline processing.

Step 12: Completion
In this step, EMV kernel sends back the final transaction result from the EMV card by the

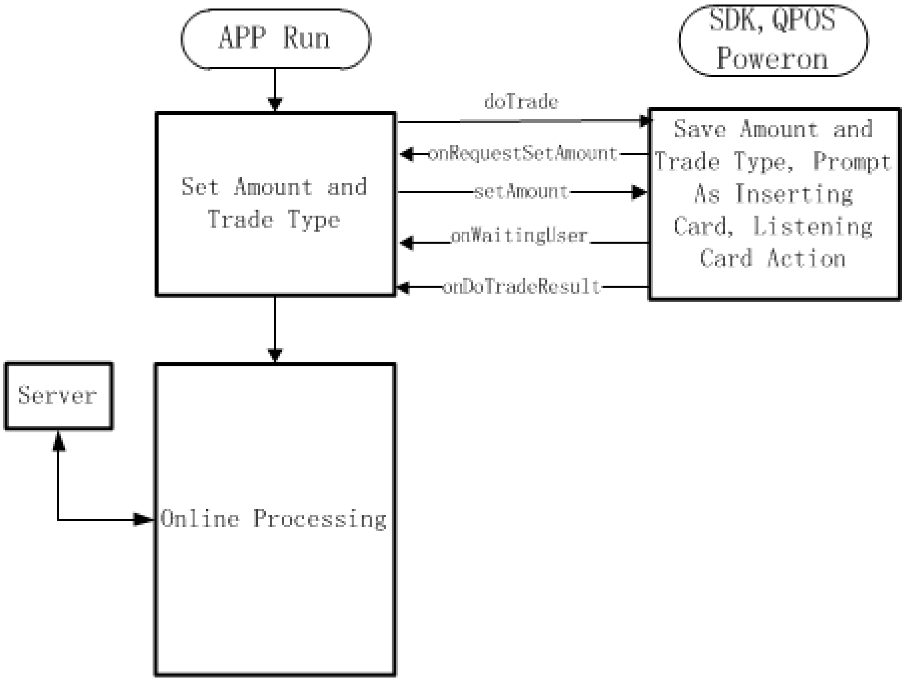
onRequestBatchData method.

public void **onRequestBatchData** (String tlv);

The data elements that are required are payment operator and issuer dependent. See the **EVM Book 3 Annex A** for the full list of tags and the TLV structure.

In this step, the client APP should store the results, display the results, print receipts, and prompt the user to remove the card from the QPOS device. Later, the batch data should be updated to the server for settlement.

2.5.4 Magnetic Transaction Flow



2.5.5 Magnetic Transaction Description

If a magnetic stripe card has been swiped, the encrypted PINBLOCK and the encrypted track data will be returned along with the **trackksn** and the **pinKsn** in the **decodeData Hashtable**.

Public void **onDoTradeResult**(DoTradeResult result, Hashtable<String, String> decodeData);

The **Hashtable** contain keys for the values:

Key	Description
maskedPAN	Masked card number showing at most the first 6 and last 4 digits with in-between digits masked by “X”

expiryDate	4-digit in the form of YYMM in the track data
cardHolderName	The cardholder name as seen on the card. This can be up to 26 characters.
serviceCode	3-digit service code in the track data
track1Length	Length of Track 1 data
track2Length	Length of Track 2 data
track3Length	Length of Track 3 data
encTracks	Reserved
encTrack1	Encrypted track 1 data with T-Des encryption key derived from **DATA-key** to be generated with **trackksn** and **IPEK**
encTrack2	Encrypted track 2 data with T-Des encryption key derived from **DATA-key** to be generated with **trackksn** and **IPEK**
encTrack3	Encrypted track 3 data with T-Des encryption key derived from **DATA-key** to be generated with **trackksn** and **IPEK**
partialTrack	Reserved
pinKsn	KSN of the Pin-block
trackksn	KSN of the track data
pinBlock[1]	Encrypted PIN data with T-Des encryption key derived from **PIN-key** to be generated with **pinKsn** and **IPEK**

Note:

1. The **PIN** format is defined as **ANSI X9.8**.

   pinBlock := (**PIN^PAN**);

2.6 API Methods Reference

2.6.1 getSdkVersion

Signature	-(NSString *) getSdkVersion
Inputs	None
Outputs	SDK version
Description	Return the SDK version
See also	

2.6.2 resetPos

Signature	-(BOOL) resetPos;
Inputs	None
Outputs	BOOL
Description	Reset and bring the QPOS back to a known initial state.
See also	

2.6.3 startAudio

Signature	-(void)startAudio;
Inputs	None
Outputs	None
Description	Start the audio instance for playing, recording and modulating.

See also	

2.6.4 stopAudio

Signature	-(void)stopAudio;
Inputs	None
Outputs	None
Description	Stop the audio instance for playing, recording and modulating.
See also	

2.6.5 connectBT

Signature	-(BOOL) connectBT: (NSString *)bluetoothName;
Inputs	Bluetooth Name.
Outputs	BOOL
Description	Connect to QPOS by Bluetooth name, using for exchanging data between APP and QPOS.
See also	
Example	[pos connectBT:@"MPOS0010900318"];

2.6.6 disconnectBT

Signature	-(void) disconnectBT;
Inputs	None
Outputs	None

Description	Connect to QPOS by Bluetooth, using for exchanging data between APP and QPOS.
See also	

2.6.7 isQposPresent

Signature	-(BOOL)isQposPresent;
Inputs	None.
Outputs	BOOL: Presence of the EmvSwipe.
Description	Check if an QPOS is connected and ready.
See also	

2.6.8 getQposId

Signature	-(void) getQPosId; / -(void) getQPosId:(NSInteger)timeout;
Inputs	timeout;
Outputs	None
Description	Retrieve the POS_ID of the QPOS device. Results are returned by onQposIdResult.
See also	onQposIdResult

2.6.9 getQposInfo

Signature	-(void) getQPosInfo;
Inputs	None
Outputs	None

Description	Retrieve parameters about the QPOS device. Results are returned by onQposInfoResult which includes: firmware version, bootloader version, USB connection and charging status, battery level, and hardware version
See also	onQposInfoResult

2.6.10 setAmount

Signature	-(void) setAmount: (NSString *)aAmount aAmountDescribe:(NSString *)aAmountDescribe currency:(NSString *)currency transactionType:(TransactionType)transactionType;
Inputs	aAmount: the amount for a transaction. aAmountDescribe: cashback amount, this parameter needs to be set if cashback transaction is performed. currency: three digits of the currency code, e.g. “840” for USD transactionType: GOODS, SERVICES, CASHBACK, INQUIRY, TRANSFER, PAYMENT
Outputs	None
Description	Set the amount, currency and type of a transaction. This method can be called before a transaction or in response to an onRequestSetAmount call requested by the EMV engine.
See also	onRequestSetAmount , cancelSetAmount
Example	[pos setAmount:@ "500" aAmountDescribe:@"123" currency: @"0156" transactionType: TransactionType_GOODS];

2.6.11 cancelSetAmount

Signature	-(void) cancelSetAmount;

Inputs	None
Outputs	None
Description	Cancel setting the amount of a transaction. This method can be called to abort a transaction in response to onRequestSetAmount
See also	onRequestSetAmount

2.6.12 doTrade

Signature	-(void) doTrade; / -(void) doTrade:(NSInteger) timeout;
Inputs	Timeout
Outputs	None
Description	Check the status of the Magnetic Card Reader, the EMV Card reader, or NFC transceiver. It checks if a card has been swiped, a NFC card has been tapped or an EMV card is inserted. The result is returned by the onDoTradeResult delegate method.
See also	onDoTradeResult

2.6.13 doCheckCard

Signature	-(void) doCheckCard;/-(void) doCheckCard:(NSInteger) timeout;
Inputs	Timeout
Outputs	None
Description	Start transaction，no pin input
See also	onDoTradeResult

2.6.14 doEmvApp

Signature	-(void) doEmvApp: (EmvOption)aemvOption;
Inputs	EmvOption
Outputs	None
Description	Start Emv app，Emv card
See also	onDoTradeResult

2.6.15 setAmount

Signature	-(void) setAmount: (NSString *)aAmount aAmountDescribe:(NSString *)aAmountDescribe currency:(NSString *)currency transactionType:(TransactionType)transactionType OR -(void) setAmount: (NSString *)aAmount aAmountDescribe:(NSString *)aAmountDescribe currency:(NSString *)currency transactionType:(TransactionType)transactionType posDisplayAmount:(BOOL)flag;
Inputs	Amount： CashbackAmount CurrencyCode TransactionType IsPosDisplayAmount
Outputs	None
Description	Set amount
See also	

