

3/F, 7 KIC, 388 Songhu Rd, Yangpu District, Shanghai，China

SUNMI PAY SDK V2 Development Document

Shanghai Sunmi Tech Co.,Ltd

目录

[ 1. Revision History 4](#_Toc155184333)

[ 2. Abstract 17](#_Toc155184334)

[2.1 Introduction 17](#_Toc155184335)

[2.2 SDK Integration 17](#_Toc155184336)

[ 3. API 18](#_Toc155184337)

[3.1 SunmiPayKernel SDK Operation Object 18](#_Toc155184338)

[3.2 Public member variable 19](#_Toc155184339)

[3.3 Basic operation module 20](#_Toc155184340)

[3.4 Card operation module 33](#_Toc155184341)

[3.5 PinPad operation module (Unsupported for TOSS) 54](#_Toc155184342)

[3.6 Security operation module 62](#_Toc155184343)

[3.7 EMV operation module 93](#_Toc155184344)

[3.8 Tax operation module (Unsupported for TOSS) 108](#_Toc155184345)

[3.9 Device certificate manager module (Unsupported for TOSS) 108](#_Toc155184346)

[4. Error Code Definition 110](#_Toc155184347)

[ 5. Entity class 119](#_Toc155184348)

[5.1 EmvTermParamV2 – Terminal parameter entity class 119](#_Toc155184349)

[5.2 AidV2 – AID entity class 120](#_Toc155184350)

[5.3 CapkV2 – CAPK entity class 120](#_Toc155184351)

[5.4 EMVTransDataV2 – Transaction data entity class 121](#_Toc155184352)

[5.5 EMVCandidateV2 – EMV Application Candidate 121](#_Toc155184353)

[5.6 PinPadConfigV2 – Pinpad configuration entity class 121](#_Toc155184354)

[5.7 PinPadTextConfigV2 – Pinpad showing text configuration entity class 122](#_Toc155184355)

[5.8 PinPadDataV2 – Pinpad layout entity class 122](#_Toc155184356)

[5.9 PinPadDataV2Ex – Pinpad layout entity class(extended) 122](#_Toc155184357)

[5.9 DrlV2 – DRL LimitSet entity class 123](#_Toc155184358)

[5.10 RevocListV2 – RevocationList entity class 123](#_Toc155184359)

[ 6. Access permission 123](#_Toc155184360)

[6.1 Permission location 123](#_Toc155184361)

[6.2 Permission definition 123](#_Toc155184362)

[ 7. Appendix 125](#_Toc155184363)

[7.1 Aidl constants class（com.sunmi.pay.hardware.aidl.AidlConstants） 125](#_Toc155184364)

[7.2 PAN data interception 135](#_Toc155184365)

[7.3 Read card flow chart 136](#_Toc155184366)

[7.4 EMV trans flow chart 136](#_Toc155184367)

[7.5 Key inject/access flow chart 138](#_Toc155184368)

[7.6 Key system and key index range 138](#_Toc155184369)

[7.7 Key partition description 139](#_Toc155184370)

[7.8 Key check value param description 140](#_Toc155184371)