2.6.16 selectEmvApp

Signature	-(void) selectEmvApp: (NSInteger)index;
Inputs	Index: select emv app index
Outputs	None
Description	Select Emv App
See also	

2.6.17 cancelSelectEmvApp

Signature	-(void) cancelSelectEmvApp;
Inputs	None
Outputs	None
Description	Cancel Select Emv App
See also	

2.6.18 finalConfirm

Signature	-(void) finalConfirm: (BOOL)isConfirmed;
Inputs	isConfirmed
Outputs	None
Description	Not support
See also	

2.6.19 sendOnlineProcessResult

Signature	-(void) sendOnlineProcessResult: (NSString *)tlv;
Inputs	tlv data consist of 8A+91+71/72

	8A: Authorisation Response Code 91: Issuer Authentication Data 71/72: Issuer Script
Outputs	None
Description	Send the tlv data that returned by the issuer to terminal.
See also	onRequestOnlineProcess
Example	[pos sendOnlineProcessResult:@" 8A023030910A6C741A0100F69FF90012720F860D84240000082129027736EDDA04"];

2.6.20 isServerConnected

Signature	-(void) isServerConnected: (BOOL)isConnected;
Inputs	isConnected
Outputs	None
Description	Send the connectivity status to EMVSwipe
See also	onRequestIsServerConnected

2.6.21 sendTime

Signature	-(void) sendTime: (NSString *)aterminalTime;
Inputs	aterminalTime
Outputs	None
Description	Send the terminal time in yyyyMMddHHmmss format to EMVSwipe
See also	onRequestTime

2.6.22 setAmountIcon

Signature	-(void)setAmountIcon:(NSString *)aAmountIcon; / -(void)setAmountIcon:(AmountType) amtType amtIcon:(NSString *)aAmountIcon;
Inputs	amtType aAmountIcon
Outputs	None
Description	Set Amount symbol. String Data is a string with length below 4 characters, and can only be ASCII string. below are some valid examples : “RUP” “USD” “CNY"
See also	

2.6.23 getPin

Signature	-(void)getPin:(NSString *)aTransactionData;
Inputs	aTransactionData
Outputs	None
Description	Get Pin
See also	onReturnGetPinResult

2.6.24 powerOnIcc

Signature	-(void)powerOnIcc;
Inputs	None
Outputs	None
Description	Provide power to ICC for level 1 APDU exchange
See also	onReturnPowerOnIccResult

2.6.25 powerOffIcc

Signature	-(void)powerOffIcc;
Inputs	None
Outputs	None
Description	Cut off power to ICC after level 1 APDU exchange
See also	onReturnPowerOnIccResult

2.6.26 sendApdu

Signature	-(void)sendApdu:(NSString *)apduStr;
Inputs	apduStr
Outputs	None
Description	Send APDU exchange to ICC. Response data are returned by onReturnApduResult
See also	onReturnApduResult

2.6.27 setPosSleepTime

Signature	-(void)setPosSleepTime:(NSInteger)sleepTime;

Inputs	sleepTime
Outputs	None
Description	Set Sleep Time
See also	onReturnSetSleepTimeResult

2.6.28 updateEmvConfig(Deprecated)

Signature	-(void)updateEmvConfig:(NSString *)emvAppCfg emvCapk:(NSString*)emvCapkCfg;
Inputs	emvAppCfg emvCapkCfg
Outputs	None
Description	update emv config files(CAPK and AID)
See also	onReturnCustomConfigResult

2.6.29 readEmvAppConfig

Signature	-(void)readEmvAppConfig;
Inputs	None
Outputs	None
Description	read emv app config
See also	onReturnCustomConfigResult

2.6.30 readEmvCapkConfig

Signature	-(void)readEmvCapkConfig;
Inputs	None

Outputs	None
Description	read emv capk config
See also	onReturnCustomConfigResult

2.6.31 udpateWorkKey

Signature	-(void)udpateWorkKey:(NSString *)pik pinKeyCheck:(NSString *)pikCheck trackKey:(NSString *)trk trackKeyCheck:(NSString *)trkCheck macKey:(NSString *)mak macKeyCheck:(NSString *)makCheck transKey:(NSString *)tnsk transKeyCheck:(NSString *)tnskCheck keyIndex:(NSInteger) mKeyIndex delay:(NSInteger)timeout;
Inputs	pik pikCheck trk trkCheck mak makCheck tnsk tnskCheck keyIndex timeout
Outputs	None
Description	Update Work Key
See also	onRequestUpdateWorkKeyResult

2.6.32 MacKeyEncrypt

Signature	-(void)MacKeyEncrypt:(NSString *)macStr;
Inputs	macStr
Outputs	None
Description	macKey Encrypt
See also	onRequestCalculateMac

2.6.33 isQposPresent

Signature	-(BOOL)isQposPresent;
Inputs	None
Outputs	BOOL
Description	Get Qpos available status
See also	

2.6.34 setMasterKey

Signature	-(void) setMasterKey:(NSString *)key checkValue:(NSString *)chkValue keyIndex:(NSInteger) mKeyIndex;
Inputs	key ckValue keyIndex timeout
Outputs	None
Description	Set Master Key
See also	onReturnSetMasterKeyResult

2.6.35 calcMacSingle

Signature	-(void)calcMacSingle:(NSString *)macStr;
Inputs	macStr
Outputs	None
Description	Calculate mac (single)
See also	onRequestCalculateMac

2.6.36 calcMacDouble

Signature	-(void)calcMacDouble:(NSString *)macStr;
Inputs	macStr keyIndex timeout
Outputs	None
Description	Calculate mac (double)
See also	onRequestCalculateMac

2.6.37 calcMacSingleNoCheck

Signature	-(void)calcMacSingleNoCheck:(NSString *)macStr delay:(NSInteger)timeout;
Inputs	macStr timeout
Outputs	None
Description	Calculate mac (single)
See also	onRequestCalculateMac

2.6.38 calcMacDoubleNoCheck

Signature	-(void)calcMacDoubleNoCheck:(NSString *)macStr keyIndex:(NSInteger) mKeyIndex delay:(NSInteger)timeout;
Inputs	macStr keyIndex timeout
Outputs	None
Description	Calculate mac (double)
See also	onRequestCalculateMac

2.6.39 downloadRsaPublicKey

Signature	-(void)downloadRsaPublicKey:(NSInteger)useType RID:(NSString*)rid keyIndex:(NSString *)index keyModule:(NSString *)module keyExponent:(NSString *)exponent delay:(NSInteger)timeout;
Inputs	useType rid：public Key RID index：Public Key Index module：Public Key Module exponent：Public Key Exponent timeout
Outputs	None
Description	Acquire server RSA public key Execute process: Step1:call host interface ,get public key from host Step2: call the interface get random key Step 3: upload randomkey to host .host return encrypted terminal master key

	Step 4: call setMasterKey to set master key.
See also	onReturnDownloadRsaPublicKey

2.6.40 updateMasterKeyRandom

Signature	-(void)updateMasterKeyRandom:(NSInteger)step keyIndex:(NSString *)index masterKey:(NSString *)mKey masterKeyCheck:(NSString *)mKeyCheck delay:(NSInteger)timeout;
Inputs	step： step index keyIndex masterKey masterKeyCheck timeout
Outputs	None
Description	Update Master Key in Random method Execute Process: Step 1: call this interface to get random Step 2: use the random number to encrypt terminal master key. Then call the interface to set master key.
See also	onUpdateMasterKeyResult

2.6.41 updateMasterKey

Signature	-(void)updateMasterKey:(NSInteger)step RN1:(NSString *)RN1Str RN2:(NSString *)RN2Str masterKey:(NSString *)mKey masterKeyCheck:(NSString *)mKeyCheck delay:(NSInteger)timeout;
Inputs	step RN1 RN2 masterKey

	masterKeyCheck timeout
Outputs	None
Description	
See also	onUpdateMasterKeyResult

2.6.42 pinKey_TDES

Signature	-(void)pinKey_TDES:(NSInteger) keyIndex pin:(NSString *)inStr delay:(NSInteger)timeout;
Inputs	keyIndex pin timeout
Outputs	None
Description	Use pinKey to encrypt data
See also	onPinKey_TDES_Result

2.6.43 pinKey_TDESNoCheck

Signature	-(void)pinKey_TDESNoCheck:(NSInteger) keyIndex pin:(NSString *)inStr delay:(NSInteger)timeout;
Inputs	keyIndex pin timeout
Outputs	None
Description	Use pinKey to encrypt data (NoCheck)
See also	onPinKey_TDES_Result

2.6.44 setSystemDateTime

Signature	- (void)setSystemDateTime:(NSString *)dateTimeStr delay:(NSInteger)timeout block:(void (^)(BOOL isSuccess, NSDictionary *resultDic))dateTimeBlock;
Inputs	dataTime timeout
Outputs	None
Description	Set SystemDate Time
See also	

2.6.45 setMerchantID

Signature	- (void)setMerchantID:(NSString *)merchantID delay:(NSInteger)timeout block:(void (^)(BOOL isSuccess, NSDictionary *resultDic))merchantIDBlock;
Inputs	merchantID timeout
Outputs	None
Description	Set Merchant ID
See also	

2.6.46 setTerminalID

Signature	- (void)setTerminalID:(NSString *)TerminalID delay:(NSInteger)timeout block:(void (^)(BOOL isSuccess, NSDictionary *resultDic))terminalIDBlock;
Inputs	terminalID timeout

Outputs	None
Description	Set Terminal ID
See also	

2.6.47 getMagneticTrackPlaintext

Signature	- (void)getMagneticTrackPlaintext:(NSInteger)timeout;
Inputs	timeout
Outputs	None
Description	Get Magnetic Track Plaintext
See also	onDoTradeResult

2.6.48 getCardNo

Signature	-(void) getCardNo;
Inputs	None
Outputs	None
Description	Acquire card number（Magnetic stripe card）
See also	onGetCardNoResult

2.6.49 getIccCardNo

Signature	-(void) getIccCardNo: (NSString *)aterminalTime;
Inputs	aterminalTime
Outputs	None
Description	Acquire card number（IC、Magnetic stripe card）
See also	onGetCardNoResult

2.6.50 powerOffNFC

Signature	- (void)powerOffNFC:(NSInteger) timeout withBlock:(void (^)(BOOL isSuccess))onPowerOffNFCResultBlock;
Inputs	timeout
Outputs	None
Description	Turn off the NFC transceiver
See also	onReturnPowerOffNFCResult

2.6.51 sendApduByNFC

Signature	- (void)sendApduByNFC:(NSString *)apduString delay:(NSInteger)timeout withBlock:(void (^)(BOOL isSuccess, NSString *apdu, NSInteger apduLen))onNFCApduResultBlock;
Inputs	apduString timeout
Outputs	None
Description	Send data to EMV card in raw APDU formats by nfc
See also	onReturnNFCApduResult

2.6.52 powerOnNFC

Signature	- (void)powerOnNFC:(NSInteger) isEncrypt delay:(NSInteger) timeout withBlock:(void (^)(BOOL isSuccess, NSString *ksn, NSString *atr, NSInteger atrLen))onPowerOnNFCResultBlock;
Inputs	isEncrypt timeout
Outputs	None

Description	Turn on the NFC transceiver
See also	onReturnPowerOnNFCResult

2.6.53 cbc_mac

Signature	- (void)cbc_mac:(NSInteger)keyLen atype:(NSInteger)algorithmType otype:(NSInteger)operatorType data:(NSString *)dataStr delay:(NSInteger)timeout withResultBlock:(void (^)(NSString *))cbcmacBlock;
Inputs	keyLen algorithmType operatorType data timeout
Outputs	None
Description	With 3DES CBC mode calculate Mac
See also	onCbcMacResult

2.6.54 cbc_macNoCheck

Signature	- (void)cbc_macNoCheck:(NSInteger)keyLen atype:(NSInteger)algorithmType otype:(NSInteger)operatorType data:(NSString *)dataStr delay:(NSInteger)timeout withResultBlock:(void (^)(NSString *))cbcmacBlock;
Inputs	keyLen algorithmType operatorType data

	timeout
Outputs	None
Description	Calculate Mac (No Check)with 3DES CBC mode
See also	onCbcMacResult

2.6.55 inquireECQAmount

Signature	-(void) inquireECQAmount: (NSString *)aterminalTime;
Inputs	transactionTime
Outputs	None
Description	Inquire IC card Electronic pocket balance
See also	onRequestBatchData

2.6.56 isIdle

Signature	-(BOOL)isIdle;
Inputs	transactionTime
Outputs	None
Description	Flag indicate pos is working
See also	

2.6.57 anlysEmvIccData

Signature	-(NSDictionary *)anlysEmvIccData:(NSString *)tlv;
Inputs	tlv
Outputs	None
Description	Parsing ICC data

See also	

2.6.58 VIPOSBatchSendAPDU

Signature	-(void)VIPOSBatchSendAPDU:(NSDictionary *) batchAPDU;
Inputs	batchAPDU
Outputs	None
Description	send Batch APDU command VIPOS protocol
See also	onReturnBatchSendAPDUResult

2.6.59 synVIPOSBatchSendAPDU

Signature	-(NSDictionary *)synVIPOSBatchSendAPDU:(NSDictionary *) batchAPDU;
Inputs	isOpen batchAPDU
Outputs	None
Description	send Batch APDU command VIPOS protocal （synchronize mode）
See also	onReturnBatchSendAPDUResult

2.6.60 anlysEmvIccData_qf

Signature	-(NSDictionary *)anlysEmvIccData_qf:(NSString *)tlv;
Inputs	tlv
Outputs	None
Description	Parsing ICC data qf

See also	

2.6.61 iccCashBack

Signature	-(void)iccCashBack:(NSString *)transactionTime random:(NSString *)aRandom;
Inputs	transactionTime random
Outputs	None
Description	Specific customer IC card cach back trade
See also	onReturniccCashBack

2.6.62 setPosPresent

Signature	-(void) setPosPresent:(BOOL) flag;
Inputs	flag
Outputs	None
Description	Flag for POS existence
See also	

2.6.63 setCardTradeMode

Signature	-(void)setCardTradeMode:(CardTradeMode) aCardTMode;
Inputs	cardTradeMode： enum{ONLY_INSERT_CARD//only IC ONLY_SWIPE_CARD//Only Magnetic stripe card SWIPE_INSERT_CARD//Magnetic stripe card and IC UNALLOWED_LOW_TRADE//the devices will remind to insert the card if the card are both chip and track }

Outputs	None
Description	Set Card Trade Mode, （Only Magnetic stripe card，only IC，Magnetic stripe card and IC ）
See also	

2.6.64 setPinPadFlag

Signature	-(void)setPinPadFlag:(BOOL)flag;
Inputs	flag
Outputs	None
Description	Device keypad flag
See also	

2.6.65 qposStatus

Signature	-(BOOL)qposStatus;
Inputs	None
Outputs	None
Description	Check APP connect status
See also	

2.6.66 readBusinessCard

Signature	- (void)readBusinessCard:(NSString *)cardType businessID:(NSInteger)businessID pin:(NSString *)pinStr address:(NSString *)addr readLen:(NSInteger)len delay:(NSInteger)timeout withResultBlock:(void (^)(BOOL isSuccess, NSString * result))readBusinessCardResultBlock;
Inputs	cardType

	address readLen cardPin vender_id timeout
Outputs	None
Description	Read Business Card
See also	onReadBusinessCardResult

2.6.67 writeBusinessCard

Signature	- (void)writeBusinessCard:(NSString *)cardType businessID:(NSInteger)businessID address:(NSString *)addr writeData:(NSString *)data cardPin:(NSString *)pin isUpdatePin:(BOOL)updateFlag delay:(NSInteger)timeout withResultBlock:(void (^)(BOOL isSuccess, NSString * result))writeBusinessCardResultBlock;
Inputs	cardType address data cardPin isUpdatePinFlag vender_id timeout
Outputs	None
Description	write Business Card
See also	onWriteBusinessCardResult

2.6.68 syncReadBusinessCard

Signature	- (NSData *)syncReadBusinessCard:(NSString *)cardType businessID:(NSInteger)businessID pin:(NSString *)pinStr address:(NSString *)addr readLen:(NSInteger)len delay:(NSInteger)timeout;
Inputs	cardType address readLen cardPin vender_id timeout
Outputs	None
Description	Read Business Card（synchronize mode）
See also	

2.6.69 syncWriteBusinessCard

Signature	- (NSInteger)syncWriteBusinessCard:(NSString *)cardType businessID:(NSInteger)businessID address:(NSString *)addr writeData:(NSString *)data cardPin:(NSString *)pin isUpdatePin:(BOOL)updateFlag delay:(NSInteger)timeout;
Inputs	cardType address data cardPin isUpdatePinFlag vender_id timeout