## 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Doc Version | Release | Changes | The appropriate SDK version |
| 3.0.0 | 2017/11/30 | 1st Draft | SunmiPaySDKService1.0.1 |
| 3.0.1 | 2018/1/22 | Revise Docs | SunmiPaySDKService1.0.3/SunmiPayHardwareService3.0.44 |
| 3.0.2 | 2018/2/23 | Add Error Code | SunmiPaySDKService1.0.4/SunmiPayHardwareService3.0.44 |
| 3.0.3 | 2018/3/21 | Add setSysParam(),  Add TransPreProcess() | SunmiPaySDKService1.0.6/SunmiPayHardwareService3.1.2 |
| 3.0.4 | 2018/4/12 | Add smartCardExchange()parameter definition. | SunmiPaySDKService1.0.6/SunmiPayHardwareService3.1.2 |
| 3.0.5 | 2018/5/3 | Support SM4 algorithm. | SunmiPayHardwareService\_v3.2.07 |
| 3.0.6 | 2018/5/18 | Add the SysParam constant definition of the keyword "RESERVED" on and off the buzzer instructions. | SunmiPayHardwareService\_v3.2.07 |
| 3.0.7 | 2018/6/5 | Add mifareReadBlock() blockData format definition | SunmiPayHardwareService\_v3.2.07 |
| 3.0.8 | 2018/8/16 | Support DUKPT. | SunmiPayHardwareService\_v3.2.18 |
| 3.0.9 | 2018/8/31 | Add detectCard() and read card info | SunmiPayHardwareService\_v3.2.19 |
| 3.1.0 | 2018/9/21 | Modify the description for apduCommand() | SunmiPayHardwareService\_v3.2.19 |
| 3.2.0 | 2018/10/12 | Support V2 api. | SunmiPayHardwareService\_v3.2.20 |
| 3.2.1 | 2018/10/22 | Add error code | SunmiPayHardwareService\_v3.2.20 |
| 3.2.2 | 2018/10/24 | Add API: smartCardExchange() and smartCardExchangeNISO() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.3 | 2018/11/01 | 1.Fix issue: wrong params order of dataEncrypt ()  2.Modify method comments in SunmiPayKernel | SunmiPayHardwareService\_v3.3.0 |
| 3.2.4 | 2018/11/02 | Modify Class EMVTransDataV2 | SunmiPayHardwareService\_v3.3.0 |
| 3.2.5 | 2018/11/05 | 1.add API: initEmvProcess()  2.Modify get/Set TLV data API  3.Modify EMV flow chart | SunmiPayHardwareService\_v3.3.0 |
| 3.2.6 | 2018/11/07 | 1.Delete fields of **EMVTransDataV2**  2.Update EMV flow chart | SunmiPayHardwareService\_v3.3.0 |
| 3.2.7 | 2018/11/15 | 1.Update describe of PinPadConfigV2  2.Modify the getKeyCheckValue description  3.EMV procedure add interfaces:  onAppFinalSelect(),importAppFinalSelectStatus().  4.remove interface apduCommandNISO() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.8 | 2018/11/16 | 1.change name smartCardExChangeNISO()to transmitApdu()  2.transmitApdu() return card data directly(remove fields len,swa,swb) | SunmiPayHardwareService\_v3.3.0 |
| 3.2.9 | 2018/11/19 | Modify interface transmitApduset, change max size of sendBuff, recvBuff to 255B | SunmiPayHardwareService\_v3.3.0 |
| 3.2.10 | 2018/11/22 | Change type of **certType** to **int** in interface EMVListenerV2. onCertVerify() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.11 | 2018/11/24 | 1.add interface EMVListenerV2. onRequestSignature(),importSignatureStatus()  2.add error code -50027(Signature error)  3.Modify interface buzzerOnDevice() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.12 | 2018/11/29 | 1.add interface  EMVListenerV2.onCardDataExchangeComplete() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.13 | 2018/12/04 | 1.add interface EMVListenerV2.  onConfirmationCodeVerified()  2.add error code -50028  3.update emv flow chart | SunmiPayHardwareService\_v3.3.0 |
| 3.2.14 | 2018/12/08 | 1.update interface EMVListenerV2.onWaitAppSelect(),change paramter type to List<EMVCandidateV2>  2.modify interface EMVOptV2.importOnlineProcStatus() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.15 | 2018/12/12 | 1.Add code depiction of callback EMVListenerV2.onTransResult()  2.Modify depiction of parameter **status** in interface **importOnlineProcStatus**() | SunmiPayHardwareService\_v3.3.0 |
| 3.2.16 | 2018/12/13 | Increase -20001—-20006 error code | SunmiPayHardwareService\_v3.3.0 |
| 3.2.17 | 2019/01/07 | Add RSA interfaces | SunmiPayHardwareService\_v3.3.16 |
| 3.2.18 | 2019/01/17 | 1.Add interface removeRSAKey()  2.Remove the T=0,T=1 description i-n interface transmitApdu() | SunmiPayHardwareService\_v3.3.17 |
| 3.2.19 | 2019/01/21 | Update the keyIndex params illustr-ation of RSA/Cetificate related interfaces | SunmiPayHardwareService\_v3.3.17 |
| 3.2.20 | 2019/03/18 | 1.Update interface saveKeyDukpt(), add code to generate initialize KSN  2.Add interface dukptGetInitKsn()  3.Update interface transmitApdu(),support contact card.  4.Update interface apduCommand(), modify fields Lc,Le,outLen value range of param ApduSendV2, ApduRecvV2 | SunmiPayHardwareService\_v3.3.28 |
| 3.2.21 | 2019/03/19 | 1.Update interface saveKeyDukpt(),remove generate initialize ksn code  2.Update interface dukptGetInitKSN()  3.Add interface EMVOptV2.abortTransactProcess() | SunmiPayHardwareService\_v3.3.29 |
| 3.2.22 | 2019/04/10 | 1.add MifarePlus SL3 card data  exchange interfaces  2.add SLE4442/4428 card data  exchange interfaces  3.add AT24C01/02/04/08/16/32/64/128/256/512 card data exchange  Interfaces  4.add embedded doc [**Sunmi PinBlock format**] | SunmiPayHardwareService\_v3.3.32 |
| 3.2.23 | 2019/05/29 | 1.add interface EMVOptV2.transactProcessEx(),EMVOptV2.importDataExchangeStatus()  2.add interface EMVListenerV2.onRequestDataExchange() | SunmiPayHardwareService\_v3.3.40 |
| 3.2.24 | 2019/07/06 | 1.add at88scxx card data exchange interfaces  2.add interface transmitApduEx(),support customized Miafre card APDU data exchange  3.remove firmware version of revision history column [The appropriate SDK version]  4.Add interface sysGetRandom()  5.Add RSA signing/verify signature interaces | SunmiPayHardwareService\_v3.3.46 |
| 3.2.25 | 2019/07/30 | 1.Add key inject interfaces injectPlaintextKey(),injectCiphertextKey()  2.CheckCardCallbackV2 add callback methods findICCardEx(),findRFCardEx(),onErrorEx()  3.Modify CheckCardCallbackV2. findMagCard(), add key cardType, trackErrorCode,etc in param info.  4.Add interface EMVOptV2.queryECBalance() | SunmiPayHardwareService\_v3.3.48 |
| 3.3.26 | 2019/09/16 | 1.Modify illustration of interface  checkCard()  2.AidV2 Add fields clsStatusCheck,  zeroCheck  3.EMVOptV2 add interfaces addDrlLimitSet(), deleteDrlLimitSet(), setTermParamEx() to support DRL | SunmiPayHardwareService\_v3.3.52 |
| 3.2.27 | 2019/10/15 | 1.Add system parameter constants: EMVVersion,PaypassVersion,PaywaveVersion,QPBOCVersion,EntryVersion,MirVersion,JCBVersion,PAGOVersion,EmvKernelCheckSum  2.Change the hit text of input PIN  3.Add Error code -70001~ -70006 | SunmiPayHardwareService\_v3.3.53 |
| 3.2.28 | 2019/10/28 | 1.EMVOptV2 add interface queryAidCapkLis()、transactPreProcess()  2.PinPadOptv2 add interface cancelInputPin()  3. add cardCategory, atqa in radio card check card call back. | SunmiPayHardwareService\_v3.3.55 |
| 3.2.29 | 2019/11/29 | 1. Security module add interfaces dataEncryptDukptEx(),dataDecryptDukptEx(),calcMacDukptEx(),verifyMacDukptEx(),saveTR31Key(),saveCiphertextKeyRSA(),saveRSAKey()  2.Add system Param key：AEVersion  3.Add new EMV flowType, value is 0x04  4.EMV module add interfaces  addRevocList()、deleteRevocList()、  sysSetTime()、sysGetTime()、clearData()  5.Modify param name in interfaces signingRSA(),verifySignatureRSA() | SunmiPayHardwareService\_v3.3.58 |
| 3.2.30 | 2020/02/19 | 1.AidV2 add fields kernelType,paramType  2.Add system parameter constants:FLASHVersion  3.Change constant names in EMV FlowType | SunmiPayHardwareService\_v3.3.62 |
| 3.2.31 | 2020/02/24 | 1.Card module add interfaces: ctx512ReadBlock(),ctx512WriteBlock(),ctx512UpdateBlock(),ctx512GetSignature(),ctx512MultiReadBlock(),mifareIncValueDx(),mifareDecValueDx(),mifareTransfer(),mifareRestore()  2.PinPad moudle add interface: setPinPadText()  3.Security moudle add interface: deleteKey(),saveKeyDukptAES()  4.SysParam add keyword:PCD\_PARAM\_A,PCD\_PARAM\_B,PCD\_PARAM\_C | SunmiPayHardwareService\_v3.3.64 |
| 3.2.32 | 2020/04/22 | 1.Card moudle add interfaces checkCardEx(),transmitApduExx(),transmitMultiApdus()  2.Basic module add interfaces setStatusBarDropDownMode(),setNavigationBarVisibility(),setHideNavigationBarItems(),sysPowerManage()  3.Add system param constants: DPASVersion、APEMVVersion  4.Add EMV transaction result code constants:AidlConstants.EMV.TransResult | SunmiPayHardwareService\_v3.3.66 |
| 3.2.33 | 2020/05/22 | 1. Modify illustration of interface setTermParamEx() | SunmiPayHardwareService\_v3.3.74 |
| 3.2.34 | 2020/06/02 | 1.PinPad moudle add interfaces setPinPadMode(), getPinPad() | SunmiPayHardwareService\_v3.3.76 |
| 3.2.35 | 2020/06/04 | 1. Modify illustration of interface setTermParamEx() | SunmiPayHardwareService\_v3.3.77 |
| 3.2.36 | 2020/06/16 | 1.Card moudle add interface checkCardEnc()  2.Add key “pan”,”name”,”expire” in CheckCardCallbackV2.findMagCard() | SunmiPayHardwareService\_v3.3.80 |
| 3.2.37 | 2020/06/17 | 1.Change doc veriosn to v3.2.37 | SunmiPayHardwareService\_v3.3.81 |
| 3.2.38 | 2020/07/03 | -- | -- |
| 3.2.39 | 2020/07/06 | 1.Modify illustration of interface EMV transactProcessEx()  2.Modify illustraction of interface checkCardEx() | SunmiPayHardwareService\_v3.3.86 |
| 3.2.40 | 2020/09/11 | 1. SecuityOptV2 add interfaces:   calcMacEx(),generateSM2Keypair(),injectSM2Key(),sm2Sign(),sm2VerifySign()、sm2EncryptData()、sm2DecryptData()、calcSecHash()   1. EMVOptV2 add interfaces:   setAccountDataSecParam(),getAccountSecData()   1. PinPadOptV2 add interface:   initPinPadEx()   1. Add system param constants：   ”PUREVersionFul”,”EFTPOSVersionFull”,”APEMVVersionFull”  5.change dukpt-3DES key index range to 0-9 or 1100-1199  6.Add dataIn length limit when dukpt keyIndex in 1100-1199 | SunmiPayHardwareService\_v3.3.95 |
| 3.2.41 | 2020/11/06 | 1.Add key index illustration in Appendix | SunmiPayHardwareService\_v3.3.95 |
| 3.3.42 | 2020/11/11 | 1.Add RSA transformation: RSA/ECB/OAEPWithSHA-1AndMGF1Padding, RSA/ECB/OAEPWithSHA-256AndMGF1Padding, RSA/ECB/OAEPWithSHA-512AndMGF1Padding | SunmiPayHardwareService\_v3.3.96 |
| 3.2.43 | 2020/12/16 | 1. EMVOptV2.setTermParamEx() add key  contactlessManualSelApp, importScriptData  2.AidlConstants.EMV.TLVOpCode add constant value  OP\_ADD\_SELF\_DEFINE\_TAG, OP\_DEL\_SELF\_DEFINE\_TAG  3.SystemParam add key “SecMode”,” PCD\_IFMVersion” | SunmiPayHardwareService\_v3.3.98 |
| 3.2.44 | 2020/12/29 | 1.update interface saveKeyDukptAES(), set param ksn length as 12 | SunmiPayHardwareService\_v3.3.99 |
| 3.2.46 | 2021/02/04 | 1. EMVOptV2 add interface  importTermRiskManagementStatus()  2. EMVListenerV2 add interface onTermRiskManagement() | SunmiPayHardwareService\_v3.3.102 |
| 3.2.47 | 2021/04/28 | 1.ReadCardOptV2 add smartCardIoControl() interface  2.SecurityOptV2 add verfiryMac()、RSA、RKI related interfaces  3.update SecurityOptV2 MKSK/Dukpt key index  4.remove keyword Sysparam.SDK\_VERSION | SunmiPayHardwareService\_v3.3.112 |
| 3.2.48 | 2021/07/27 | 1.PinPadOptV2 add interface setAntiExhaustiveProtectionMode(),getAntiExhaustiveProtectionMode() | SunmiPayHardwareService\_v3.3.126 |
| 3.2.49 | 2021/08/30 | 1.update Error code, add libbase error code | SunmiPayHardwareService\_v3.3.136 |
| 3.2.50 | 2021/09/16 | 1. add constant  AidlConstants.EMV.KernelType.RUPAY  2.Update comment of interface dukptIncreaseKSN () | SunmiPayHardwareService\_v3.3.137 |
| 3.2.51 | 2021/10/21 | 1. 1.update code values in EMVListenerV2.onTransResult() | SunmiPayHardwareService\_v3.3.137 |
| 3.2.52 | 2021/11/11 | 1.EMV add interface EMVListenerV2. onPreFirstGenAC()，EMVOptV2. importPreFirstGenACStatus()  2.Basic add interface BasicOptV2. sysSetWakeup()  3.update error code table  4.update interface BasicOptV2. buzzerOnDevice | SunmiPayHardwareService\_v3.3.149 |
| 3.2.53 | 2021/12/23 | 1.Param of interfaces getPinPadMode(),setPinPadMode() add key monitorClearKey | SunmiPayHardwareService\_v3.3.160 |
| 3.2.54 | 2022/01/25 | 1.EMVOptV2.setTermParamEx() add keyword “contactlessManualSelAppGeneralEx”  2.AidV2 add field extSelectSupFlg | SunmiPayHardwareService\_v3.3.166 |
| 3.2.55 | 2022/03/08 | 1.SysParam add keyword “KBBeepMode” | SunmiPayHardwareService\_v3.3.170 |
| 3.2.56 | 2022/04/08 | 1.Update illustration of interface calcMac() | SunmiPayHardwareService\_v3.3.170 |
| 3.2.57 | 2022/05/07 | 1.SecurityOptV2 add interface secKeyIoControl(), apacsMac()  2.AidlConstants.Secuirty add Mac type MAC\_ALG\_X9\_19\_DEA  3.AidlConstants.SysParam add key“SAM” | SunmiPayHardwareService\_v3.3.178 |
| 3.2.58 | 2022/07/07 | 1. EMV add API EMVListenerV2.onDataStorageProc（），EMVOptV2.importDataStorage ();  2.EMVOptV2.setTermParamEx()add key：dpasV2Support、dpasDeferredAuthSupport、dpasDataStorageSupport、dpasExtendedLoggingSupport、dpasTearingRecoverySupport | SunmiPayHardwareService\_v3.3.191 |
| 3.2.59 | 2022/08/17 | 1.SecurityOptV2 add interface saveKeyEx() | SunmiPayHardwareService\_v3.3.191 |
| 3.2.60 | 2022/10/20 | 1.mark SecurityOptV2.generateRSAKeys(),getRSAPublicKey(),getRSAPrivateKey(),dataEncryptRSA(),dataDecryptRSA(),removeRSAKey(),signingRSA(),verifySignatureRSA(),saveCiphertextKeyRSA(),saveRSAKey() as deprecated  2.add error code -2549: Type–A not support ISO14443–4, activation process aborted  3.PinPadOptV2.setPinPadMode()/getPinPadMode() support key word cancelToClear.  4.delete chapter【7.4 key index illustration】  5.modify the comment of interface rsaEncryptOrDecryptData()  6.Add chapter 【7.4 key inject/access flow chart】 | SunmiPayHardwareService\_v3.3.215 |
| 3.2.61 | 2022/11/09 | Update EMV result code description | SunmiPayHardwareService\_v3.3.215 |
| 3.2.62 | 2022/11/10 | 1.BasicOptV2 add interface installSharedLib(),deleteSharedLib()  2.Interface injectCiphertextKeyEx() add param dataMode, iv  3.Interface injectKeyDukptEx() add param dataMode, iv  4.Interface saveKeyEx () add param dataMode, iv, isEncrypt  5.Add error code -20000: function not supported, -2530: the card refused command | SunmiPayHardwareService\_v3.3.225 |
| 3.2.63 | 2022/11/25 | 1.SecurityOptV2 add interface generateSymKey(),injectSymKey(),generateSymKeyEx(),injectSymKeyEx()  2.AdilConstants add inject symmetric key related constatns. | SunmiPayHardwareService\_v3.3.228 |
| 3.2.64 | 2022/12/20 | 1.Add device certificate manager module | SunmiPayHardwareService\_v3.3.232 |
| 3.2.65 | 2022/12/22 | 1.Delete [in]/[out] param in empty input param method | SunmiPayHardwareService\_v3.3.233 |
| 3.2.66 | 2023/01/05 | 1.SecurityOptV2 add interface generateRSAKeypairEx(),injectRSAKeyEx()  2.AidlConstants.Security add constants KEY\_TYPE\_RSA\_KPK, KEY\_TYPE\_RSA\_KEK  3. SecurityOptV2 support close key partition  4.Update the illustration of param keyIndex in SecurityOptV2 interfaces  5.Update [7.4 Key system and key index range] table | SunmiPayHardwareService\_v3.3.234 |
| 3.2.66T02 | 2023/01/12 | 1. transmitMultiApdus():change the max send APDU count as 7 at one time. 2. checkCardEnc() parameter Bundle add key: encKeyAlgType,indicates the algorithm type of the key; 3. calcSecHash():the parameter length of dataIn description is changed to “less than or equal to”; |  |
| 3.2.67 | 2023/02/13 | 1. PinPadOptV2() add API setVisualImpairmentModeParam(),getVisualImpairmentModeParam(); 2. initPinPadEx() add parameter discription; 3. transmitApduExx() add parameter description for bit 6; 4. Error Code add -2904，-2909； 5. rsaEncryptOrDecryptData() update parameter description; 6. update param dataIn length for interfaces sm2Sign()，sm2VerifySign(),sm2EncryptData(),m2DecryptData(). 7. saveKeyDukpt() add comment; 8. setPinPadMode() parameter add key:visualImpairment; |  |
| 3.2.68 | 2023/03/10 | 1.update description of interface calcSecHash()  2. add error code -80001，-2911  3.update description of interface savePlaintextKey() |  |
| 3.2.69 | 2023/04/21 | 1.update description of interface injectCiphertextKey()  2.add note for RSA signature NONEWithRSA  3. |  |
| 3.2.70 | 2023/05/10 | 1. update param ttsLanguage value range for interface setVisualImpairmentModeParam(),getVisualImpairmentModeParam() 2. update kcv param description 3. saveKeyEx() add key param kcvMode 4. update description of interface saveTR31Key() 5. Update error code, add code -16000~-16008 | SunmiPayHardwareService\_v3.3.304 |
| 3.2.71 | 2023/05/17 | 1.Update description of interface  dataEncryptRSA()，dataDecryptRSA()，signingRSA()，verifySignatureRSA()  2.BasicOptV2, ReadCardOptV2, SecurityOptV2, PinPadOptV2, EMVOptV2 Add Brazil-CKD and TOSS special interfaces  3. Update the backup column in [Key system and key index range] table | SunmiPayHardwareService\_v3.3.307 |
| 3.2.72 | 2023/05/31 | 1.update emv transaction flow chart  2.Update description of interface injectDeviceCertPrivateKey()  3.interface injectCiphertextKeyEx()、injectKeyDukptEx() add params kcvMode, kcvMacType, kcvInData | SunmiPayHardwareService\_v3.3.308 |
| 3.2.73 | 2023/06/10 | 1. SecurityOptV2 add interface()  dataEncryptEx(),dataDecryptEx()  2.Update [7.4 Key system and key index range] table | SunmiPayHardwareService\_v5.0.15 |
| 3.2.74 | 2023/07/05 | 1.Update description of interface setPreferredNetworkMode(),readRSAKey(),getKeyLength(),writeKeyVariable(),generateRSAKeypairEx(),injectRSAKeyEx()  2.Update description of interface dataEncryptEx()、dataDecryptEx()  2.Update [7.4 Key system and key index range] table  3.Update error code table | SunmiPayHardwareService\_v5.0.16 |
| 3.2.75 | 2023/07/13 | 1.PinPadOptV2.setPinPadMode(), getPinPadMode interfaces add key param longTimeoutTime | SunmiPayHardwareService\_v3.3.311 |
| 3.2.76 | 2023/08/07 | 1.Param bundle of interface PinPadoptV. setVisualImpairmentModeParam(),getVisualImpairmentModeParam() add key rnibSelectMode, rnibHoldTime  2.update description of interface checkCard(),checkCardEx(),checkCardEnc(),checkCardForToss() | SunmiPayHardwareService\_v3.3.313 |
| 3.2.77 | 2023/08/22 | 1.Update description of interface setVisualImpairmentModeParam() | SunmiPayHardwareService\_v3.3.313 |
| 3.2.78 | 2023/09/05 | 1. Update description of interface BasicOptV2.setSysParam()  2.Update description of modules and interfaces, add note for interafces which unsupported on TOSS | SunmiPayHardwareService\_v3.3.313 |
| 3.2.79 | 2023/09/14 | 1. PinPadOptV2 add interface importPinPadData()  2.Update EMVOptV2. importPinInputStatus(), add new value 4 for param inputResult | SunmiPayHardwareService\_v3.3.313 |
| 3.2.80 | 2023/10/11 | 1. PinPadOptV2 add interface importPinPadDataEx() | SunmiPayHardwareService\_v3.3.318 |
| 3.2.81T | 2023/10/30 | 1. Update the comments of interface checkCard(), checkCardEx(), add magstripe track data risk comments | SunmiPayHardwareService\_v3.3.318 |
| 3.2.81 |  | Checkcard for toss,findMagCard add return param track2Raw | SunmiPayHardwareService\_v3.3.320 |
| 3.2.82 | 2023/11/29 | 1.Update 7.1.30 Led constant definition | SunmiPayHardwareService\_v5.0.22 |
| 3.2.83 | 2023/12/13 | 1.Update chaper[2.2.1 Quick integration], add device type P2, P2\_smartPad etc，add illustration for integrate paylib.aar from maven central  2. Update description of interfaces getModuleAccessibility(),setModuleAccessibility(),getPedMode(),setPedMode(),setScheduleReboot(),clearScheduleReboot(),readRSAKey(), remove [Brazil-CKD special] flag  3.Update chapter [7.1.11 RSA padding mode], add constant PADDING\_OAEP\_SHA1  4.Add chapter [7.1.27 EMV Online result]  5.Update chapter [7.1.31 Led constant definition], add supported device type for corner LED and indicator LED  6.Update commnets of interfaces checkCard(),checkCardEx(),checkCardEnc(),checkCardForToss(), add power-off illustration  7.Update description of interface cardOff()  8.Add chapter [7.3 Read card flow chart]  9.Update [3.3 Basic operation module]， add interface getRtcBatVol()  10.Update[7.1.30 System Parameter constant definition] add key IfmLibVersion, MsrVersion | SunmiPayHardwareService\_v5.0.23 |
| 3.2.84 | 2023/12/26 | 1.Update[7.1.30 System Parameter constant definition] add key posapiVersion, RTCBATVOLDET | SunmiPayHardwareService\_v5.0.24 |
| 3.2.85 | 2024/01/02 | 1. Update commnets for interfaces of save key and save device cert, add illustration of key duration and deletability  3.Update description of interface calcSecHash()  4.Update interface checkCardEnc()，support RSA encrypt track data, param bundle add key panAppendContent, panAppendMode  5.Update interface CheckCardCallbackV2.findMagCard(), add field appendedPanEnc, appendedPanEncBytes in returned param bundle  6.Updatea description of param sendBuff, recvBuff in interface transmitApdu(),transmitApduEx(), transmitApduExx()  7. ReadcardOptV2 add interface smartCardExChangePASS(), smartCardExChangePASSNoLength()  8.Update interface initPinPad(), initPinPadEx(), startInputPin(), change param pinPadType value rang from 0-1 to 0-5  9. add interface PinPadListenerV2.onHover()  10. Update [7.1.33 PinBlock format constant definition] | SunmiPayHardwareService\_v3.3.324 |
| 3.2.86 | 2024/03/07 | 1.add constants AidlConstants.EMV. TLVOpCode.OP\_CPACE, AidlConstants.EMV.KernelType.CPACE, AidlConstants.SysParam.CPACE\_VERSION、AidlConstants.SysParam.CPACE\_RELEASE\_DATE  2. add keys supportPOI、CertifiedEP、AutoRun、KernelsForCertEP for param bundle in interface setTermParamEx() | SunmiPayHardwareService\_v3.3.326 |
| 3.2.87 | 2024/03/13 | 1. SecurityOptV2 add interface queryKeyMappingRecordList() | SunmiPayHardwareService\_v3.3.327 |
| 3.2.88 | 2024/03/20 | 1. add sred description for interfaces checkCard(),checkCardEx(),getTlv(),getTlvList(),onConfirmCardNo(),setAccountDataSecParam()  2.add chapter [7.9 SRED description]  3.SecurityOptV2 add interface queryKeyMappingRecordListWL(), injectTR31Key()  4.BasicOptV2 add interface readPuk()  5.Update description of interface BasicOptV2.setPedMode(), getPedMode() | SunmiPayHardwareService\_v3.3.328  SunmiPayHardwareService\_v5.0.28 |
| 3.2.89 | 2024/04/05 | 1. SecuirtyOptV2 add interfaces readSM2Key()、calcSM3HashWithID()、sm2SingleSign() | SunmiPayHardwareService\_v3.3.328 |
| 3.2.90 | 2024/04/23 | 1. Update [7.1.14 Inject symmetric key mode definition], add mode INJECT\_OAEP\_MODE, INJECT\_PKCS1\_MODE  2. Update description of interface injectSymKeyEx() | SunmiPayHardwareService\_v3.3.329 |
| 3.2.91 | 2024/05/17 | 1.[3.5 PinPad operarion module] add description for device with physical keyboard  2.Update[7.5 Key inject/access flow chart], add svg flow chart file  3.Update[7.1.30 System parameter constant definition], add emv library release date keyword | SunmiPayHardwareService\_v3.3.331  SunmiPayHardwareService\_v5.0.32 |
| 3.2.92 | 2024/06/07 | 1. Update description of param **cardType** in interface getCardExistStatus() | SunmiPayHardwareService\_v3.3.331  SunmiPayHardwareService\_v5.0.32 |
| 3.2.93 | 2024/06/13 | 1.Add chapter [7.10 EMV default data]  2.Update description of interfaces addAid(),deleteAid(),addCapk(),deleteCapk()  3.Update chapter[7.1.4 MAC algorithm constant definition], add mac type MAC\_ALG\_HMAC\_SHA1, MAC\_ALG\_HMAC\_SHA256, MAC\_ALG\_CMAC | SunmiPayHardwareService\_v3.3.331  SunmiPayHardwareService\_v5.0.32 |
| 3.2.94 | 2024/06/26 | 1.Update the description of param in interface checkCardEnc()  2.Update the description of interface setAccountDataSecParam(), add key encKeyAlgType,panAppendContent, panAppendMode in param bundle  3.Update interface getAccountSecData()，add key appendedPanEncBytes , appendedPanEnc in output param bundle | SunmiPayHardwareService\_v5.0.33 |
| 3.2.95 | 2024/07/31 | 1.AidlConstants.SysParam add keywords “PCIPTSVersion”, “RNIBVersion”, “sred”  2. Update interface cardOff()，support composite card types when deactive card  3.Update description for interfaces startInputPin(), getPinBlock(), remove Brazil-CKD limitation  4.Update description of interface setAccountDataSecParam()  5.Update interface setTermParamEx(), add key “quickChip”, “noSignatureOrPINThreshold”,” dpasContactlessSpeedupSupport”,” jcbContactlessSpeedupSupport”,” AEContactlessSpeedupSupport”,” AEOnlineProcessSupport”,” SupportAE4.1”,  6.Update chapter [7.9 SRED description]  7.Update interface setSysParam(), add serd description  8.Update description of interface transmitApdu() | SunmiPayHardwareService\_v5.0.34 |
| 3.2.96 | 2024/08/17 | 1. Add chapter [3.10 HCE manager module]  2.Update error code table, add HCE relevant code -2401~-2408 | SunmiPayHardwareService\_v5.0.35 |
| 3.2.97 | 2024/08/29 | 1.Updata interfaces checkCard(),checkCardEx(), checkCardEnc(),checkCardForToss(), change param timeout max value to 600  2.Update PinPadListenerV2.onHover(), update describle of param data | SunmiPayHardwareService\_v3.3.335 |

## 2. Abstract

### 2.1 Introduction

SunmiPaySDK is a set of interfaces which is based on firmware encapsulation and is close to Java developer to call hardware. Through this SDK, developer can quickly call the corresponding firmware interfaces of Sunmi financial instruments and implement their own business logic. SDK mainly includes: terminal information basic module, card operation module, PinPad module, EMV module, security module.

This document is the SunmiPaySDK V2 interface documentation. Compared to the V1 interface, the V2 interface is easier to integration and understanding.

### 2.2 SDK Integration

#### 2.2.1 Quick integration

This document is an integrated development guide for the SUNMI’s POS. To guide the use method of SDK, It requires the reader to have already familiar with Android develop IDE (Android Studio), and have certain Android programming experiences, familiar with financial related specifications, processes and concepts (eg: key, pinblock, pan,emv, MAC, etc.). At present, SDK only supported run on Sunmi finance devices(Currently, they’re P1N, P1\_4G, P2, P2Pro, P2Lite, P2mini, P2\_smartPad, P2\_SE, P2\_Xpro, TOSS, P2liteSE, P3\_MIX). Please read this document carefully before using SDK. Check the following prerequisites firstly:

1. Device type is one of P1N, P1\_4G, P2, P2Pro, P2Lite, P2mini, P2\_smartPad, P2\_SE, P2\_Xpro, TOSS, P2liteSE, P3\_MIX. (Setting-About device-Model number)
2. Go to Setting—>APP—>top right cornor—>show system processes, confirm the version of installed SunmiPayHardwareService is v3.3.xxx or later
3. Android studio quick integrate SDK library:
4. Integrate SDK library by local: put the PayLib-release-xxx.aar file to libs folder, then add following codes to app/build.gradle file

|  |
| --- |
| repositories {  flatDir {  dirs 'libs'  }  }  dependencies {  ......  compile(name: 'PayLib-release-xxx', ext: 'aar')  } |

1. Integrate SDK library by maven: add following codes to project root build.gradle file:

|  |
| --- |
| buildscript {  repositories {  mavenCentral()  }  }  allprojects {  repositories {  mavenCentral()  }  } |

then add following codes to app/build.gradle file:

|  |
| --- |
| dependencies {  implementation 'com.sunmi:PayLib-release:2.0.07'  } |

then sync and rebuild the Project.

#### 2.2.2 Android version and IDE version supported by the SDK

SDK only support API-19(Android 4.4) or the latest version.

SDK only support Android studio、Intellij.

## 3. API

### 3.1 SunmiPayKernel SDK Operation Object

#### 3.1.1 Get an instance of SunmiPayKernel

|  |  |
| --- | --- |
| Prototype | SunmiPayKernel getInstance() |
| Feature | Get kernel instance object of SDK |
| Parameter | None |
| Return | SunmiPayKernel |
| Comment | None |

#### 3.1.2 Connect to PaySDK

|  |  |  |
| --- | --- | --- |
| Prototype | initPaySDK (Context context,ConnectCallback connectCallback) | |
| Feature | Initialize Pay SDK. | |
| Parameter | context[in] | Context |
| connectCallback[in] | Callback of connection status. Refer to [ConnectCallback](#ConnectCallback) |
| Return | None | |
| Comment | It is recommended to call this method in your app’s Application class, if failure to call this method, any invoking of SDK API method may throw NullPointerException. | |

#### 3.1.3 Disconnect to PaySDK

|  |  |
| --- | --- |
| Prototype | destroyPaySDK () |
| Feature | Destroy the Pay SDK. |
| Parameter | None |
| Return | None |
| Comment | This method will disconnect your app to Pay SDK. After calling this method, any function module in SunmiPayKernel will be set to **null**, and any invoke of SDK API method will throw NullPointerException. To prevent a memory leak, call this methos in Activity’s onDestroy() is best practice. |

#### 3.1.4 ConnectCallback

##### 3.1.4.1 Connect to PaySDK callback

|  |  |
| --- | --- |
| Prototype | onConnectPaySDK () |
| Feature | Callback method, indicate connect to PaySDK success |
| Parameter | None |
| Return | None |
| Comment | Any SDK API method calls must be made after receiving this callback. After initializing the SDK, developers should pay attention to this method. When this method called, it indicates the SDK was initialized successfully, and the SDK related API can be invoked rightly. |

##### 3.1.4.2 Disconnect to PaySDK callback

|  |  |
| --- | --- |
| Prototype | onDisconnectPaySDK () |
| Feature | Disconnect the PaySDK |
| Parameter | None |
| Return | None |
| Comment | When this method called, it indicates your app has disconnected to Pay SDK, any invoke of SDK API method may produce Exception. To use SDK API, please call method InitPaySDK() firstly. |

### 3.2 Public member variable

After connected to PaySDK, in SunmiPayKernel, there are some function modules can be used to interactive with PaySDK, as following table showing:

|  |  |  |
| --- | --- | --- |
| Variable name | Description | Comment |
| mBasicOptV2 | Basic operation module | Include APIs of get basic info，control LED，control buzzer,etc |
| mReadCardOptV2 | Read card module | Include card APIs |
| mPinPadOptV2 | PinPad operation module | Includes PinPad API |
| mSecurityOptV2 | Security module | Include APIs of save key，mac calculation，encrypt data,etc |
| mEMVOptV2 | EMV module | Include APIs of EMV |
| mTaxOptV2 | Tax module | Include APIs of Tax |
| mDevCertManagerV2 | Device certificate manage module | Includes APIs of device certificate manage |

### 3.3 Basic operation module

#### 3.3.1 Get system parameter

|  |  |  |
| --- | --- | --- |
| Prototype | String getSysParam(String key) | |
| Feature | Read the properties of the system resource key through the user parameter keyword. | |
| Parameter | key[in] | Refer to [AidlConstantsV2. SysParam](#_SysParam__SysParam常量定义) |
| Return | Attribute values queried | |
| Comment | 1.If the required attribute value does not exist, then return "NULL". | |

#### 3.3.2 Set system parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int setSysParam(String key, String value) | |
| Feature | Set system parameter | |
| Parameter | key[in] | Refer to [AidlConstantsV2. SysParam](#_SysParam__SysParam常量定义) |
| value[in] | value |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. key, value refer to the constant values at [AidlConstantsV2. SysParam](#_SysParam__SysParam常量定义)   |  |  |  | | --- | --- | --- | | key | value | Function | | SysParam.RESERVED | JSON string | Config non sdk default value | | SysParam.TERM\_STATUS | One of the following values:  SysParam.CLEAR\_TAMPER\_LOG  SysParam.CLEAR\_TAMPER | Clear tamper log  Clear tamper | | SysParam.PINPAD\_MODE | One of the following values:  PinPadMode.MODE\_NORMAL  PinPadMode.MODE\_MEITUAN  PinPadMode.MODE\_SILENT  PinPadMode.MODE\_LEDOFF | Set PinPad mode | | SysParam.PCD\_PARAM\_A | Hex string, eg: 95018F0F | Set A card contactless param | | SysParam.PCD\_PARAM\_B | Hex string, eg: 8D018F03 | Set B card contactless param | | SysParam.PCD\_PARAM\_C | Hex string, eg: 8D018F03 | Set Felica card contactless param | | SysParam.SEC\_MODE | “0”or “1” | Set open or close key same checking flag. If open, save the same keyValue MKSK or DUKPT key will fail | | SysParam.KB\_BEEP\_MODE | One of the following values:  KBBeepMode.MODE\_ON  KBBeepMode.MODE\_OFF | Set beep mode of physical keyboard key | | SysParam.EMV\_MASK | Hex string | Set EMV mask | | SysParam.SRED | “0” | Disable sred, for enable sred, please refer to interface [setAccountDataSecParam()](#_3.7.1.35_Set_account) |   2.Note for **key**, **value**:  (1) for all key, value, it’s not recommended to change value, use sdk default is recommend  (2) **TERM\_STATUS** is used for produce device, client APP should nott use this key  (3) PINPAD\_MODE is recommended use, please use PinPadOptV2.setPinPadMode() to set PinPad mode  (4) **PCD\_PARAM\_A**,**PCD\_PARAM\_B**,**PCD\_PARAM\_C** only supported on P2 or P2\_PRO, for other device types, it is unsupprted  (5) **KB\_BEEP\_MOD** only supported on P2\_smartPad device  (6) **EMV\_MASK** only used for EMV authentication | |

#### 3.3.3 Control Buzzer

|  |  |  |
| --- | --- | --- |
| Prototype | void buzzerOnDevice(int count, int freq, int duration, int interval) | |
| Feature | Control the Buzzer on the device. | |
| Parameter | count[in] | Beep times (0~100) |
| freq[in] | The frequency of tone (unit: Hz) |
| duration[in] | The duration time (unit: ms) |
| interval[in] | The time interval between two beeps (unit: ms) 0~10000 |
| Return | None | |
| Comment | On device P2lite/P2Pro/P2/P2Mini/P2\_xpro, there are no really buzzer exist, beep sound made by play audio file and generate by loudspeaker.  1. param **freq** rule**:**  (1) 0-750，play 750Hz audio file  (2) 751-1000，play 1000Hz audio file  (3) 1001-1500，play 1500Hz audio file  (4) 1501-1750，play 1750Hz audio file  (5) 1751-2500，play 2500Hz audio file  (6) 2501-2750，play 2750Hz audio file  (7) 2751-3250，play 3250Hz audio file  (8) 3250以上，play 4000Hz audio file  2. param **duration** default value is 200ms, other values are ignored | |

#### 3.3.4 Control Led

|  |  |  |
| --- | --- | --- |
| Prototype | int ledStatusOnDevice (int ledIndex, int ledStatus) | |
| Feature | Control Led status on device | |
| Parameter | ledIndex[in] | Refer to LedLight definition: [AidlConstantsV2.LedLight](#_LedLight__LedLight常量定义) |
| ledStatus[in] | LED status, 1-LED off; 0-LED on. |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.On P3 device, the physical LED lights represented by indexes 6-8 are a group of lights, and the index represents a certain color of this group of lights, and the colors can be overlaid. Eg: turn-on the lights at index 6 and 7, the color of the light is displayed as a combination of red and green, which is yellow**  **2.This interface is not supported on TOSS device** | |

#### 3.3.5 Set screen mode

|  |  |  |
| --- | --- | --- |
| Prototype | int setScreenMode(int mode) | |
| Feature | Set the screen exclusive, disable the bottom navigation bar and the SystemUI drop-down box, and disable the volume key. | |
| Parameter | mode[in] | Set screen exclusive mode.1：Set screen exclusive，-1: Set screen exclusive exit. |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.Setting up screen exclusivity usually needs an appropriate time to cancel. It is recommended that the user use this interface with the power management lock to keep the screen always bright and unlocked. Otherwise, the screen exclusivity can be maintained after the information screen appears. Only by unplugging the battery and restarting the device can the screen exclusivity be removed.  **2. This interface is not supported on TOSS device** | |

#### 3.3.6 Get random data

|  |  |  |
| --- | --- | --- |
| Prototype | int sysGetRandom(byte[] randData, int len) | |
| Feature | Get a specific length random data from SDK | |
| Parameter | randData[out] | Buffer, store the random data |
| len[in] | The data length, range 0~256 |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

#### 3.3.7 Control Led

|  |  |  |
| --- | --- | --- |
| Prototype | int ledStatusOnDeviceEx(int redStatus, int greenStatus, int yellowStatus, int blueStatus) | |
| Feature | Control all Leds status on device at one time | |
| Parameter | redStatus[in] | Red LED status, 0-on, 1-off |
| greenStatus[in] | Green LED status, 0-on, 1-off |
| yellowStatus[in] | yellow LED status, 0-on, 1-off |
| blueStatus[in] | Blue LED status, 0-on, 1-off 6 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.3.8 Set status bar dropdown mode

|  |  |  |
| --- | --- | --- |
| Prototype | int setStatusBarDropDownMode(int mode); | |
| Feature | Set status bar dropdown mode | |
| Parameter | mode[in] | 0-enable dropdown, 1-disable dropdown |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.3.9 Set navigation bar visibility

|  |  |  |
| --- | --- | --- |
| Prototype | int setNavigationBarVisibility(int visibility) | |
| Feature | Set navigation bar visibility | |
| Parameter | visibility[in] | Navigation bar visibility, 0-Gone, 1-Visible |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.3.10 Set hide navigation bar items

|  |  |  |
| --- | --- | --- |
| Prototype | int setHideNavigationBarItems(int flag) | |
| Feature | Hide navigation bar items | |
| Parameter | flag[out] | Composite value, the items to be hide:  STATUS\_BAR\_DISABLE\_HOME=0x00200000;//hide home key  STATUS\_BAR\_DISABLE\_BACK = 0x00400000;//hide back key  STATUS\_BAR\_DISABLE\_RECENT = 0x01000000;//hide recent key  Set **flag=STATUS\_BAR\_DISABLE\_HOME | STATUS\_BAR\_DISABLE\_BACK** to hide home and back key |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.3.11 Manage device power

|  |  |  |
| --- | --- | --- |
| Prototype | int sysPowerManage(int mode) | |
| Feature | Manage deivce power | |
| Parameter | mode[in] | Device power mode, 1-dormant(not support), 2-shutdown, 3-reboot |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