Outputs	None
Description	Write Business Card（synchronize mode）
See also	

2.6.70 confirmAmount

Signature	- (void)confirmAmount:(NSString *)wKey delay:(NSInteger)timeout withResultBlock:(void (^)(BOOL isSuccess))confirmAmountBlock;
Inputs	amount timeout
Outputs	None
Description	Confirm Amount
See also	onConfirmAmountResult

2.6.71 setAmount

Signature	-(void) setAmount: (NSString *)aAmount aAmountDescribe:(NSString *)aAmountDescribe currency:(NSString *)currency transactionType:(TransactionType)transactionType posDisplayAmount:(BOOL)flag;
Inputs	amount：transaction amount cashbackAmount：enchashment amount currencyCode：currency code transactionType：transaction type isPosDisplayAmount：whether display on POS device or not
Outputs	None
Description	Set Amount

See also	

2.6.72 getPin

Signature	- (void)getPin:(NSInteger)encryptType keyIndex:(NSInteger)keyIndex maxLen:(NSInteger)maxLen typeFace:(NSString *)typeFace cardNo:(NSString *)cardNo data:(NSString *)data delay:(NSInteger)timeout withResultBlock:(void (^)(BOOL isSuccess, NSDictionary * result))getPinBlock;
Inputs	encryptType：default:0 keyIndex：default:0 maxLen：max length of pin typeface：display the font cardNo data：attached data timeout
Outputs	None
Description	Get Pin
See also	onReturnGetPinResult
Example	[pos getPin:1 keyIndex:0 maxLen:6 typeFace:@"Pls Input Pin" cardNo:@"622262XXXXXXXXX4406" data:@"" delay:30 timeout:20];

2.6.73 doUpdateIPEKOperation

Signature	-(void)doUpdateIPEKOperation:(NSString *)groupKey tracksn:(NSString *)trackksn trackipek:(NSString *)trackipek trackipekCheckValue:(NSString *)trackipekCheckValue emvksn:(NSString *)emvksn emvipek:(NSString *)emvipek

	emvipekcheckvalue:(NSString *)emvipekcheckvalue pinksn:(NSString *)pinksn pinipek:(NSString *)pinipek pinipekcheckValue:(NSString *)pinipekcheckValue block:(void(^)(BOOL isSuccess,NSString *stateStr))EMVBlock;
Inputs	groupKey、tracksn、trackipek、trackipekCheckValue、emvksn、emvipek、emvipekcheckvalue、pinksn、pinipek、pinipekcheckValue
Outputs	The result of updating IPEK
Description	Update IPEK

Example	[pos doUpdateIPEKOperation:@"00"
	tracksn:@"00000510F462F8400004"
	trackipek:@"293C2D8B1D7ABCF83E665A7C5C6532C9"
	trackipekCheckValue:@"93906AA157EE2604"
	emvksn:@"00000510F462F8400004"
	emvipek:@"293C2D8B1D7ABCF83E665A7C5C6532C9"
	emvipekcheckvalue:@"93906AA157EE2604"
	pinksn:@"00000510F462F8400004"
	pinipek:@"293C2D8B1D7ABCF83E665A7C5C6532C9"
	pinipekcheckValue:@"93906AA157EE2604"
	block:^(BOOL isSuccess, NSString *stateStr) {
	    if (isSuccess) {
	        self.textViewLog.text = stateStr;
	    }
	}];

2.6.74 updateIPEKOperationByKeyType

Signature	- (void)updateIPEKOperationByKeyType:(NSString *)groupKey tracksn:(NSString *)trackksn trackipek:(NSString *)trackipek trackipekCheckValue:(NSString *)trackipekCheckValue emvksn:(NSString *)emvksn emvipek:(NSString *)emvipek emvipekcheckvalue:(NSString *)emvipekcheckvalue

	pinksn:(NSString *)pinksn pinipek:(NSString *)pinipek pinipekcheckValue:(NSString *)pinipekcheckValue block:(void(^)(BOOL isSuccess,NSString *stateStr))EMVBlock;
Inputs	groupKey、tracksn、trackipek、trackipekCheckValue、emvksn、emvipek、emvipekcheckvalue、pinksn、pinipek、pinipekcheckValue
Outputs	The result of updating IPEK
Description	Update IPEK

2.6.75 doSetBuzzerOperation

Signature	-(void)doSetBuzzerOperation:(NSInteger)timeOut block:(void (^)(BOOL isSuccess, NSString*stateStr))buzzerBlock;
Inputs	timeOut
Outputs	The result of setting buzzer
Description	Set buzzer operation

2.6.76 updateEMVRID

Signature	-(void)updateEMVRID:(NSInteger)operationType data:(NSString *)data block:(void(^)(BOOL isSuccess,NSString *stateStr))EMVBlock;
Inputs	operationType data
Outputs	Update result
Description	Update emv RID configuration

See also	onReturnUpdateEMVRIDResult

2.6.77 updateAID

Signature	-(void)updateAID:(NSInteger)operationType data:(NSString *)data block:(void(^)(BOOL isSuccess,NSString *stateStr))EMVBlock;
Inputs	operationType data
Outputs	Update result
Description	Update emv AID configuration

2.6.78 setAIDwithBool

Signature	-(void)setAIDwithBool:(BOOL)isTrue data:(NSString *)data block:(void(^)(BOOL isSuccess,NSString *stateStr))EMVBlock
Inputs	isTrue data
Outputs	Update result
Description	Setting AID

2.6.79 setOnlineTime

Signature	-(void)setOnlineTime:(NSInteger)aTime
Inputs	aTime
Outputs	None

Description	Set the time to wait for ARPC

2.6.80 getUpdateCheckValueBlock

Signature	-(void)getUpdateCheckValueBlock:(void(^)(BOOL isSuccess,NSString *stateStr))updateCheckValueBlock
Inputs	None
Outputs	upgrade key
Description	Read upgrade key

2.6.81 doSetShutDownTime

Signature	-(void)doSetShutDownTime:(NSString *)timeOut
Inputs	timeOut
Outputs	None
Description	Set shutdown time
See also	onReturnSetSleepTimeResult

2.6.82 getBluetoothState

Signature	-(BOOL)getBluetoothState
Inputs	None
Outputs	Bluetooth State
Description	Get Bluetooth State

2.6.83 setBTAutoDetecting

Signature	-(void)setBTAutoDetecting: (BOOL)flag
Inputs	flag
Outputs	None
Description	Set BT Auto Detecting

2.6.84 getConnectedPeripheral

Signature	-(CBPeripheral*)getConnectedPeripheral:(NSString *)bluetoothName;
Inputs	Bluetooth Name
Outputs	Peripheral
Description	Get Connected Peripheral

2.6.85 connectBluetoothByCBPeripheral

Signature	-(BOOL)connectBluetoothByCBPeripheral: (CBPeripheral*)myCBPeripheral;
Inputs	CBPeripheral
Outputs	Connected result
Description	Connect Bluetooth By CBPeripheral

2.6.86 connectBluetoothNoScan

Signature	-(BOOL)connectBluetoothNoScan: (NSString*)bluetoothName;
Inputs	bluetoothName
Outputs	Connected result

Description	You can connect bluetooth without scanning bluetooth

2.6.87 getConnectedDevices

Signature	-(NSArray *)getConnectedDevices;
Inputs	None
Outputs	Connected Devices
Description	Get Connected Devices

2.6.88 cancelTrade

Signature	-(BOOL)cancelTrade:(BOOL)isUserCancel;
Inputs	isUserCancel
Outputs	The result of canceling trade
Description	Cancel Trade

2.6.89 asynresetPosStatus

Signature	-(void)asynresetPosStatus;
Inputs	None
Outputs	None
Description	Asyn reset Pos Status
See also	onAsyncResetPosStatus

2.6.90 setDoTradeMode

Signature	-(void)setDoTradeMode:(DoTradeMode)doTradeMode;
Inputs	doTradeMode
Outputs	None
Description	set DoTradeMode before calling doTrade

2.6.91 setFormatID

Signature	-(void)setFormatID:(NSString *)formatID;
Inputs	formatID
Outputs	None
Description	Set format ID before calling doTrade

2.6.92 setAmountPoint

Signature	-(void)setAmountPoint:(BOOL)amoutPoint;
Inputs	amoutPoint
Outputs	None
Description	Set Amount Point before calling doTrade

2.6.93 doTrade_QF

Signature	-(void)doTrade_QF:(NSInteger)tradeMode TradeRandomString:(NSString *)randomString TradeExtraString:(NSString *)extraString timeout:(NSInteger) delay;