#### 3.3.12 Allow dynamic permission (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int allowDynamicPermission(String packageName) | |
| Feature | Allow dynamic permission for a specific package | |
| Parameter | packageName [in] | The target package name |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.13 Set global wifi proxy (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setGlobalProxy(String proxy) | |
| Feature | Set global wifi proxy | |
| Parameter | proxy [in] | The wifi proxy, format ip:port or url:port |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.14 Install CA certificate (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int installApplicationCertificate(String name, String contents) | |
| Feature | Install a CA certificate | |
| Parameter | name[in] | The certificate name |
| contents[in] | The certificate contents |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.15 Uninstall CA certificate (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int uninstallApplicationCertificate(String name) | |
| Feature | Uninstall a certificate which installed by **installApplicationCertificate()** | |
| Parameter | name[in] | The certificate name |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.16 Get CPU usage (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | String getCpuUsage() | |
| Feature | Get CPU usage | |
| Parameter | [in] | None |
| Return | The CPU usage | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.17 Get CPU temperature (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | String getCpuTemperature() | |
| Feature | Get CPU temperature | |
| Parameter | [in] | None |
| Return | The CPU temperature | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.18 Set schedule reboot time

|  |  |  |
| --- | --- | --- |
| Prototype | int setScheduleReboot(int hour, int minute, int second, int millisecond) | |
| Feature | Set a schedule reboot time for the device | |
| Parameter | hour[in] | The hour, range 0~23 |
| minute[in] | The minute, range 0~59 |
| second[in] | The second, range 0~59 |
| millisecond[in] | The millisecond, range 0~999 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.The functionality of interface is independent of system, that’s , if call this interface and set scheduled reboot with system interface at the same time, both the two setting will take effect.** | |

#### 3.3.19 Clear schedule reboot time

|  |  |  |
| --- | --- | --- |
| Prototype | int clearScheduleReboot() | |
| Feature | Clear the schedule reboot time which set by **setScheduleReboot()** | |
| Parameter | [in] | None |
| Return | 0: Success  Other value: Fail | |
| Comment |  | |

#### 3.3.20 Customize function key (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int customizeFunctionKey(Bundle bundle) | |
| Feature | Customize the function of function key | |
| Parameter | bundle[in] | Input param, contain following keys:  key: String, key button name, “volume\_1”,”volume\_2”, ”function\_1”,”function\_2”  type: String, fuction type, “function”, ”native”,”app\_launch”  value: String, the corresponding value of type.  when type is “function”, value can be“volume\_up”or “volume\_down”,  when type is “native”, value should be“native”  when type is “app\_launch“，value should be the package name of the app that you want to launch. |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.21 Set low memory killer package (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setLMKPackage(String packageName) | |
| Feature | Set a package to system low memory killer white list | |
| Parameter | packageName[in] | App package name |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.22 Remove low memory killer package (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int removeLMKPackage(String packageName) | |
| Feature | Remove package from system low memory killer white list | |
| Parameter | packageName[in] | App package name |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.23 Set device wakeup sources

|  |  |  |
| --- | --- | --- |
| Prototype | int sysSetWakeup(int channel, int mode, Bundle attr) | |
| Feature | Set device wakeup sources | |
| Parameter | channel [in] | Wakeup sources, 1-IC card wakeup, 2-Magnetic stripe card wakeup, 3- press key wakeup |
| mode[in] | 0-disable, 1-enable |
| attr[in] | Reserved |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.3.24 Set preferred network type (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setPreferredNetworkMode(int mode, int slotIndex) | |
| Feature | Set preferred network type | |
| Parameter | mode[in] | Network type, contains following values:  0-GSM/WCDMA (WCDMA preferred)  1-GSM only  \*2-WCDMA only  3-GSM/WCDMA (auto mode, according to PRL) AVAILABLE Application Settings menu  4-CDMA and EvDo (auto mode, according to PRL) AVAILABLE Application Settings menu  5-CDMA only  6-EvDo only  7-GSM/WCDMA, CDMA, and EvDo (auto mode, according to PRL) AVAILABLE Application Settings menu  8-LTE, CDMA and EvDo  9-LTE, GSM/WCDMA  10-LTE, CDMA, EvDo, GSM/WCDMA  11-LTE Only mode.  12-LTE/WCDMA  13-TD-SCDMA only  14-TD-SCDMA and WCDMA  15-TD-SCDMA and LTE  16-TD-SCDMA and GSM  17-TD-SCDMA,GSM and LTE  18-TD-SCDMA, GSM/WCDMA  19-TD-SCDMA, WCDMA and LTE  20-TD-SCDMA, GSM/WCDMA and LTE  21-TD-SCDMA,EvDo,CDMA,GSM/WCDMA  22-TD-SCDMA/LTE/GSM/WCDMA, CDMA, and EvDo  30-LTE/GSM  31-LTE TDD Only mode.  32-CDMA,GSM(2G Global)  33-CDMA,EVDO,GSM  34-LTE,CDMA,EVDO,GSM(4G Global, 4M) |
| slotIndex[in] | The SIM card slot index, range: 0~1 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.**  **2.Param mode’s value should be one type of value returned by interface getSupportedNetworkType()** | |

#### 3.3.25 Get supported network type (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | String getSupportedNetworkType(int slotIndex) | |
| Feature | Get supported network type | |
| Parameter | slotIndex[in] | The SIM card slot index, range: 0~1 |
| Return | Not empty: Current slot sim card support network type  Empty: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.26 Enable or disable airplane mode(Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setAirplaneMode(boolean enable) | |
| Feature | Enable or disable airplane mode | |
| Parameter | enable[in] | true-enable airplane mode, false-disable airplane mode |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.27 Enable or disable data roaming of SIM card (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setDataRoamingEnable(int slotIndex, boolean enable) | |
| Feature | Enable or disable data roaming | |
| Parameter | slotIndex[in] | The SIM card slot index, range: 0~1 |
| enable[in] | true-enable data roaming, false-disable data roaming |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.28 Enable or disable phone call (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int enablePhoneCall(boolean enable) | |
| Feature | Enable or disable phone call | |
| Parameter | enable[in] | ture-enable phone call, false-disable phone call |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.29 Get card usage count

|  |  |  |
| --- | --- | --- |
| Prototype | int getCardUsageCount(int cardType, boolean isSuccess) | |
| Feature | Get mag swipe card, IC/NFC check card success/failure count | |
| Parameter | cardType[in] | Card type, can be MAGNETIC/IC/NFC |
| isSuccess[in] | ture-get success count, false-get failure count |
| Return | >=0: Card usage count  Other value: Error code | |
| Comment |  | |

#### 3.3.30 Get module accessibility

|  |  |  |
| --- | --- | --- |
| Prototype | int getModuleAccessibility(int module) | |
| Feature | Get module accessibility | |
| Parameter | module[in] | Module, 1-MAG，2-ICC，3-PICC，4-PinPad |
| Return | 0-disabled, 1-Enabled  <0: Error code | |
| Comment |  | |

#### 3.3.31 Set module accessibility

|  |  |  |
| --- | --- | --- |
| Prototype | int setModuleAccessibility(int module, int ability) | |
| Feature | Set module accessibility | |
| Parameter | module[in] | Module, 1-MAG，2-ICC，3-PICC，4-PinPad |
| ability[in] | 0-disable, 1-enable |
| Return | 0: Success  Other value: Error code | |
| Comment |  | |

#### 3.3.32 Get PED mode

|  |  |  |
| --- | --- | --- |
| Prototype | int getPedMode() | |
| Feature | Get PED mode | |
| Parameter | [in] | None |
| Return | >=0: PED mode, 1-Sharing mode, 2-Isolation mode, 3-Mixed mode  Other value: Error code | |
| Comment | 1. For brazil CDK device, this interface return 1 or 2, for other device types it only return 2 | |

#### 3.3.33 Set PED mode

|  |  |  |
| --- | --- | --- |
| Prototype | int setPedMode(int mode) | |
| Feature | Set PED mode | |
| Parameter | mode[in] | PED mode: 1-Sharing mode, 2-Isolation mode, 3-Mixed mode |
| Return | 0: Success  Other value: Error code | |
| Comment | **1. For brazil CKD device type, this interface support set PED mode as 1 or 2, and switch PED mode will clear all stored keys.**  2.For non-brazil-CKD device type, it’s not support set PED mode, this interface always return -2147483648(Unknown Error) | |

#### 3.3.34 Get PED keys info (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int getPedKeysInfo(Bundle info) | |
| Feature | Get PED keys info | |
| Parameter | info[out] | PED keys info |
| Return | 0: Success  Other value: Error code | |
| Comment | 1.Currently, SDK not support get PED keys info, this interface always return -2147483648(Unknown Error)  **2.This interface is Brazil-CKD special, not supported on other device types.** | |

#### 3.3.35 Install shared lib

|  |  |  |
| --- | --- | --- |
| Prototype | int installSharedLib(String path) | |
| Feature | Install a shared lib | |
| Parameter | path[in] | The shared lib absolutely path, eg: “/sdcard/emlib/libAE.so” |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

#### 3.3.36 Delete shared lib

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteSharedLib(String name) | |
| Feature | Delete a shared lib | |
| Parameter | name[in] | The shared lib’s name, eg:“libAE.so” |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

#### 3.3.37 Install Liteso file (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoInstaller(int index, String filePath) | |
| Feature | Install a Liteso ile | |
| Parameter | index[in] | Liteso index, range: 0~11 |
| filePath[in] | Liteso file path, length<=128 |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.Liteso file size<=200KB  2.This method override install previous installed Liteso(if exist) at **index**.  3.This method need access right for Liteso file speicified by **filePath**  **4.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.38 Run Liteso file (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoRun(int index) | |
| Feature | Run a installed Liteso file which specified by param index | |
| Parameter | index [in] | Liteso index, range: 0~11 |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.Make sure Liteso installed at **index** before run it  2.Run Liteso will clear all the memory data last time run and restart SE  3.In the chip lifetime, run Liteso no more than 10000 times  **4.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.39 Get Liteso info (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoInfo(int index, Bundle info) | |
| Feature | Get Liteso information | |
| Parameter | index[in] | Liteso index, range: 0~11 |
| info[out] | Liteso information, contains following key:  name: String, Liteso name  desc: String, Liteso app description  vender: String, Liteso vernder  version: String, Liteso version |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.40 Get currently running Liteso info (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoRunInfo(Bundle info) | |
| Feature | Get the information of currently running Liteso file | |
| Parameter | info[in] | Liteso information, contains following key:  name: String, Liteso name  desc: String, Liteso app description  vender: String, Liteso vernder  version: String, Liteso version |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.41 Exchange data with Liteso (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoCustomCmd(int cmd, byte[] dataIn, byte[] dataOut) | |
| Feature | Exchange data with Liteso | |
| Parameter | cmd [in] | The command code |
| dataIn[in] | Additional data for **cmd**，length<=1500B |
| dataOut[out] | Buffer, store output data, max output data length<=1960B |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.42 Remove Liteso (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int litesoRemove(int index) | |
| Feature | Remove a installed Liteso file | |
| Parameter | index [in] | Liteso index, range: 0~11 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is TOSS special, not supported on other device types.** | |

#### 3.3.43 Get RTC battery voltage

|  |  |  |
| --- | --- | --- |
| Prototype | int getRtcBatVol(Bundle info) | |
| Feature | Get RTC battery voltage | |
| Parameter | info[out] | RTC battery voltage information, contains following key:  vol: int, The voltage value, unit: mv (millivolt)  fromAdc: int, the origin of voltage value, 0- from cache, 1- from ADC |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. If the cumulative time has expired, it will enable the detection circuit to read the voltage of the button battery, update the cache value, and set the returned **fromAdc** value as 1  2. If the cumulative time does not expire, it will read the cached value of the button battery voltage, and set the returned **fromAdc** value as 0 | |

#### 3.3.44 Read verification PUK (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int readPuk(int pukType, Bundle info) | |
| Feature | Read verification PUK | |
| Parameter | pukType[in] | Puk type, 1-manufacturer，2-resaller (not support) |
| info[out] | The PUK information, contains following key:  pubKey: byte[], public key data, length is 264B, format is: modulus length(4B, MSB, value is LEN)+modulus(LEN B)+exponent(4B, MSB)  digestTime: long, effective date of the PUK, unit: ms  validDate: long, the expire data of the PUK, unit: ms  owner: String, the owner of this PUK |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.** | |

### 3.4 Card operation module

#### 3.4.1 ReadCard APIs

##### 3.4.1.1 Check card

|  |  |  |
| --- | --- | --- |
| Prototype | void checkCard(int cardType, CheckCardCallbackV2 callback, int timeout) | |
| Feature | For card checking, magnetic stripe card, IC card and NFC card are supported. After checking, the card type will be put into CheckCardCallbackV2. | |
| Parameter | cardType[in] | Composite card types, support checking multi cards. This param can be a composite value of CardType.getValue(), for example, check MAGNETIC,NFC,IC at the same time, set this param as:  CardType.MAGNETIC.getValue()|CardType.NFC.getValue()| CardType.IC.getValue() |
| callback[in] | Check card callback. Refer to [CheckCardCallback](#_3.4.3_CheckCardCallback_检卡回调对象)V2 |
| timeout[in] | Timeout (unit: second). Effective time range: 1-600s |
| Return | None | |
| Comment | 1.No distinction between bank card and non bank card.  2.If param **timeout**<=0, SDK use default value 60, and if **timeout**>120, SDK use default value 120  3. If cardType is a composite value, the default check card process is not stop when occurred error(except contact cards, for contact cards, it is always stop when occurred error)  **4. When checking magstripe card, this interface will return the cleartext track data of track 1,2,3, please be aware of the risks when using it.**  **5.For checking contact card, this interface will power-on and activate card, please call** [**cardOff()**](#CardOff) **to power-off card after card data exchange finished. If not deactive card for a long time, device power consumption will be high.**  **6. For checking contactless card, this interface will open the nfc reader and keep the open status, please call** [**cardOff()**](#CardOff) **to close nfc reader after card data exchange finished. If not close nfc reader for a long time, device power consumption will be high and device may overheat.**  7. If opened sred, CheckCardCallbackV2.onError() return error code -30001: read the card failed, refer to [SERD description](#SREDDescription) | |

##### 3.4.1.2 Cancel check card

Interface instructions: Artificial return must call cancelCheckCard(), terminate the underlying blocking thread, otherwise the next execution function will fail (for example, click the physical return key, click the interface navigation bar return key to call the function)

|  |  |
| --- | --- |
| Prototype | void cancelCheckCard() |
| Feature | Cancel check card |
| Parameter | None |
| Return | None |
| Comment | The function needs to be called when the CheckCard is not returned (CheckCardCallbackV2 Success or Failed Interface is not callback) and leaves the interface. |