Inputs	tradeMode 、randomString、extraString、delay
Outputs	None
Description	Do trade for QF

2.6.94 setIsQuickEMV

Signature	-(void)setIsQuickEMV:(BOOL)isQuickEMV;
Inputs	isQuickEMV
Outputs	None
Description	Set quick emv before calling doTrade

2.6.95 getQuickEMV

Signature	-(BOOL)getQuickEMV;
Inputs	None
Outputs	QuickEMV
Description	Get Quick EMV

2.6.96 getNFCBatchData

Signature	-(NSDictionary *)getNFCBatchData;
Inputs	None
Outputs	NFC Batch Data
Description	You can use it to get NFC batch data

2.6.97 powerOffNFC

Signature	-(void)powerOffNFC:(NSInteger) timeout withBlock:(void (^)(BOOL isSuccess))onPowerOffNFCResultBlock;
Inputs	timeout
Outputs	Power Off NFC Result
Description	Power Off NFC

2.6.98 sendPinEntryResult

Signature	-(void)sendPinEntryResult:(NSString *)pin;
Inputs	pin
Outputs	None
Description	Send Pin Entry Result to terminal

2.6.99 cancelPinEntry

Signature	-(void)cancelPinEntry;
Inputs	None
Outputs	None
Description	Cancel Pin Entry

2.6.100 bypassPinEntry

Signature	-(void)bypassPinEntry;
Inputs	None

Outputs	None
Description	Bypass Pin Entry

2.6.101 updateEMVConfigByXml

Signature	-(void)updateEMVConfigByXml:(NSString *)xmlStr;
Inputs	xmlStr
Outputs	None
Description	Update Emv configuration through xml file and add mac verification.
See also	onReturnCustomConfigResult

Example	NSData *xmlData = [self readLine:@"emv_profile_tlv"];
	NSString *xmlStr =[QPOSUtil asciiFormatString:xmlData];
	[pos updateEMVConfigByXml:xmlStr];

2.6.102 updateEmvAPPByTlv

Signature	-(void)updateEmvAPPByTlv:(EMVOperation)emvOperation appTlv:(NSString *)appTlv;
Inputs	emvOperation : EMVOperation_clear,//Clear all emv AID EMVOperation_add,//Add a set of emv AID EMVOperation_delete,//Delete a set of emv AID EMVOperation_getList,//Get AID list EMVOperation_update,// Update a specific tag EMVOperation_quickemv//Get the parameters of the specified AID

	appTlv
Outputs	None
Description	update emv configure by appTlv
See also	onReturnUpdateEMVResult onReturnGetEMVListResult

Example	[pos updateEmvAPPByTlv: EMVDataOperation_clear appTlv:@ ""];
	[pos updateEmvAPPByTlv: EMVDataOperation_add appTlv:@ "9F0607A0000000031010DF0101009F0902008C9F1B04000000009F1E0838333230314943439F6604B60040009F812701009F812801009F812904000000009F812A05D84000A8009F812B05001080000009F812C05D84004F8009F92810004050100009F92810101009F92810A02FE009F92810D0699999999999999F92810E06000000020000019F92810F06000000000000"];
	[pos updateEmvAPPByTlv: EMVDataOperation_delete appTlv:@ "9F0607A0000000031010"];
	[pos updateEmvAPPByTlv: EMVDataOperation_getList appTlv:@ ""];
	[pos updateEmvAPPByTlv: EMVDataOperation_update appTlv:@ "9F0607A00000000310109F3303E0F8C8"];
	[pos updateEmvAPPByTlv: EMVDataOperation_quickemv appTlv:@ "9F0607A0000000031010"];

2.6.103 updateEmvCAPKByTlv

Signature	-(void)updateEmvCAPKByTlv:(EMVOperation)emvOperation capkTlv:(NSString *)capkTlv;
Inputs	capkTlv
Outputs	None

Description	update emv configure by capkTlv
See also	onReturnUpdateEMVRIDResult onReturnGetEMVListResult

2.6.104 updateEmvAPP(Deprecated)

Signature	-(void)updateEmvAPP:(NSInteger)operationType data:(NSArray*)data block:(void (^)(BOOL isSuccess, NSString *stateStr))updateCAPKBlock;
Inputs	operationType、data
Outputs	Update result
Description	Update emv app configure
See also	onReturnUpdateEMVResult onReturnGetEMVListResult

2.6.105 updateEmvCAPK(Deprecated)

Signature	-(void)updateEmvCAPK:(NSInteger)operationType data:(NSArray *)data block:(void (^)(BOOL isSuccess, NSString *stateStr))updateCAPKBlock;
Inputs	operationType、data
Outputs	Update result
Description	update emv capk configure
See also	onReturnUpdateEMVRIDResult onReturnGetEMVListResult

2.6.106 lcdShowCustomDisplay

Signature	-(void)lcdShowCustomDisplay:(LcdModeAlign) alcdModeAlign lcdFont:(NSString *)alcdFont delay:(NSInteger)timeout;
Inputs	alcdModeAlign、alcdFont、timeout
Outputs	None
Description	you can use it to show custom text on pos
See also	onLcdShowCustomDisplay

2.6.107 lcdShowCloseDisplay

Signature	-(void)lcdShowCloseDisplay;
Inputs	None
Outputs	None
Description	Close the function of lcdShowCustomDisplay

2.6.108 getUpdateProgress

Signature	-(NSInteger)getUpdateProgress;
Inputs	None
Outputs	None
Description	you can use it to get progress of updating pos firmware

2.6.109 updatePosFirmware

Signature	-(NSInteger)updatePosFirmware:(NSData*)aData address:(NSString*)devAddress;

Inputs	Firmware data、bluetooth address
Outputs	None
Description	you can use it to update pos firmware
See also	onUpdatePosFirmwareResult

2.6.110 syncGetPin

Signature	-(NSDictionary*)syncGetPin:(NSInteger)encryptType keyIndex:(NSInteger)keyIndex maxLen:(NSInteger)maxLen typeFace:(NSString *)typeFace cardNo:(NSString *)cardNo date:(NSString *)data delay:(NSInteger)timeout;
Inputs	encryptType、keyIndex、maxLen、typeFace、cardNo、Data、timeout
Outputs	Get pin result
Description	Sync get pin block and pin ksn

2.6.111 getICCTag

Signature	-(NSDictionary *)getICCTag:(EncryptType)encryTypeStr cardType:(NSInteger)cardType tagCount:(NSInteger) mTagCount tagArrStr:(NSString*)mTagArrStr;
Inputs	encryTypeStr: EncryptType_plaintext, EncryptType_encrypted cardType: “0” for ICC and “1” for NFC mTagCount: the number of the tag mTagArrStr: string spliced by the tag
Outputs	Value of special emv tag
Description	you can use it to get value of special emv tag

Example	[pos getICCTag: EncryptType_encrypted cardType:1 mTagCount:2 mTagArrStr:@"5F205F24"];

2.6.112 customInputDisplay

Signature	-(void)customInputDisplay:(NSInteger)operationType displayType:(NSInteger)dispType maxLen:(NSInteger)maxLen DisplayString:(NSString *)displayStr delay:(NSInteger)timeout withResultBlock:(void (^)(BOOL isSuccess, NSString * result))customInputDisplayResult;
Inputs	operationType 、dispType 、maxLen、displayStr 、timeout
Outputs	Custom Input Display Result
Description	you can use it to custom input on pos

2.6.113 isCardExist

Signature	-(void)isCardExist:(NSInteger)timeout withResultBlock:(void (^)(BOOL))isCardExistBlock;
Inputs	timeout
Outputs	The result of checking card if exist
Description	Check card if exist

2.6.114 isCardExistInOnlineProcess

Signature	-(void)isCardExistInOnlineProcess:(NSInteger)timeout withResultBlock:(void (^)(BOOL))isCardExistBlock;
Inputs	timeout
Outputs	The result of Card Exist

Description	Check card if exist in this method “onRequestOnlineProcess”

2.6.115 cbc_mac_cn_all

Signature	-(void)cbc_mac_cn_all:(NSInteger)keyLen atype:(NSInteger)algorithmType otype:(NSInteger)operatorType data:(NSString *)dataStr delay:(NSInteger)timeout withResultBlock:(void (^)(NSString *))cbcmacBlock;
Inputs	keyLen、algorithmType、operatorType、dataStr、timeout
Outputs	The result of calculating mac
Description	Calculate mac

2.6.116 getKsn

Signature	-(void)getKsn:(void(^)(BOOL isSuccess,NSDictionary *dict))getKsnBlock;
Inputs	None
Outputs	Ksn
Description	Get Ksn