##### 3.4.1.3 APDU command exchange

|  |  |  |
| --- | --- | --- |
| Prototype | int apduCommand (int cardType, ApduSendV2 send, ApduRecvV2 recv) | |
| Feature | IC card operation function. This function support the general interface protocol for IC card.(T=0 and T=1)  This support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardTypy[in] | Card type currently in operation |
| send[in] | class ApduSendV2 {  byte[] command; // Command[] = {CLA，INS，P1，P2}.  short lc; // Length of dataIn (0~256)  byte[] dataIn; // Data of the IC card which you want to send, max length is 256 bytes.  short le; // Expected length of the returned data (0~256)  } |
| recv[out] | class ApduRecvV2 {  short outLen; //Actual data length returned from IC card (0~256)  byte[] outData; // Data returned from IC card. max length is 256 bytes.  byte swa; // Status byte 1  byte swb; // Status byte 2  } |
| Return | 0: Success  Other value: Fail | |
| Comment | For variable format of ApduSendV2 , refer to the doc **apdu format and implement in sunmi way** | |

##### 3.4.1.4 APDU command exchange (not recommend)

|  |  |  |
| --- | --- | --- |
| Prototype | int smartCardExchange (int cardType, byte[] apduend, byte[] apduRecv) | |
| Feature | IC card operation function. This function support the general interface protocol for IC card.(T=0 and T=1)  This support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardTypy[in] | Card type currently in operation |
| apduSend[in] | The APDU command which to send, format is:  Command(4B)+LC(1B, value is **len**)+inData(**len** B)+LE(1B)  LC is unsigned, range is 0~255 |
| apduRecv[out] | The out data buffer, apduRecv.length>=260，receive data format is：  outLen(2B, Big endian,value is **len**)+outData(**len** B)+SWA(1B)+SWB(1B) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.4.1.5 Transmit APDU command to card

|  |  |  |
| --- | --- | --- |
| Prototype | int transmitApdu (int cardType, byte[] sendBuff, byte[] recvBuff) | |
| Feature | Transmit the **sendBuff** to card directly.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| sendBuff[in] | Data which want to transmit to card. Max length is 1929B |
| recvBuff [out] | Data received from card. The max receive data length is 2046B,  recvBuff.lenght>=2046 |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | 1. This interface not support transmit APDU for mifare Ultralight C card and mifare Ultralight EV serial cards, for mifare Ultralight C card and mifare Ultralight EV serial APDU transmission, please refer to interface [transmitApduEx()](#_3.4.1.14_Transmit_APDU) or [transmitApduExx()](#_3.4.1.17_Transmit_APDU) | |

##### 3.4.1.6 Power-off contact or contactless card

|  |  |  |
| --- | --- | --- |
| Prototype | int cardOff(int cardType) | |
| Feature | Power off contact IC card or turn off nfc carrier for contactless card. | |
| Parameter | cardType[in] | Card type, support composite card types, value can be one either of the following 2 types:  1. AidlConstants.CardType.IC.getValue()  2. AidlConstants.CardType.IC.getValue() | AidlConstants.CardType.NFC.getValue() |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. When **cardType** is a composite card type, if any card in the composite card type fails to power-off, the interface call fails, and the return value is the error code corresponding to the first card type that failed to power-off | |

##### 3.4.1.7 Check if card exist on slot

|  |  |  |
| --- | --- | --- |
| Prototype | int getCardExistStatus(int cardType) | |
| Feature | Check if card exist or not | |
| Parameter | cardType[in] | Card type (non-composite type), support only one card type each time |
| Return | >=0: Card exist status, refer to: [Card exist status(AidlConstants.CardExistStatus)](#_7.1.35_Card_exist)  <0: Error code | |
| Comment | None | |

##### 3.4.1.8 Mifare Classic

###### 3.4.1.8.1 Verify password of M1 card sector

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareAuth(int keyType, int block, byte[] key) | |
| Feature | Submit password A or B of responding read-write block when verifying the access of M1 card. | |
| Parameter | keyType[in] | Specified password type:  0 --- submit password A  1 --- submit password B |
| block[in] | Block number for specified access. The available range is 0~63 for M1 card with 1K. |
| key[in] | Point to submitted password buffer(6 Bytes) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.8.2 M1 card read block data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareReadBlock( int block, byte[] outData) | |
| Feature | Read the contents in the specified block of M1 card. (16 bytes) | |
| Parameter | block[in] | Block number for specified access. The available range is 0~63 for M1 card with 1K. |
| outData [out] | Block data. |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

###### 3.4.1.8.3 M1 card write block data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareWriteBlock( int block, byte[] data) | |
| Feature | Write specified content in specified block for M1 card (16 bytes). | |
| Parameter | block[in] | Block number for specified access. The available range is 0~63 for M1 card with 1K. |
| data[in] | Block data. |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.8.4 M1 card operate sector data (Increment operation)

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareIncValue( int block, byte[] value) | |
| Feature | Charge for wallet of M1 card. | |
| Parameter | block[in] | Use to assign the number of block where the wallet is in it. The available range is 0~63 for M1 card with 1K. |
| value[in] | Point to first address of buffer for the amount of charging. Totally 4 bytes for the amount and the lower byte is at the beginning. |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.8.5 M1 card operate sector data (decrement operation)

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareDecValue( int block, byte[] value) | |
| Feature | Charge back for wallet of M1 card. | |
| Parameter | block[in] | Use to assign the number of block where the wallet is in it. The available range is 0~63 for M1 card with 1K. |
| value[in] | Point to first address of buffer for the amount of charging back. Totally 4 bytes for the amount and the lower byte is at the beginning. |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.8.6 M1 card operate sector data (Increment operation, shrink method)

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareIncValueDx(int block, byte[] value) | |
| Feature | Charge for wallet of M1 card, not contains transfer operation. | |
| Parameter | block[in] | Use to assign the number of block where the wallet is in it. The available range is 0~63 for M1 card with 1K. |
| value[in] | Point to first address of buffer for the amount of charging. Totally 4 bytes for the amount and the lower byte is at the beginning. |
| Return | 0: Success  Other value: Fail | |
| Comment | M1 card charge for wallet contains 2 steps:  1.Charge, and cache the value data in data register  2.Transfer the cached value to block  Interface **mifareIncValue()** contains the 2 steps, **mifareIncValueDx()** contains step 1 only. | |

###### 3.4.1.8.7 M1 card operate sector data (decrement operation, shrink method)

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareDecValueDx(int block, byte[] value) | |
| Feature | Charge back for wallet of M1 card, not contains transfer operation. | |
| Parameter | block[in] | Use to assign the number of block where the wallet is in it. The available range is 0~63 for M1 card with 1K. |
| value[in] | Point to first address of buffer for the amount of charging back. Totally 4 bytes for the amount and the lower byte is at the beginning. |
| Return | 0: Success  Other value: Fail | |
| Comment | M1 card charge for wallet contains 2 steps:  1.Charge, and cache the value data in data register  2.Transfer the cached value to block  Interface **mifareDecValue()** contains the 2 steps, **mifareDecValueDx()** contains step 1 only. | |

###### 3.4.1.8.8 M1 card transfer data to block

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareTransfer(int destBlock) | |
| Feature | Transfer data register data to block | |
| Parameter | destBlock[in] | Block number |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.8.9 M1 card restore block data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareRestore(int srcBlock) | |
| Feature | Restore block data to data register | |
| Parameter | srcBlock[in] | Block number |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.4.1.9 Mifare Ultralight C

###### 3.4.1.9.1 Verify Password of Mifare Ultralight C Card Sector

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareUltralightCAuth(byte[] authKey) | |
| Feature | Submit password A or B of responding read-write block when verifying the access of Mifare ultralight c. | |
| Parameter | authKey[in] | Point to submitted password buffer |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

###### 3.4.1.9.2 Mifare Ultralight C Read Block Data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareUltralightCReadData(int block,byte[] outData) | |
| Feature | Read the contents in the specified block of Mifare ultralight c | |
| Parameter | block[in] | Block number |
| outData[out] | Block data. |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

###### 3.4.1.9.3 Mifare Ultralight C Write Block Data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifareUltralightCWriteData(int block,byte[] data) | |
| Feature | Write specified content in specified block for Mifare ultralight c. | |
| Parameter | block[in] | Block number |
| data[in] | Block data. |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.4.1.10 Mifare Plus SL3

###### 3.4.1.10.1 Mifare Plus read block data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifarePlusReadBlock(int block, byte[] key, byte[] outData) | |
| Feature | Mifare plus read block data. | |
| Parameter | block[in] | Block number for specified access. The available range is depends on card,  eg: MF1PLUS80x contains 32 4block sectoers and 8 16block secoters, each block size is 16 bytes, total memory is 32\*4\*16+8\*16\*16=4096, total block block count is 32\*4+8\*16=256, block range is 00~FF. |
| key[in] | The block key, 16 bytes. |
| outData[out] | Buffer, store the block data, **outData**.lenght>=16 |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

###### 3.4.1.10.2 Mifare Plus write block data

|  |  |  |
| --- | --- | --- |
| Prototype | int mifarePlusWriteBlock(int block, byte[] key, byte[] data) | |
| Feature | Mifare plus write block data. | |
| Parameter | block[in] | Block number for specified access. The available range is depends on card,  eg: MF1PLUS80x contains 32 4block sectoers and 8 16block secoters, each block size is 16 bytes, total memory is 32\*4\*16+8\*16\*16=4096, total block block count is 32\*4+8\*16=256, block range is 00~FF. |
| key[in] | The block key, 16 bytes. |
| data[in] | The data to be write, 16 bytes |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |

###### 3.4.1.10.3 Mifare Plus change block key

|  |  |  |
| --- | --- | --- |
| Prototype | int mifarePlusChangeBlockKey(int block, byte[] oldKey, byte[] newKey) | |
| Feature | Mifare plus change block key | |
| Parameter | block[in] | Block number for specified access. The available range is depends on card,  eg: MF1PLUS80x contains 32 4block sectoers and 8 16block secoters, each block size is 16 bytes, total memory is 32\*4\*16+8\*16\*16=4096, total block block count is 32\*4+8\*16=256, block range is 00~FF. |
| oldKey [in] | The old key, 16 bytes. |
| newKey [in] | The new key, 16 bytes |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |

##### 3.4.1.11 SLE4442/SLE4428

###### 3.4.1.11.1 SLE verify password

|  |  |  |
| --- | --- | --- |
| Prototype | int sleAuthKey(byte[] key) | |
| Feature | SLE verify password | |
| Parameter | key[in] | The password, SLE4442 is 3 bytes, SLE4428 is 2 bytes, default value is all ‘F’ |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |

###### 3.4.1.11.2 SLE change password

|  |  |  |
| --- | --- | --- |
| Prototype | int sleChangeKey(byte[] newKey) | |
| Feature | SLE change password | |
| Parameter | newKey[in] | The new password, SLE4442 is 3 bytes, SLE4428 is 2 bytes |
| Return | 0: Success  Other value: Error code | |
| Comment | This method only useful after calling **sleAuthKey()** sucess. | |

###### 3.4.1.11.3 SLE read data

|  |  |  |
| --- | --- | --- |
| Prototype | int sleReadData(int startAddress, int length, byte[] outData) | |
| Feature | SLE read a successive memory segment data | |
| Parameter | startAddress [in] | The start address |
| length[in] | The data length of read |
| outData[out] | Buffer, store the read data |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

###### 3.4.1.11.4 SLE write data

|  |  |  |
| --- | --- | --- |
| Prototype | int sleWriteData(int startAddress, byte[] data) | |
| Feature | SLE write a successive memory segment data | |
| Parameter | startAddress [in] | The start address |
| data[in] | The data to write, max length is 253 bytes |
| Return | 0: Success  Other value: Error code | |
| Comment | If card has password, this method only useful after calling **sleAuthKey()** sucess. | |

###### 3.4.1.11.5 SLE get remain authentication count

|  |  |
| --- | --- |
| Prototype | int sleGetRemainAuthCount() |
| Feature | SLE get remain authentication count |
| Parameter | None |
| Return | >=0: The remain authentication count  <0: Error code |
| Comment | If the remain authentication count is 0, card are locked and can not verify password or write data. |

###### 3.4.1.11.6 SLE write protection memory

|  |  |  |
| --- | --- | --- |
| Prototype | int sleWriteProtectionMemory(int startAddress, int length) | |
| Feature | SLE lock a successive memory segment, make it can not be written, no matter verify password success or not | |
| Parameter | startAddress[in] | The start address |
| length[in] | The length of memory |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |

###### 3.4.1.11.7 SLE read memory protection status

|  |  |  |
| --- | --- | --- |
| Prototype | int sleReadMemoryProtectionStatus(int startAddress, int length, byte[] dataOut) | |
| Feature | SLE read a successive memory protection status. | |
| Parameter | startAddress[in] | The start address |
| length[in] | The length of memory |
| dataOut[out] | Buffer, store each memory unit locking status, 0-locked, 1-not locked |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

##### 3.4.1.12 AT24C01/02/04/08/16/32/64/128/256/512

###### 3.4.1.12.1 AT24C read data

|  |  |  |
| --- | --- | --- |
| Prototype | int at24cReadData(int startAddress, int length, byte[] outData) | |
| Feature | AT24C read a successive memory segment data | |
| Parameter | startAddress[in] | The start address |
| length[in] | The length of memory |
| dataOut[out] | Buffer, store the read data |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |

###### 3.4.1.12.2 AT24C write data

|  |  |  |
| --- | --- | --- |
| Prototype | int at24cWriteData(int startAddress, byte[] data) | |
| Feature | AT24C write a successive memory segment data | |
| Parameter | startAddress[in] | The start address |
| data [in] | The data to write |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |

##### 3.4.1.13 AT88SC

###### 3.4.1.13.1 AT88SC verify password

|  |  |  |
| --- | --- | --- |
| Prototype | int at88scAuthKey(byte[] key, int rwFlag, int zoneNo) | |
| Feature | AT88SC verify password | |
| Parameter | key[in] | The password,AT88SC1608 is 3 bytes ,default value is all ‘F’ |
| rwFlag[in] | Write/Read flag, 0- The write key,1-The read key |
| zoneNo[in] | User zone number, range 0~7 |
| Return | 0: Success  Other value: Error code | |
| Comment | For AT88SC1608, there are 8 user zones(0~7), each zone contains 256 bytes, total 2048 bytes, address range 0~2047(0~7FF) | |

###### 3.4.1.13.2 AT24C88 change password

|  |  |  |
| --- | --- | --- |
| Prototype | int at88scChangeKey(byte[] newKey, int rwFlag, int zoneNo) | |
| Feature | AT88SC change password | |
| Parameter | newKey[in] | The new password,AT88SC1608 is 3 bytes |
| rwFlag[in] | Write/Read flag, 0-The write key, 1-The read key |
| zoneNo[in] | User zone number, range 0~7 |
| Return | 0: Success  Other value: Error code | |
| Comment | This method only useful after calling **at88scAuthKey()** sucess. | |

###### 3.4.1.13.3 AT88SC read data

|  |  |  |
| --- | --- | --- |
| Prototype | int at88scReadData(int startAddress, int length, int zoneFlag, byte[] outData) | |
| Feature | AT88SC read a successive memory segment data | |
| Parameter | startAddress [in] | The start address |
| length[in] | The data length of read |
| zoneFlag[in] | The zone flag, 0-Config zone,1-User zone |
| outData[out] | Buffer, store the read data |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |
| Example | None | |