2.6.117 getKeyCheckValue

Signature	-(void)getKeyCheckValue:(CHECKVALUE_KEYTYPE)checkValueType keyIndex:(NSInteger)keyIndex;
Inputs	checkValueType、keyIndex
Outputs	None
Description	Get Key Check Value

See also	onGetKeyCheckValue

2.6.118 setBuzzerStatus

Signature	-(void)setBuzzerStatus:(NSInteger)status;
Inputs	status
Outputs	None
Description	Set Buzzer Status
See also	onReturnBuzzerStatusResult

2.6.119 setAESKey

Signature	-(void)setAESKey:(NSString *)AESCiphertext CRC:(NSString *)CRC timeout:(NSInteger)timeout;
Inputs	AESCiphertext 、 CRC、timeout
Outputs	None
Description	Set AES Key
See also	onReturnSetAESResult

2.6.120 getAESTransmissionKey

Signature	-(void)getAESTransmissionKey:(NSInteger)timeout;
Inputs	timeout
Outputs	None
Description	Get AES Transmission Key

See also	onReturnAESTransmissonKeyResult

2.6.121 getShutDownTime

Signature	-(void)getShutDownTime;
Inputs	None
Outputs	None
Description	Get Shut Down Time
See also	onGetShutDownTime

2.6.122 setPanStatus

Signature	-(void)setPanStatus:(NSInteger)panStatus;
Inputs	PanStatus_DEFAULT PanStatus_PLAINTEXT PanStatus_ENCRYPTED
Outputs	None
Description	Set Pan Status

2.6.123 getDevicePublicKey

Signature	-(void)getDevicePublicKey:(NSInteger)timeout;
Inputs	timeout
Outputs	None
Description	Get Device Public Key

See also	onGetDevicePublicKey

2.6.124 setShutDownTimeOnConnected

Signature	-(void)setShutDownTimeOnConnected:(NSInteger)time;
Inputs	time
Outputs	None
Description	Set Shut Down Time On Connected
See also	onReturnSetConnectedShutDownTimeResult

2.6.125 getShutDownTimeOnConnected

Signature	-(void)getShutDownTimeOnConnected;
Inputs	None
Outputs	None
Description	Get Shut Down Time On Connected
See also	onReturnGetConnectedShutDownTimeResult

2.6.126 getCvmPinTryLimit

Signature	-(NSInteger)getCvmPinTryLimit;
Inputs	None
Outputs	Pin Try Limit
Description	Get Cvm Pin Try Limit

2.6.127 updateKeyByTR_31

Signature	-(void)updateKeyByTR_31:(NSInteger)keyIndex keyBlock:(NSString *)keyBlock; -(void)updateKeyByTR_31:(NSInteger)keyIndex keyBlock:(NSString *)keyBlock timeout:(NSInteger)timeout;
Inputs	keyIndex、keyBlock timeout
Outputs	None
Description	Update Key By TR_31
See also	onReturnUpdateKeyByTR_31Result

2.6.128 generateSessionKeys

Signature	-(void)generateSessionKeys;
Inputs	None
Outputs	None
Description	Generate Session Keys
See also	onQposGenerateSessionKeysResult

2.6.129 updateRSA

Signature	-(void)updateRSA:(NSString *)publicKey pemFile:(NSString *)pemFile;
Inputs	publicKey、pemFile
Outputs	None

Description	Update RSA
See also	onDoSetRsaPublicKey

2.6.130 getEncryptData

Signature	-(void)getEncryptData:(NSData *)data keyType:(NSString*)keyType keyIndex:(NSString *)keyIndex timeOut:(NSInteger)timeout;
Inputs	Data、keyType、keyIndex、timeout
Outputs	None
Description	Get Encryp tData
See also	onReturnGetEncryptDataResult

2.6.131 getMPUCardInfo

Signature	(void)getMPUCardInfo:(void(^)(NSDictionary *dict))mpuInfoBlock;
Inputs	None
Outputs	MPU card Info
Description	Get MPU Card Info

2.6.132 getMIccCardData

Signature	-(void)getMIccCardData:(NSString *)transactionTime;
Inputs	transactionTime
Outputs	None
Description	Get ICC Data

See also	onRequestBatchData

2.6.133 updateKeyByTR_31VersionD

Signature	-(void)updateKeyByTR_31VersionD:(NSInteger)keyIndex ksn:(NSString *)ksn keyBlock:(NSString *)keyBlock;
Inputs	keyIndex、ksn、keyBlock
Outputs	Update result
Description	Update Key By TR_31VersionD

2.6.134 powerOnFelica

Signature	-(void)powerOnFelica:(NSInteger)timeout;
Inputs	timeout
Outputs	None
Description	Power On Felica
See also	onReturnPowerOnFelicaResult

2.6.135 powerOffFelica

Signature	-(void)powerOffFelica:(NSInteger)timeout;
Inputs	timeout
Outputs	None
Description	Power Off Felica
See also	onReturnPowerOffFelicaResult

2.6.136 sendApduByFelica

Signature	-(void)sendApduByFelica:(NSString *)apduString timeout:(NSInteger)timeout;
Inputs	apduString、timeout
Outputs	None
Description	Send Apdu By Felica
See also	onReturnPowerOffFelicaResult

2.6.137 generateTransportKey

Signature	-(void)generateTransportKey:(NSInteger)timeout dataBlock:(void(^)(NSDictionary *))dataBlock;
Inputs	timeout
Outputs	Transport Key
Description	Generate Transport Key

2.6.138 sendCVV

Signature	-(void)sendCVV:(NSString *)cvvStr resultBlock:(void(^)(BOOL))resultBlock;
Inputs	cvvStr
Outputs	Send CVV result
Description	Send CVV to terminal

2.6.139 getEncryptedDataBlock

Signature	-(void)getEncryptedDataBlock:(NSInteger)keyIndex dataBlock:(void(^)(NSDictionary *))dataBlock;

Inputs	keyIndex
Outputs	Encrypted Data Block
Description	Get Encrypted Data Block

2.6.140 setIsSaveLog

Signature	-(void)setIsSaveLog:(BOOL)IsSaveLog block:(void(^)(BOOL isSuccess,NSString *stateStr))IsSaveLogBlock;
Inputs	IsSaveLog
Outputs	result
Description	Set Is Save Log

2.6.141 doTradeLogOperation

Signature	-(void)doTradeLogOperation:(NSInteger)operationType data:(NSInteger)data block:(void(^)(BOOL isSuccess,NSInteger markType, NSDictionary *stateStr))doTradeLogBlock;
Inputs	operationType data
Outputs	Get Trade Log
Description	Get Trade Log Operation

2.6.142 operateLEDByType

Signature	- (void)operateLEDByType:(LEDType)ledType colorValue:(NSString *)colorValue ledDirection:(LEDDirection)ledDirection ledStatus:(LEDStatus)ledStatus

	lightTime:(NSInteger)lightTime lightOffTime:(NSInteger)lightOffTime blinksTimes:(NSInteger)blinksTimes resultBlock:(void(^)(BOOL isSuccess))resultBlock;
Inputs	ledType: fixed color or RGB colorValue: fixed value (0x0000 Blue、0x0001 Yellow、0x0002 Green、0x0003 Red) or RGB value ledDirection: up、down、left、right、all ledStatus: on、off、blink lightTime: only for blink lightOffTime: only for blink blinksTimes:only for blink
Outputs	resultBlock
Description	Operate LED By Type

2.6.143 playBuzzerByType

Signature	- (void)playBuzzerByType:(BuzzerType)buzzerType buzzerOnTime:(NSInteger)buzzerOnTime buzzerOffTime:(NSInteger)buzzerOffTime buzzerTimes:(NSInteger)buzzerTimes resultBlock:(void(^)(BOOL isSuccess))resultBlock;
Inputs	buzzerType: buzzer control、payment success、payment error、pairing success buzzerOnTime: only for buzzer control buzzerOffTime: only for buzzer control buzzerTimes: only for buzzer control
Outputs	resultBlock
Description	Play Buzzer By Type

2.6.144 pollOnMifareCard

Signature	-(void)pollOnMifareCard:(NSInteger)timeout dataBlock:(void(^)(NSDictionary *))dataBlock;
Inputs	timeout
Outputs	dataBlock
Description	Poll on mifare card