###### 3.4.1.13.4 AT88SC write data

|  |  |  |
| --- | --- | --- |
| Prototype | int at88scWriteData(int startAddress, int zoneFlag, byte[] dataIn) | |
| Feature | SLE write a successive memory segment data | |
| Parameter | startAddress [in] | The start address |
| zoneFlag[in] | The zone flag, 0-Config zone,1-User zone |
| data[in] | The data to write, max length is 253 bytes |
| Return | 0: Success  Other value: Error code | |
| Comment | If card has password, this method only useful after calling **at88scAuthKey()** sucess. | |
| Example | None | |

###### 3.4.1.13.5 AT88SC get remain authentication count

|  |  |  |
| --- | --- | --- |
| Prototype | int at88scGetRemainAuthCount(int rwFlag, int zoneNo) | |
| Feature | AT88SC get remain authentication count | |
| Parameter | rwFlag[in] | Write/Read flag, 0-The write key, 1-The read key |
| zoneNo[in] | User zone number, range 0~7 |
| Return | >=0: The remain authentication count  <0: Error code | |
| Comment | If the remain authentication count is 0, card are locked and can not verify password or write data. | |
| Example | None | |

##### 3.4.1.14 Transmit APDU command to card(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int transmitApduEx(int cardType, byte[] sendBuff, byte[] recvBuff) | |
| Feature | Transmit the **sendBuff** to card directly.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| sendBuff[in] | Data which want to transmit to card. Max length is 1929B |
| recvBuff [out] | Data received from card. The max receive data length is 2046B,  recvBuff.lenght>=2046 |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | This method is different from **transmitApdu** only when cardtype is **Mifare**,the following text show the differences:  1. When card type is **Mifare**, the first byte(B1) of send data represents the setting of communication parameters:  Bit0: 1-enable Rx CRC, 0-disable Rx CRC  Bit1: 1-enable Tx CRC, 0-disable Tx CRC  Bit2: 0-enable Rx parity, 1-disable Rx parity  Bit3: 0-enable Tx parity, 1-disable Tx parity  Bit4-bit7: TxLastBits, TxLastBits = 0 - Send whole data of the last byte, TxLastBits = n (n≠0) - Send n bits of the last byte.  2.For interface **transmitApdu()**, param **sendBuff** should **not contains** B1. SDK defaults to send **0x03 + sendBuff** to card.  3. For interface **transmitApduEx()**, param sendBuff should **contains** B1, sendBuff[0] is B1,the format of B1 should comply to paragraph 1. | |

##### 3.4.1.15 CTX512B

###### 3.4.1.15.1 CTX512B read block data

|  |  |  |
| --- | --- | --- |
| Prototype | int ctx512ReadBlock(int block, byte[] outData) | |
| Feature | CTX512B card read block data | |
| Parameter | block[in] | Block number |
| outData[out] | Buffer, store the read data (2B) |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |
| Example | None | |

###### 3.4.1.15.2 CTX512B write block data

|  |  |  |
| --- | --- | --- |
| Prototype | int ctx512WriteBlock(int block, byte[] data) | |
| Feature | CTX512B card write block data | |
| Parameter | block[in] | Block number |
| data [in] | The data to be written(2B) |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |
| Example | None | |

###### 3.4.1.15.3 CTX512B update block data

|  |  |  |
| --- | --- | --- |
| Prototype | int ctx512UpdateBlock(int block, byte[] data) | |
| Feature | CTX512B card update block data | |
| Parameter | block[in] | Block number |
| data[in] | The data to be updated(2B) |
| Return | 0: Success  Other value: Error code | |
| Comment | None | |
| Example | None | |

###### 3.4.1.15.4 CTX512B get signature data

|  |  |  |
| --- | --- | --- |
| Prototype | int ctx512GetSignature(int block, byte[] random, byte[] outData) | |
| Feature | CTX512B card get signature data | |
| Parameter | block[in] | Block number |
| random[in] | Random data (6B) |
| outData[out] | Buffer, store the signature data (2B) |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |
| Example | None | |

###### 3.4.1.15.5 CTX512B read 4 successive blocks data

|  |  |  |
| --- | --- | --- |
| Prototype | int ctx512MultiReadBlock(int startBlock, byte[] outData) | |
| Feature | CTX512B card read 4 successive blocks data | |
| Parameter | startBlock[in] | The start block number |
| outData[out] | Buffer, store the read data (8B) |
| Return | >=0: The valid data length of **outData**  <0: Error code | |
| Comment | None | |
| Example | None | |

##### 3.4.1.16 Check card(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void checkCardEx(int cardType, int ctrCode, int stopOnError, CheckCardCallbackV2 checkCardCallback, int timeout) | |
| Feature | For card checking, magnetic stripe card, IC card and NFC card are supported. After checking, the card type will be put into CheckCardCallbackV2. | |
| Parameter | cardType[in] | Composite card types, support checking multi cards. This param can be a composite value of CardType.getValue(), for example, check MAGNETIC,NFC,IC at the same time, set this param as:  CardType.MAGNETIC.getValue()|CardType.NFC.getValue()| CardType.IC.getValue() |
| ctrCode[in] | Card active control code  Bit0-bit1: Contact card working voltage  0: VCC\_3000mV  1: VCC\_1800mV  2: VCC\_5000mV  3: Reserved  Bit2: The speed of reset on Contact CPU card or SAM card  0: SPD\_1X  1: SPD\_4X  Bit3: support PPS or not  0: NOT SUPPORT  1: SUPPORT  Bit4: The procedure of reach agreement on contact CPU card or SAM card  0: ICC\_SPEC  1: ICC\_EMV  Bit5: whether select the second protocol of card supported or not  0: not select  1: select |
| stopOnError[in] | Whether to stop immediately when checking special card type error (only take effect when cardType is composite value) |
| callback[in] | Check card callback. Refer to [CheckCardCallback](#_3.4.3_CheckCardCallback_检卡回调对象)V2 |
| timeout[in] | Timeout (unit: second). Effective time range: 1-600s |
| Return | None | |
| Comment | 1.No distinction between bank card and non bank card.  2.If param **timeout**<=0, SDK use default value 60, and if **timeout**>120, SDK use default value 120  3.Regardless of the value of param **stopOnError**, for contact cards, it is always stop when occurred error  **4. When checking magstripe card, this interface will return the cleartext track data of track 1,2,3, please be aware of the risks when using it.**  **5.For checking contact card, this interface will power-on and activate card, please call** [**cardOff()**](#CardOff) **to power-off card after card data exchange finished. If not deactive card for a long time, device power consumption will be high.**  **6. For checking contactless card, this interface will open the nfc reader and keep the open status, please call** [**cardOff()**](#CardOff) **to close nfc reader after card data exchange finished. If not close nfc reader for a long time, device power consumption will be high and device may overheat.**  7. If opened sred, CheckCardCallbackV2.onError() return error code -30001: read the card failed, refer to [SERD description](#SREDDescription) | |

##### 3.4.1.17 Transmit APDU command to card (second time extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int transmitApduExx(int cardType, int ctrCode, byte[] sendBuff, byte[] recvBuff); | |
| Feature | Transmit the **sendBuff** to card directly.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| ctrCode[in] | Card data exchange control code:  Bit0-Bit3: set contactless CPU card apdu fragment APDU wait time(fwi)   |  |  |  |  | | --- | --- | --- | --- | | fwi(bit0–bit3) | Wait time  (unit: ms) | fwi(bit0–bit3) | Wait time  (unit: ms) | | 0x0-0x3 | card specified time | 0xA | 309 | | 0x4 | 4.832 | 0xB | 618.5 | | 0x5 | 9.664 | 0xC | 1237 | | 0x6 | 19.3 | 0xD | 2474 | | 0x7 | 38.7 | 0xE | 4948 | | 0x8 | 77.3 | Other | card specified time | | 0x9 | 154.3 |  |  |   Bit4-bit5：contactless CPU card APDU repeat times  0-Not repeat  1-Repeat 1 time  2-Repeat 2 times  3-Reserved  bit6:  0: Enable automatic response this time (default);  1: Turn off automatic response this time (SCC0 SAM); |
| sendBuff[in] | Data to transmit to card. Max length is 1929B |
| recvBuff [out] | Data received from card. The max receive data length is 2046B,  recvBuff.lenght>=2046 |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | **If cardType is Mifare, prama sendBuff shoule has the same format as sendBuff of transmitApduEx().** | |

##### 3.4.1.18 Transmit multi APDUs to card

|  |  |  |
| --- | --- | --- |
| Prototype | int transmitMultiApdus(int cardType, int ctrCode, List<String> sendList, List<String> recvList) | |
| Feature | Transmit multi APDUs to card directly. At one time, the max send APDU number is 7.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| ctrCode[in] | Card data exchange control code:  Bit0-Bit3: set contactless CPU card apdu fragment APDU wait time(fwi)   |  |  |  |  | | --- | --- | --- | --- | | fwi(bit0–bit3) | Wait time  (unit: ms) | fwi(bit0–bit3) | Wait time  (unit: ms) | | 0x0-0x3 | card specified time | 0xA | 309 | | 0x4 | 4.832 | 0xB | 618.5 | | 0x5 | 9.664 | 0xC | 1237 | | 0x6 | 19.3 | 0xD | 2474 | | 0x7 | 38.7 | 0xE | 4948 | | 0x8 | 77.3 | Other | card specified time | | 0x9 | 154.3 |  |  |   Bit4-bit5：contactless CPU card APDU repeat times  0-Not repeat  1-Repeat 1 time  2-Repeat 2 times  3-Reserved |
| sendList[in] | APDU(Hex) list to transmit to card, each item is one APDU |
| recvList[out] | The respond data of each sent APDU from card(Hex) |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | **If cardType is Mifare, prama sendBuff shoule has the same format as sendBuff of transmitApduEx().** | |

##### 3.4.1.19 Check card with ciphertext track data

|  |  |  |
| --- | --- | --- |
| Prototype | int checkCardEnc(Bundle bundle, CheckCardCallbackV2 checkCardCallback, int timeout) | |
| Feature | For card checking, magnetic stripe card, IC card and NFC card are supported. After checking, the card type will be put into CheckCardCallbackV2.  **For card type CardType.MAGNETIC.getValue(), checkCardCallback return ciphertext track data** | |
| Parameter | bundle [in] | check card params, contains key:  cardType: int, card type, composite value, eg:  CardType.MAGNETIC.getValue()|CardType.NFC.getValue()  |CardType.IC.getValue()  encKeySystem: int, the encryption key system, refer to [KEY\_SYSTEM\_CONSTANTS](#KeySystemTypeConstantDefinition). Only support SEC\_MKSK, SEC\_DUKPT, SEC\_RSA  encKeyIndex: int**,** the encryption key index, refer to [key system and key index range](#KeySystemAndKeyIndexRange).  encKeyAlgType：int, the algorithm of encryption key, refer to [Key algorithm type constant](#KeyAlgoritmTypeConstantDefinition). Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encMode: int, Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encIv: byte[], IV(initial vector), if **encMode** is ECB，value is **null**, otherwise value is 8 or 16 bytes data. Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encPaddingMode: byte, The padding mode on encryption. If **encKeySystem** is SEC\_MKSK or SEC\_DUKPT, and encryption key is DES/AES/SM4, and track data length is not a multiply of 8 or 16, then padding track data at tail with **encPaddingMode** to a multiple of 8 or 16 in length, if **encKeySystem** is SEC\_RSA, padding mode refer to [RSA Padding Mode](#RSAPaddingMode), mode NOTHING\_PADDING is unsupported  encMaskStart: int, 0~6, how much characters are plaintext in front of PAN  encMaskEnd: int, 0~4, how much characters are plaintext in rear of PAN  encMaskWord: char, 0 or non-digit character，The mask character for **EncMaskStart**~**encMaskWord** PAN, default is ‘**\***’  panAppendContent: String, the appended data to track2 before RSA encrypt track data (TID)  panAppendMode: int, the concatenate mode of **panAppendContent(TID)** and track2 before RSA encryption, 0- TID+track2, 1-track2+TID  ctrCode: int, card active control code(default value is 0):  b0~b1:00-VCC\_3000mV, 01-VCC\_1800mV, 02-VCC\_5000mV, 03-reserved  b2:0-SPD\_1X,1-SPD\_4X  b3: Is support PPS,0-not support,1-support  b4: contact CPU card and SAM card protocol procedure,0-ICC\_SPEC,1-ICC\_EMV  stopOnError: int, whether to stop immediately or not when checking special card type error, 0-not stop, 1-stop (only take effect when cardType is composite value) |
| callback[in] | Check card callback. Refer to [CheckCardCallbackV2](#CheckCardCallbackV2) |
| timeout[in] | Timeout (unit: second). Effective time range: 1-600s |
| Return | None | |
| Comment | 1.No distinction between bank card and non bank card.  2.This interface is not supported on device type TOSS  3.If param **timeout**<=0, SDK use default value 60, and if **timeout**>120, SDK use default value 120  4.Regardless of the value of param **stopOnError**, for contact cards, it is always stop when occurred error  **5.For checking contact card, this interface will power-on and activate card, please call** [**cardOff()**](#CardOff) **to power-off card after card data exchange finished. If not deactive card for a long time, device power consumption will be high.**  **6. For checking contactless card, this interface will open the nfc reader and keep the open status, please call** [**cardOff()**](#CardOff) **to close nfc reader after card data exchange finished. If not close nfc reader for a long time, device power consumption will be high and device may overheat.**  7. Open sred has no effect to this interface, refer to [SERD description](#SREDDescription) | |

##### 3.4.1.20 Set smart card related params

|  |  |  |
| --- | --- | --- |
| Prototype | int smartCardIoControl(int cardType, int ctrCode, byte[] dataIn, byte[] dataOut) | |
| Feature | Set smart card related params | |
| Parameter | cardType [in] | The card type |
| ctrCode [in] | The control code, has following values:  0-Set system code of Felica cad polling command (default is 0xffff),2B,Big endian  1-Set contactless apdu timeout time, unit: ms, 4B, Big endian  2-Get contactless register config(TLV format)  3-Set contactless register config(TLV format)  4-Get contactless param config(TLV format)  5-Set contactless param config(TLV format) |
| dataIn[in] | The input data, refer to **ctrCode** |
| dataOut[out] | Buffer, store output data. |
| Return | >=0: The valid data length in dataOut  <0: Error code | |
| Comment | None | |
| Example | None | |