2.6.145 authenticateMifareCard

Signature	-(void)authenticateMifareCard:(MifareCardType)mifareCardType keyType:(MifareKeyType)keyType block:(NSString *)block keyValue:(NSString *)keyValue timeout:(NSInteger)timeout resultBlock:(void(^)(BOOL isSuccess))resultBlock;
Inputs	mifareCardType：CLASSIC、ULTRALIGHT keyType：KEY_A 、 KEY_B block: block address timeout
Outputs	resultBlock
Description	Authenticate mifare card

2.6.146 operateMifareCardData

Signature	-(void)operateMifareCardData:(MifareCardOperationType)operationType block:(NSString *)block data:(NSString *)data timeout:(NSInteger)timeout dataBlock:(void(^)(NSDictionary *))dataBlock;
Inputs	operationType: ADD、REDUCE、RESTORE

	block：block address data timeout
Outputs	dataBlock
Description	Operate mifare card

2.6.147 readMifareCard

Signature	-(void)readMifareCard:(MifareCardType)mifareCardType block:(NSString *)block timeout:(NSInteger)timeout dataBlock:(void(^)(NSDictionary *))dataBlock;
Inputs	mifareCardType：CLASSIC、ULTRALIGHT block：block address timeout
Outputs	dataBlock
Description	Read mifare card

2.6.148 writeMifareCard

Signature	-(void)writeMifareCard:(MifareCardType)mifareCardType block:(NSString *)block data:(NSString *)data timeout:(NSInteger)timeout resultBlock:(void(^)(BOOL isSuccess))resultBlock;
Inputs	mifareCardType：CLASSIC、ULTRALIGHT block：block address data timeout
Outputs	resultBlock

Description	Write mifare card

2.6.149 finishCard

Signature	-(void)finishCard:(NSInteger)timeout resultBlock:(void(^)(BOOL isSuccess))resultBlock;
Inputs	timeout
Outputs	resultBlock
Description	Finsh mifare card operation

2.6.150 setIsOperateMifare

Signature	-(void)setIsOperateMifare:(BOOL)isOperateMifare;
Inputs	BOOL(True or False) True is value block and False is data block
Outputs	None
Description	Set mifare operation is value block or data block

2.6.151 setIsSupportClsSelectEmvApp

Signature	-(void)setIsSupportClsSelectEmvApp:(BOOL)isSupportClsSelectEmvApp;
Inputs	BOOL(True or False)
Outputs	None
Description	Set contactless if support “select Emv APP” or not

2.6.152 getEncryptDataDict

Signature	-(NSDictionary *)getEncryptDataDict;
Inputs	None
Outputs	RandomData、AESKey、PAN、IsOnlinePin、PinTryLimit
Description	Without keyboard devices (eg: CR100 and Qpos reader) will use this key when to encrypt PIN.

2.6.153 sendCvmPin

Signature	-(void)sendCvmPin:(NSString *)pin isEncrypted:(BOOL)isEncrypted;
Inputs	BOOL(True or False)
Outputs	None
Description	If input is True will send encrypt pinblock (ISO-4) to terminal and if not will send plaintext PIN.

2.6.154 getCvmKeyList

Signature	-(NSString *)getCvmKeyList;
Inputs	None
Outputs	random key list
Description	Get the random key list from terminal which sdk can use this list to look up the corresponding index for PIN and send it to terminal.

2.7 Delegate Methods Reference

2.7.1 onRequestWaitingUser

Signature	-(void) onRequestWaitingUser
Inputs	None
Outputs	None
Description	In the callback to waiting user operating
See also	

2.7.2 onQposIdResult

Signature	-(void) onQposIdResult: (NSDictionary*)posId
Inputs	None
Outputs	posId;
Description	Callback of pos id response
See also	getQposId

2.7.3 onQposInfoResult

Signature	-(void) onQposInfoResult: (NSDictionary*)posInfoData
Inputs	None
Outputs	POS Information Data
Description	Callback of getQposInfo response
See also	getQposInfo

2.7.4 onDoTradeResult

Signature	-(void) onDoTradeResult: (DoTradeResult)result DecodeData:(NSDictionary*)decodeData
Inputs	None
Outputs	result decodeData
Description	The callback of transaction data
See also	doTrade doEmvApp: (EmvOption)aemvOption

2.7.5 onRequestSetAmount

Signature	-(void) onRequestSetAmount
Inputs	None
Outputs	None
Description	The callback of request user set amount
See also	doTrade() -(void) setAmount: (NSString *)aAmount aAmountDescribe:(NSString *)aAmountDescribe currency:(NSString *)currency transactionType:(TransactionType)transactionType

2.7.6 onRequestSelectEmvApp

Signature	-(void) onRequestSelectEmvApp: (NSArray*)appList
Inputs	None
Outputs	appList

Description	Callback method of request user to select emv app
See also	doTrade； -(void) selectEmvApp: (NSInteger)index; -(void) cancelSelectEmvApp;

2.7.7 onRequestIsServerConnected

Signature	-(void) onRequestIsServerConnected
Inputs	None
Outputs	None
Description	In the method judge the network state
See also	doTrade -(void) isServerConnected: (BOOL)isConnected;

2.7.8 onRequestFinalConfirm

Signature	-(void) onRequestFinalConfirm;
Inputs	None
Outputs	None
Description	Final confirm amount response
See also	doTrade -(void) finalConfirm: (BOOL)isConfirmed;

2.7.9 onRequestOnlineProcess

Signature	-(void) onRequestOnlineProcess: (NSString*) tlv
Inputs	None

Outputs	Tlv tag+length+value
Description	Response of request data to server
See also	doTrade anlysEmvIccData sendOnlineProcessResult

2.7.10 onRequestTime

Signature	-(void) onRequestTime
Inputs	None
Outputs	None
Description	Set terminal time response
See also	doTrade sendTime

2.7.11 onRequestTransactionResult

Signature	-(void) onRequestTransactionResult: (TransactionResult)transactionResult
Inputs	None
Outputs	None
Description	Transaction response
See also	doTrade

2.7.12 onRequestTransactionLog

Signature	-(void) onRequestTransactionLog: (NSString*)tlv
Inputs	None

Outputs	Tlv tag+length+value
Description	reserve
See also	

2.7.13 onRequestBatchData

Signature	-(void) onRequestBatchData: (NSString*)tlv
Inputs	None
Outputs	Tlv tag+length+value
Description	Response of ICC transactions
See also	doTrade

2.7.14 onRequestQposConnected

Signature	-(void) onRequestQposConnected
Inputs	None
Outputs	None
Description	Callback of connected success
See also	connectBluetoothDevice

2.7.15 onRequestQposDisconnected

Signature	-(void) onRequestQposDisconnected
Inputs	None
Outputs	None
Description	Callback of disconnect
See also	disconnectBT

2.7.16 onRequestNoQposDetected

Signature	-(void) onRequestNoQposDetected
Inputs	None
Outputs	None
Description	Callback of connected fail
See also	connectBT

2.7.17 onError(Deprecated)

Signature	-(void) onError: (Error)errorState
Inputs	None
Outputs	enum Error errorState error type information
Description	SDK error information response
See also	

2.7.18 onRequestDisplay

Signature	-(void) onRequestDisplay: (Display)displayMsg
Inputs	None
Outputs	enum Display displayMsg: display content
Description	Notify the terminal displays the current related content
See also	

2.7.19 onRequestUpdateWorkKeyResult

Signature	-(void) onRequestUpdateWorkKeyResult: (UpdateInformationResult)updateInformationResult
Inputs	None
Outputs	enum UpdateInformationResult result
Description	Update work key response
See also	udpateWorkKey

2.7.20 onGetCardNoResult

Signature	-(void) onGetCardNoResult:(NSString *)result
Inputs	None
Outputs	String result card no
Description	Get Card number response
See also	getCardNo

2.7.21 onReturnReversalData

Signature	-(void) onReturnReversalData: (NSString*)tlv
Inputs	None
Outputs	tlv : tag+length+value
Description	ic card reversal data response
See also	doEmvApp

2.7.22 onReturnGetPinResult

Signature	-(void) onReturnGetPinResult:(NSDictionary*)decodeData

Inputs	None
Outputs	pinKsn , pinBlock
Description	Get pin response
See also	getPin

2.7.23 onReturnPowerOnIccResult

Signature	-(void) onReturnPowerOnIccResult:(BOOL) isSuccess KSN:(NSString *) ksn ATR:(NSString *)atr ATRLen:(NSInteger)atrLen
Inputs	None
Outputs	result: true － success, false － failure ksn: EMV KSN used for encryption of ATR data and APDU data. If ksn is all FF, then ATR and APDU data are not encrypted. atr: data returned in ATR atrLen: length of the ATR data
Description	Respond to powerOnIcc. If ksn is all FF, then ATR and APDU are not encrypted. Otherwise, ATR and APDU are encrypted by the key derived from EMV KSN.
See also	powerOnIcc