##### 3.4.1.21 Check card for TOSS (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | void checkCardForToss(Bundle bundle, CheckCardCallbackV2 callback, int timeout) | |
| Feature | For card checking, magnetic stripe card, IC card and NFC card are supported. After checking, the card type will be put into CheckCardCallbackV2. | |
| Parameter | bundle [in] | check card params, contains key:  cardType: int, card type, composite value, eg:  CardType.MAGNETIC.getValue()|CardType.NFC.getValue()  |CardType.IC.getValue()  ctrCode: int, Card active control code, default value is 0  Bit0-bit1: Contact card working voltage:  0: VCC\_3000mV  1: VCC\_1800mV  2: VCC\_5000mV  3: Reserved  Bit2: The speed of reset on Contact CPU card or SAM card  0: SPD\_1X  1: SPD\_4X  Bit3: support PPS or not  0: NOT SUPPORT  1: SUPPORT  Bit4: The procedure of reach agreement on contact CPU card or SAM card  0: ICC\_SPEC  1: ICC\_EMV  Bit5: whether select the second protocol of card supported or not  0: not select  1: select  code: int, vanCode, range: 0x01-KICC, 0x02-NICE, 0x03-KIS, 0x04-SMATRO, 0x05-KSNET  type: int, card type, range: 'I'-IC, 'M'-MS/Fallback, 'K'-KeyIn, 'B'-Barcode  maskStart: int, how much characters are plaintext in front of PAN, range: 0~8  maskEnd: int, how much characters are plaintext in rear of PAN, range: 0~4, as default, the last number of PAN will be masked  maskChar: char, the mask character for **maskStart** ~ **maskEnd** PAN, default is ‘**\***’, 0 or non-digit character，if set as characters between ‘0’~ ‘9’, the set value will be replaced with‘\*’  stopOnError: int, Whether or not to stop immediately when checking special card type error, 0-not stop, 1-stop (only take effect when cardType is composite value) |
| callback[in] | Check card callback. Refer to [CheckCardCallbackV2](#CheckCardCallbackV2) |
| timeout[in] | Timeout (unit: second). Effective time range: 1-600s |
| Return | None | |
| Comment | 1.Not distinction between bank card and non bank card.  2.This interface only return track2 data, not return track1, track3  3. If got correct track2 data, check card will return:  (1) The encrypted track2 data  (2) The masked PAN in track2  (3) The service code in track2  4. If track2 has no data or track2 decode failure, and track3 has correct data, track3 will replace track2 and returned in step 2.  5.If param **timeout**<=0, SDK use default value 60, and if **timeout**>120, SDK use default value 120  6.Regardless of the value of param **stopOnError**, for contact cards, it is always stop when occurred error  **7.For checking contact card, this interface will power-on and activate card, please call** [**cardOff()**](#CardOff) **to power-off card after card data exchange finished. If not deactive card for a long time, device power consumption will be high.**  **8.For checking contactless card, this interface will open the nfc reader and keep the open status, please call** [**cardOff()**](#CardOff) **to close nfc reader after card data exchange finished. If not close nfc reader for a long time, device power consumption will be high and device may overheat.**  9. If opened sred, CheckCardCallbackV2.onError() return error code -30001: read the card failed, refer to [SERD description](#SREDDescription)  **10.This interface is TOSS special, not supported on other device types.** | |

##### 3.4.1.22 PASS mode transmit APDU to card

|  |  |  |
| --- | --- | --- |
| Prototype | int smartCardExChangePASS(int cardType, byte[] apduSend, byte[] apduRecv) | |
| Feature | PASS mode transmit the **sendBuff** to card directly.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| sendBuff[in] | Data which want to transmit to card. Max length is 255B |
| recvBuff [out] | Data received from card.  apduRecv.length>=260, receive data format is：  outLen(2B, Big endian,value is **len**)+outData(**len** B)+SWA(1B)+SWB(1B) |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | None | |

##### 3.4.1.23 PASS mode transmit APDU to card, return data not contains Length field

|  |  |  |
| --- | --- | --- |
| Prototype | int smartCardExChangePASSNoLength(int cardType, byte[] apduSend, byte[] apduRecv) | |
| Feature | PASS mode transmit the **sendBuff** to card directly.  This function support the general interface protocol for IC card.(T=0 and T=1), and also support the general interface protocol for contactless card. (T=CL) | |
| Parameter | cardType[in] | Card type currently in operation |
| sendBuff[in] | Data which want to transmit to card. Max length is 255B |
| recvBuff [out] | Data received from card.  apduRecv.length>=260, receive data format is：  outData(**len** B)+SWA(1B)+SWB(1B) |
| Return | >=0: The valid data length in recvBuff  <0: Error code | |
| Comment | None | |

#### 3.4.2 CheckCardCallbackV2

##### 3.4.2.1 Find magnetic card

|  |  |  |
| --- | --- | --- |
| Prototype | void findMagCard(Bundle info) | |
| Feature | Find magnetic stripe card. | |
| Parameter | info[in] | Contains the following data:  cardType: int, the card type  TRACK1: String, track1 data  TRACK2: String, track2 data  TRACK3: String, track3 data  track2Raw: byte[], track2 data(returned by checkCardForToss())  pan: String, PAN data (returned by checkCardEnc() )  name: String, cardhold name (returned by checkCardEnc() )  expire: String, card expire date (returned by checkCardEnc() )  servicecode: String, card service code (returned by checkCardEnc() )  appendedPanEnc: String, RSA encrypted TID+track2 or track2+TID data, Hex String format (returned by checkCardEnc())  appendedPanEncBytes: byte[], RSA encrypted TID+track2 or track2+TID data (returned by checkCardEnc())  track1ErrorCode: int, track1 error code  track2ErrorCode: int, track2 error code  track3ErrorCode: int, track3 error code  The track error code can be one of the following values:  -1: Track no data  -2: Track parity check error  -3: Track LRC check error |
| Return | None | |
| Comment | None | |

##### 3.4.2.2 Find IC card

|  |  |  |
| --- | --- | --- |
| Prototype | void findICCard(String atr) | |
| Feature | Find IC card | |
| Parameter | atr[in] | The card ATR |
| Return | None | |
| Comment | None | |

##### 3.4.2.3 Find radio frequency card

|  |  |  |
| --- | --- | --- |
| Prototype | void findRFCard(String uuid) | |
| Feature | Find RF Card | |
| Parameter | uuid[in] | The card UUID |
| Return | None | |
| Comment | None | |

##### 3.4.2.4 Check card error

|  |  |  |
| --- | --- | --- |
| Prototype | void onError(int code, string message) | |
| Feature | Check card error callback. | |
| Parameter | code[in] | Error code |
| message[in] | Error message |
| Return | None | |
| Comment | None | |

##### 3.4.2.5 Find IC card (extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void findICCardEx(Bundle info) | |
| Feature | Find IC card | |
| Parameter | info[in] | Contains the following data:  cardType: int, the checked card type  atr: String, the card ATR |
| Return | None | |
| Comment | Compare with **findICCard()**, This method provide more detail info in param [info], Both this method and **findICCard()** will be called in check card process. Client app can implement one of the two to do process. | |

##### 3.4.2.6 Find radio frequency card (extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void findRFCardEx(Bundle info) | |
| Feature | Find RF Card | |
| Parameter | info[in] | Contains the following data:  cardType: int, the checked card type  uuid: String, the card UUID  ats: String, the card ATS  cardCategory: int, the card category, ‘A’or ‘B’  atqa: byte[], the card ATQA |
| Return | None | |
| Comment | Compare with **findRFCard()**, This method provide more detail info in param [info], Both this method and **findRFCard()** will be called in check card process. Client app can implement one of the two to do process. | |

##### 3.4.2.7 Check card error (extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void onErrorEx(Bundle info) | |
| Feature | Check card error callback. | |
| Parameter | info[in] | Contains the following data:  cardType: int, the failed card type  code: int, the error code  message: String, the error message |
| Return | None | |
| Comment | Compare with **onError()**, This method provide more detail info in param [info], Both this method and **onError()** will be called in check card process. Client app can implement one of the two to do process. | |

### 3.5 PinPad operation module (Unsupported for TOSS)

#### 3.5.1 PinPad APIs

##### 3.5.1.1 Initialize PinPad

|  |  |  |
| --- | --- | --- |
| Prototype | String initPinPad(inPadConfigV2 config, PinPadListenerV2 listener) | |
| Feature | Initialize PinPad: import configuration parameter and register callback listener. | |
| Parameter | config[in] | PinPad configuration, refer to [PinPadConfigV2](#PinPadConfigV2)  If set PinPadConfigV2.pinPadType as 0, an SDK built-in PinPad will be shown |
| listerner[in] | Callback object of Inputing PIN, refer to [PinPadListenerV2](#PinPadListenerV2) |
| Return | Key numbers in order or disorder sequence | |
| Comment | 1. For Sunmi PinBlock format, please refer to **Sunmi PinBlock format**  2. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to start visualImpairement keyboard will fail | |

##### 3.5.1.2 Import PinPad data

|  |  |  |
| --- | --- | --- |
| Prototype | void importPinPadData(PinPadDataV2 data) | |
| Feature | Import PinPad layout data | |
| Parameter | data[in] | PinPad layout data, refer to [PinPadDataV2](#PinPadDataV2) |
| Return | None | |
| Comment | 1. The client app only needs to call this interface when using a customized keyboard (when calling initPinPad() , set config.pinpadType=1)  2. The UI layout and key mapping of the custom keyboard are all implemented by the client app itself  3. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to import virtual keyboard layout data has no effect | |

##### 3.5.1.3 Cancel input PIN

|  |  |
| --- | --- |
| Prototype | void cancelInputPin() |
| Feature | Cancel input PIN. |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.5.1.4 Set PinPad showing text

|  |  |  |
| --- | --- | --- |
| Prototype | void setPinPadText(PinPadTextConfigV2 config) | |
| Feature | Set PinPad showing text | |
| Parameter | config[in] | Showing text config, refer to [PinPadTextConfigV2](#PinPadTextConfigV2) |
| Return | None | |
| Comment | 1. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to set PinPad text has no effect | |

##### 3.5.1.5 Set PinPad mode

|  |  |  |
| --- | --- | --- |
| Prototype | int setPinPadMode(Bundle bundle) | |
| Feature | Set PinPad mode | |
| Parameter | bundle[in] | PinPad mode config, contain the following keys:  normal: int, normal mode (0-disable, 1-enable)  longPressToClear: int, long press clear key to clear while inputting PIN (0-disable, 1-enable)  silent: int, keep silent while inputting PIN (0-disable, 1-enable)  greenLed: int, lit green LED while inputting PIN (0-disable, 1-enable)  monitorClearKey: int, if set, client receive PinPadListenerV2.onPinLength(0) callback when press clear key directly without inputing any number (0-disable, 1-enable)  cancelToClear: int, press cancel key to clear input key number(0-disable, 1-enable)  visualImpairment：int，visually impaired mode (0-off, 1-on. This mode and other modes have mutual exclusion)  longTimeoutTime: int, the maximum input PIN timeout time is 10 minutes(0-disable, 1-enable) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.The default mode is Normal mode (Key sound+ short prcess to clear + lit green LED)  2.**Normal mode** mutex to other modes, and if set normal as 1, other value will be discarded  3.The set mode only valid for next time input PIN, and after input PIN finished, SDK auto restore mode to **Normal mode**.  4. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to set PinPad as visualImpairement mode has no effect | |

##### 3.5.1.6 Get PinPad mode

|  |  |  |
| --- | --- | --- |
| Prototype | int getPinPadMode(Bundle bundle) | |
| Feature | Get current PinPad mode | |
| Parameter | bundle[out] | PinPad mode config, contain the following keys:  normal: int, normal mode (0-disable, 1-enable)  longPressToClear: int, long press clear key to clear while inputting PIN (0-disable, 1-enable)  silent: int, keep silent while inputting PIN (0-disable, 1-enable)  greenLed: int, lit green LED while inputting PIN (0-disable, 1-enable)  monitorClearKey: int, if set, client receive PinPadListenerV2.onPinLength(0) callback when press clear key directly without inputing any number (0-disable, 1-enable)  cancelToClear: int, press cancel key to clear input key number(0-disable, 1-enable)  longTimeoutTime: int, the maximum input PIN timeout time is 10 minutes(0-disable, 1-enable) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.5.1.7 Initialize PinPad (extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | String initPinPadEx(Bundle config, PinPadListenerV2 listener) | |
| Feature | Initialize PinPad | |
| Parameter | config[in] | PinPad config, contain following keys:  pinPadType: int, 0-SDK built-in normal PinPad(defualt),  1-Client customized normal PinPad,  2-SDK built-in blind PinPad,  3-SDK built-in rnib auth blind PinPad,  4-SDK built-in rnib auth normal PinPad,  5-Client customized blind PinPad  pinType: int, 0-online PIN, 1-offline PIN  isOrderNumKey: int, 0-random order keyboard (default), 1-in order keybaord  pan: byte[], ASCII to bytes, eg:”123456”.getBytes("US-ASCII"),length 12~19  pinKeyIndex: int, PIK (PIN key) index  minInput: int, default 0, minimum input PIN numbers  maxInput: int, default 6, maximum input PIN numbers  inputStep: int, default 1, input PIN step  timeout: int, unit: ms, default 60000, input PIN timeout time  isSupportbypass: int, 0-not support, 1-support(default)  pinblockFormat: int, default 0, refer to [PINBLOCK\_FORMAT](#PinBlock_format)  algorithmType: int, 0-3DES(8B PinBlock), 1-SM4(16B PinBlock), 2-AES(16B PinBlock), algorithm of ecryption PIN  keySystem: int, 0-SEC\_MKSK(default), 1-SEC\_DUKPT  diversify: byte[], 3DES PIK operate with diversify to generate new PIK, then use new PIK to calculate PinBlock |
| listener[in] | Callback object of Inputing PIN, refer to [PinPadListenerV2](#PinPadListenerV2) |
| Return | Key numbers in order or disorder sequence | |
| Comment | 1. Currently, pinPayType not support type 3 and 4  2. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to start visualImpairement keyboard will fail | |

##### 3.5.1.8 Set PIN anti-exhaustive protection mode

|  |  |  |
| --- | --- | --- |
| Prototype | int setAntiExhaustiveProtectionMode(int level) | |
| Feature | Set PIN anti-exhaustive protection mode | |
| Parameter | level[in] | Protection level, range 1-5, each value indicates max input PIN times in  specified time range:  1-2 minutes input 4 times  2-6 minutes input 12 times  3-15 minutes input 30 times  4-30 minutes input 60 times  5-60 minutes input 120 times |
| Return | >=0: The time should to wait before new mode take effect, 0 means new mode take effect immediately  Other value: Fail | |
| Comment | None | |

##### 3.5.1.9 Get PIN anti-exhaustive protection mode

|  |  |
| --- | --- |
| Prototype | int getAntiExhaustiveProtectionMode() |
| Feature | Get PIN anti-exhaustive protection mode |
| Parameter | None |
| Return | >=0: Current PIN anti-exhaustive protection level, range 1-5  Other value: Fail |
| Comment | None |

##### 3.5.1.10 Set Visual Impairment Mode Param

|  |  |  |
| --- | --- | --- |
| Prototype | int setVisualImpairmentModeParam(Bundle param) | |
| Feature | Set Visual Impairment Mode Param | |
| Parameter | bundle [in] | PinPad mode params, contain following keys:  timeoutGap1：int，screen touch time，0~100，unit：100ms, default: 10  timeoutGap2：int，time between two screen taps，0~100，unit：100ms, default: 10  ttsLanguage：int，language of the voice announcement (0-follow system (default), 1-English, 2-Polish, 3-French, 4-Portugal, 5-chinese, 6-Spanish)  rnibSelectMode: int, PIN number confirm mode, 0-double tap to confirm(default), 1-long press to confirm  rnibHoldTime：int, the necessary press time of long press to confirm mode, 0~100，unit: 100ms, default: 30 |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. If key in param **bundle** is absent, SDK use default value for it, eg: if key **rnibHoldTime** is absent, SDK use 30(default vaule) to set **rnibHoldTime**  2. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to set visualImpairement param has no effect | |