2.7.24 onReturnPowerOffIccResult

Signature	-(void) onReturnPowerOffIccResult:(BOOL) isSuccess
Inputs	None
Outputs	result: true － success, false － failure

Description	Respond to powerOffIcc.
See also	powerOffIcc

2.7.25 onReturnApduResult

Signature	-(void) onReturnApduResult:(BOOL)isSuccess APDU:(NSString *)apdu APDU_Len:(NSInteger) apduLen
Inputs	None
Outputs	isSuccess: true – success, false – failure apdu: data returned apduLength: length of the apdu data
Description	Return data in response to the level 1 EMV method sendApdu. If the apdu data are encrypted, the KSN returned after the powerOnIcc command is used for encryption.
See also	sendApduByNFC sendApdu

2.7.26 onReturnSetSleepTimeResult

Signature	-(void)onReturnSetSleepTimeResult:(BOOL)isSuccess
Inputs	None
Outputs	isSuccess: true – success, false – failure
Description	The response of set device sleep time
See also	setPosSleepTime

2.7.27 onRequestCalculateMac

Signature	-(void)onRequestCalculateMac:(NSString *)calMacString
Inputs	None
Outputs	calMac: calculate mac result data
Description	Calculate mac response method
See also	doCalculateMac doDoubleMac doSingleMac

2.7.28 onReturnCustomConfigResult

Signature	-(void)onReturnCustomConfigResult:(BOOL)isSuccess config:(NSString*)resutl
Inputs	None
Outputs	isSuccess: true – success, false – failure result : data of result
Description	Save Emv Config response
See also	updateEmvConfig

2.7.29 onReturnSetMasterKeyResult

Signature	-(void) onReturnSetMasterKeyResult: (BOOL)isSuccess
Inputs	None
Outputs	isSuccess: true – success, false – failure
Description	Set masterKey response
See also	setMasterKey

2.7.30 onReturnBatchSendAPDUResult

Signature	-(void) onReturnBatchSendAPDUResult:(NSDictionary *)apduResponses
Inputs	None
Outputs	apduResponses
Description	Response of batch APDU instruction execution
See also	VIPOSBatchSendAPDU

2.7.31 onReturniccCashBack

Signature	-(void) onReturniccCashBack: (NSDictionary*)result
Inputs	None
Outputs	result
Description	Cash back response（Special needs）
See also	iccCashBack getIccCardNo inquireECQAmount

2.7.32 onUpdatePosFirmwareResult

Signature	-(void) onUpdatePosFirmwareResult:(UpdateInformationResult)result
Inputs	None
Outputs	enum UpdateInformationResult result
Description	Response of upgrade the firmware

See also	updatePosFirmware

2.7.33 onReturnDownloadRsaPublicKey

Signature	-(void) onDownloadRsaPublicKeyResult:(NSDictionary *)result
Inputs	None
Outputs	result
Description	Callback of encrypt random number using RSA public key
See also	downloadRsaPublicKey

2.7.34 onPinKey_TDES_Result

Signature	-(void) onPinKeyTDESResult:(NSString *)encPin
Inputs	None
Outputs	result: data of response
Description	Response of 3DES encrypt pin
See also	pinKey_TDES

2.7.35 onUpdateMasterKeyResult

Signature	-(void) onUpdateMasterKeyResult:(BOOL)isSuccess aDic:(NSDictionary *)resultDic
Inputs	None
Outputs	(BOOL)isSuccess aDic:(NSDictionary *)resultDic
Description	Update masterKey response
See also	updateMasterKeyRandom updateMasterKey

2.7.36 onEmvICCExceptionData

Signature	void onEmvICCExceptionData (String tlv)
Inputs	None
Outputs	Tlv : tag+length+value
Description	Emv ICC Exception response
See also	

2.7.37 onReturnUpdateEMVRIDResult

Signature	-(void)onReturnUpdateEMVRIDResult:(BOOL)isSuccess
Inputs	None
Outputs	result: true – success, false – fail
Description	The response of updateEMVRID

2.7.38 onLcdShowCustomDisplay

Signature	-(void)onLcdShowCustomDisplay: (BOOL)isSuccess
Inputs	None
Outputs	result: true – success, false – failure
Description	The response of lcdShowCustomDisplay

2.7.39 onGetKeyCheckValue

Signature	-(void)onGetKeyCheckValue:(NSDictionary *)checkValueResult
Inputs	None
Outputs	checkValueResult
Description	The response of Get Key Check Value

2.7.40 onGetShutDownTime

Signature	-(void)onGetShutDownTime:(NSString *)time
Inputs	None
Outputs	time
Description	The response of get Shut Down Time

2.7.41 onGetDevicePublicKey

Signature	-(void)onGetDevicePublicKey:(NSString *)clearKeys
Inputs	None
Outputs	clearKeys
Description	The response of get Device Public Key

2.7.42 onReturnUpdateKeyByTR_31Result

Signature	-(void)onReturnUpdateKeyByTR_31Result:(BOOL)result
Inputs	None

Outputs	result: true － success, false － failure
Description	The response of update key by TR_31

2.7.43 onQposGenerateSessionKeysResult

Signature	-(void)onQposGenerateSessionKeysResult:(NSDictionary *)result
Inputs	None
Outputs	result
Description	The response of update key by TR_31

2.7.44 onReturnGetEncryptDataResult

Signature	-(void)onReturnGetEncryptDataResult:(NSDictionary*)tlv
Inputs	None
Outputs	Tlv: tag+length+value
Description	The response of get encrypt data

2.7.45 onReturnPowerOnFelicaResult

Signature	-(void)onReturnPowerOnFelicaResult:(FelicaStatusCode)result
Inputs	None
Outputs	result
Description	The response of Power On Felica

2.7.46 onReturnPowerOffFelicaResult

Signature	-(void)onReturnPowerOffFelicaResult:(FelicaStatusCode)result
Inputs	None
Outputs	result
Description	The response of Power Off Felica

2.7.47 onReturnSendApduFelicaResult

Signature	-(void)onReturnSendApduFelicaResult:(FelicaStatusCode)result responseLen:(NSString *)responseLen responseData:(NSString *)responseData
Inputs	None
Outputs	Result、responseLen、responseData
Description	The response of send apdu Felica

2.7.48 onAsyncResetPosStatus

Signature	-(void)onAsyncResetPosStatus:(NSString *)isReset
Inputs	None
Outputs	Reset
Description	The response of setPosSleepTime

2.7.49 onReturnUpdateEMVResult

Signature	-(void)onReturnUpdateEMVResult:(BOOL)isSuccess
Inputs	None
Outputs	result: true － success, false － failure
Description	The response of updateEmvAPPByTlv and updateEmvAPP

2.7.50 onReturnGetEMVListResult

Signature	-(void)onReturnGetEMVListResult:(NSString *)result
Inputs	None
Outputs	result
Description	The response of updateEmvAPPByTlv &updateEmvCAPKByTlv&updateEmvAPP&updateEmvCAPK

2.7.51 onReturnBuzzerStatusResult

Signature	-(void)onReturnBuzzerStatusResult:(BOOL)isSuccess
Inputs	None
Outputs	result: true － success, false － failure
Description	callback of Setting whether the buzzer

2.7.52 onReturnSetAESResult

Signature	-(void)onReturnSetAESResult:(BOOL)isSuccess resultStr:(NSString *)result

Inputs	None
Outputs	result: true – success, false – failure
Description	The response of set AES

2.7.53 onReturnAESTransmissonKeyResult

Signature	- (void)onReturnAESTransmissonKeyResult:(BOOL)isSuccess resultStr:(NSString *)result
Inputs	None
Outputs	result: true – success, false – failure
Description	The response of getAESTransmissionKey

2.7.54 onReturnSetConnectedShutDownTimeResult

Signature	- (void)onReturnSetConnectedShutDownTimeResult:(BOOL)isSuccess
Inputs	None
Outputs	result: true － success, false － failure
Description	The response of setShutDownTimeOnConnected

2.7.55 onReturnGetConnectedShutDownTimeResult

Signature	- (void)onReturnGetConnectedShutDownTimeResult:(NSString *)time
Inputs	None

Outputs	result: true – success, false – failure
Description	The response of getShutDownTimeOnConnected

2.7.56 onDoSetRsaPublicKey

Signature	-(void)onDoSetRsaPublicKey:(BOOL)result
Inputs	None
Outputs	result: true － success, false － failure
Description	The response of update RSA