##### Get Visual Impairment Mode Param

|  |  |  |
| --- | --- | --- |
| Prototype | int getVisualImpairmentModeParam(Bundle param) | |
| Feature | Get Visual Impairment Mode Param | |
| Parameter | bundle [out] | PinPad mode params, contain following keys:  timeoutGap1：int，screen touch time，0~100，unit：100ms  timeoutGap2：int，time between two screen taps，0~100，unit：100ms  ttsLanguage：int，language of the voice announcement (type: int, 0-follow system (default), 1-English, 2-Polish, 3-French, 4-Portugal, 5-chinese, 6-Spanish)  rnibSelectMode: int, PIN number confirm mode, 0-double tap to confirm(default), 1-long press to confirm  rnibHoldTime：int, the necessary press time of long press to confirm mode, 0~100，default: 30, unit: 100ms |
| Return | 0: Success  Other value: Fail | |
| Comment |  | |

##### 3.5.1.12 Start input PIN

|  |  |  |
| --- | --- | --- |
| Prototype | int startInputPin(Bundle bundle, PinPadListenerV2 listener) | |
| Feature | start input PIN | |
| Parameter | bundle [in] | Contain following keys:  pinPadType: int, 0-SDK built-in normal PinPad(defualt),  1-Client customized normal PinPad,  2-SDK built-in blind PinPad,  3-SDK built-in rnib auth blind PinPad,  4-SDK built-in rnib auth normal PinPad,  5-Client customized blind PinPad  pinType：int, pint type(0-online,1-offline)  isOrderNumKey：int, is order keyboard(0-disorder keyboard,1-order keyboard)  minInput: int, the minimum number of input(default 0)  maxInput: int, the maximum number of input(default 6)  inputStep:int, PIN step(default 1)  expLen:int, the number of PIN digits allowed to be input, split by ",", e.g. "0,4,6" means 0/4/6 digits allowed to be input (type: String, this field is mutually exclusive with minInput/maxInput/inputStep, if both exist, expLen will be used first)  isSupportbypass: int, whether to support bypass PIN (type: int, 0 - not supported, 1 - supported (default))  timeout: int, timeout time, unit: ms (default 60000) |
| listener | startInputPin callback |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. Currently, pinPayType not support type 3 and 4  2. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to start visualImpairement keyboard will fail | |

##### 3.5.1.13 Get PIN block

|  |  |  |
| --- | --- | --- |
| Prototype | int getPinBlock(Bundle bundle, byte[] dataOut) | |
| Feature | get pinblock | |
| Parameter | bundle [in] | Contain following keys:  keySystem：int，key system(0-MKSK,1-DUKPT)  pinKeyIndex：int, PIK(PIN key index)  algorithmType：int, encryption Pin algorithm type ( 0-3DES (return 8 bytes PinBlock), 1-SM4 (return 16 bytes PinBlock), 2-AES (return 16 bytes PinBlock))  pinblockFormat: int, PinBlock type(default 0)  pan: byte[], ASCII format converted to byte for example "123456".getBytes("US-ASCII") |
| dataOut [out] | pinblock result |
| Return | >=0: length of valid data in dataOut  <0: Fail | |
| Comment | **1.Before invoking this interface, the interface startInputPin() should be invoked firstly, and the App which invoking this interface should be the same App which invoking startInputPin()** | |

##### 3.5.1.14 Offline PIN verify (Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int offlinePinVerify(Bundle paramIn, Bundle paramOut) | |
| Feature | Veify offline pin | |
| Parameter | paramIn [in] | Contain following keys:  offlineType：int，offline type(0-offline plaintext,1-offline ciphertext)  modulus：byte [], RSA key modulus  exponent：byte [], RSA key exponent  random: byte [], random number |
| paramOut [out] | Contain following keys:  sw1：int, SW1  sw2：int, SW2 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is Brazil-CKD special, not support on other device types.** | |

##### 3.5.1.15 Import PinPad data (extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void importPinPadDataEx(PinPadDataV2Ex data) | |
| Feature | Import PinPad layout data | |
| Parameter | data[in] | PinPad layout data, refer to [PinPadDataV2Ex](#PinPadDataV2Ex) |
| Return | None | |
| Comment | 1. The client app only needs to call this interface when using a customized keyboard (when calling initPinPad() , set config.pinpadType=1)  2. The UI layout and key mapping of the custom keyboard are all implemented by the client app itself  3. Compare to PinPadDataV2, PinPadDataV2Ex added enterX, enterY, enterH, enterW fields to support customize Enter key position, and added clearX, clearY, clearH, clearW fields to support customize Clear key position  4. For device with physical keyboard(eg: P2\_smartPad, P3K), only physical PIN keyboard is supported, virtual PIN keyboard and visualImpairement PIN keyboard are not supported, call this interface to import virtual keyboard layout data has no effect | |

#### 3.5.2 PinPadListenerV2

**Note: This interface is the interface passed between AIDLs and must be implemented in accordance with the Aidl interface when the callback is passed.**

##### 3.5.2.1 Press number key

|  |  |  |
| --- | --- | --- |
| Prototype | void onPinLength(int len) | |
| Feature | Import the PIN length, which is used for display when input PIN. | |
| Parameter | len[in] | Length of PIN entered |
| Return | None | |
| Comment | None | |

##### 3.5.2.2 Press confirm key

|  |  |  |
| --- | --- | --- |
| Prototype | void onConfirm(int type, byte[] pinBlock) | |
| Feature | Press confirm key, return PinBlock.  (1) type = 0 (online PIN):  (a) pinBlock=null: bypass mode (the user did not enter pin and press confirm key directly)  (b) pinBlock !=null: the returned PinBlock, length is 8B(3DES) or 16B(SM4/AES)  (2) type = 1(offline PIN): pinBlock length is always 1, and pinBlock[0] is 0, pinBlock is meaningless in this case. | |
| Parameter | type[in] | Pin type, 0-online PIN, 1-offline PIN. |
| pinBlock [out] | Pin block data |
| Return | None | |
| Comment | **Offline PIN is verified by card directly, return PinBlock to client is not necessary, SDK just return 1 byte array to mark this situation.** | |

##### 3.5.2.3 Press cancel key

|  |  |
| --- | --- |
| Prototype | void onCancel() |
| Feature | Cancel PIN input |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.5.2.4 Input pin error

|  |  |  |
| --- | --- | --- |
| Prototype | void onError (int errorCode) | |
| Feature | Error message | |
| Parameter | errorCode [in] | Error code, refer to [Error Code Definition](#Error_Code_Definition) |
| Return | None | |
| Comment | None | |

##### 3.5.2.5 Finger hover on key button in visualImpairement mode PinPad

|  |  |  |
| --- | --- | --- |
| Prototype | void onHover(int event, byte[] data) | |
| Feature | Finger hover on button in visualImpairement mode PinPad | |
| Parameter | event[in] | Finger event:  4-Finger touched ENTER key  5-Finger touched CLEAR key  6-Finger touched CANCEL key  7-Finger touched invalid area in keyboard(deprecated)  8-Inputted PIN length has reached the upper limit  9-Finger touched number key (need play beep)  10-Finger touched the upper invalid area in keyboard  11-Finger touched the below invalid area in keyboard(unsupported)  12-Finger touched the left invalid area in keyboard  13-Finger touched the right invalid area in keyboard |
| data[in] | The data of event:  1. event=5 or 6, data[0] is the inputted number length  2. event !=5 and 6, data is null, this is a invalid value and should be ignored |
| Return | None | |
| Comment | None | |

### 3.6 Security operation module

#### 3.6.1 Save plaintext key

|  |  |  |
| --- | --- | --- |
| Prototype | int savePlaintextKey(int keyType, byte[] keyValue, byte[] checkValue, int keyAlgType, int keyIndex) | |
| Feature | Save a plaintext key | |
| Parameter | keyType[in] | Key type. Refer to: [Aidl constants Key type constant definition](#KeyTypeConstantDefinition) |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition).. |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.When saving TMK, Client can specify the **encryptIndex** as a TMK (which already stored) index, this can support multi-layer TMK.  2.When isEncrypt is false, the decryption key parameter is 1 by default.  3.Client should guarantee key index mutex, that is, if stored a MAC key in index 4, then store a track key in index 4 too, the MAC key will be overwritten by the track key.  4.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  5.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **6.This interface is not supported on TOSS device** | |

#### 3.6.2 Save ciphertext key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveCiphertextKey(int keyType, byte[] keyValue, byte[] checkValue, int encryptIndex, int keyAlgType, int keyIndex) | |
| Feature | Save ciphertext key | |
| Parameter | keyType[in] | Key Type: Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition). KEY\_TYPE\_KEK Type cannot be saved in ciphertext |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| encryptIndex[in] | Used to decrypt the key ciphertext index, note that here is the index of TMK. |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition). |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.When saving TMK, Client can specify the **encryptIndex** as a TMK (which already stored) index, this can support multi-layer TMK.  2.When isEncrypt is false, the decryption key parameter is 1 by default.  3.Client should guarantee key index mutex, that is, if stored a MAC key in index 4, then store a track key in index 4 too, the MAC key will be overwritten by the track key.  4.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  5.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **6.This interface is not supported on TOSS device** | |

#### 3.6.3 Calculate MAC

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMac (int keyIndex, int macType , byte[] dataIn, byte[] dataOut) | |
| Feature | Calculate MAC | |
| Parameter | keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| macType[in] | MAC encryption algorithm, refer to: Aidl constant.[MAC algorithm type](#MACAlgorithmConstantDefinition) |
| dataIn[in] | The original data |
| dataOut[out] | MAC result |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.DataIn is the packet data to be computed. Many MAC algorithms have been implemented in the SDK. The results will be returned through dataOut, and the incoming dataOut will be fixed to 8 bytes long byte such as: byte [] dataOut = new byte [8].  **2.This interface is not supported on TOSS device** | |

#### 3.6.4 Encrypt data

|  |  |  |
| --- | --- | --- |
| Prototype | int dataEncrypt (int keyIndex , byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Encrypt data | |
| Parameter | keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| datain[in] | The original data |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 byte vector for other encryption modes. |
| dataout[out] | Encryption data |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg：byte[] dataIn = new byte[16];  dataOut length should be 16  **3.This interface is not supported on TOSS device** | |

#### 3.6.5 Decrypt data

|  |  |  |
| --- | --- | --- |
| Prototype | int dataDecrypt(int keyIndex , byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Decrypt data | |
| Parameter | keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| dataIn[in] | Ciphertext data |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 byte vector for other encryption modes. |
| dataOut[out] | plaintext |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg：byte[] dataIn = new byte[16];  dataOut length should be 16  **3.This interface is not supported on TOSS device** | |

#### 3.6.6 Save dukpt key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveKeyDukpt(int keyType, byte[] keyValue, byte[] checkValue, byte[] ksn, int encryptType, int keyIndex) | |
| Feature | Save dukpt key | |
|  | keyType[in] | Key type Contain:  BDK- base dispersed key: corresponding to Aidl constant class key type constant KEY\_TYPE\_DUPKT\_BDK.  IPEK- initial PIN encryption key: corresponding Aidl constant class key type constant KEY\_TYPE\_DUPKT\_IPEK. |
| Parameter | keyValue[in] | Key data |
| checkValue[in] | Key check value(When injecting BDK null values). Refer to [Key check value param description](#KcvParamDescription) |
| ksn[in] | KSN |
| encryptType[in] | Key algorithm type, refer to: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition) |
| keyIndex[in] | Key index , range: refer to [key system and key index range.SEC\_DUKPT.3DES](#KeySystemAndKeyIndexRange_SEC_DUKPT_3DES) |
| Return | 0: Success  <0: Fail | |
| Comment | 1.This interface can only inject DES keys, and the key length must be equal to 16  2.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  3.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **2.This interface is not supported on TOSS device** | |

#### 3.6.7 Calculate mac (Based on the DUKPT)

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMacDukpt(int keyIndex, int macType , byte[] dataIn, byte[] dataOut) | |
| Feature | Dukpt key calculate mac. | |
| Parameter | keyIndex [in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| macType [in] | Mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition) |
| dataIn [in] | The original data |
| dataOut[out] | Mac result. |
| Return | 0: Success  <0: Fail | |
| Comment | 1.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1016  **2.This interface is not supported on TOSS device** | |

#### 3.6.8 Encrypt data (Based on the DUKPT)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataEncryptDukpt(int keyIndex, byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Dukpt key encrypt data | |
| Parameter | keyIndex [in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 byte vector for other encryption modes. |
| dataIn [in] | The original data |
| dataOut[out] | Encryption data |
| Return | 0: Success  <0: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg: byte[] dataIn = new byte[16];  dataOut length should be 16  3.hen keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1024  **4.This interface is not supported on TOSS device** | |

#### 3.6.9 Decrypt data (Based on the DUKPT)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataDecryptDukpt(int keyIndex, byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Dukpt Key decrypt data | |
| Parameter | keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 byte vector for other encryption modes. |
| dataIn[in] | Encryption data |
| dataOut[in] | Decryption results |
| Return | 0: Success  <0: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg: byte[] dataIn = new byte[16];  dataOut length should be 16  3.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1024  **4.This interface is not supported on TOSS device** | |

#### 3.6.10 Dukpt's KSN increased by 1

|  |  |  |
| --- | --- | --- |
| Prototype | int dukptIncreaseKSN(int keyIndex) | |
| Feature | Dukpt's KSN increased by 1 | |
| Parameter | keyIndex[in] | Dukpt key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | 0: Success  <0: Fail | |
| Comment | **1.Client APP should pay attention to the return value, if return value <0(failure), Client App shoule repeat call this interface again until it return 0, this can avoid failure in next time access dukpt**  **2.This interface is not supported on TOSS device** | |

#### 3.6.11 Dukpt gets the current KSN

|  |  |  |
| --- | --- | --- |
| Prototype | int dukptCurrentKSN(int keyIndex, byte[] outKSN) | |
| Feature | Dukpt gets the current KSN | |
| Parameter | keyIndex[in] | Dukpt key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| outKSN[out] | buffer, store KSN. Dukpt-3DES KSN is 10 bytes, Dukpt-AES KSN is 12 bytes |
| Return | 0: Success  <0: Fail | |
| Comment | 1.Key for Dukpt key system  **2.This interface is not supported on TOSS device** | |

#### 3.6.12 Get key CheckValue

|  |  |  |
| --- | --- | --- |
| Prototype | int getKeyCheckValue(int keySystem, int keyIndex, byte[] dataOut) | |
| Feature | Get key CheckValue | |
| Parameter | keySystem[in] | Key system. Reference appendix: [KEY\_SYSTEM\_CONSTANTS](#KeySystemTypeConstantDefinition) |
| keyIndex[in] | Key index.  For keySystem SEC\_MKSK, key index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK).  For keySystem SEC\_DUKPT, key index range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| dataOut[out] | Fixed to 4 bytes |
| Return | =0：Success  <0: Fail | |
| Comment | 1.The key index obtained by injecting the BDK is the CheckValue of the BDK corresponding to the IPEK.  **2.This interface is not supported on TOSS device** | |

#### 3.6.13 Get TUSN encrypted data (Only for the Chinese market)

|  |  |  |
| --- | --- | --- |
| Prototype | int getTUSNEncryptData (String dataIn, byte[] dataOut) | |
| Feature | Get TUSN encrypted data | |
| Parameter | dataIn[in] | Used to calculate the scatter value of ciphertext |
| dataOut[out] | Generated ciphertext |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.DataIn encryption random factor (bank card for card number 6, sweep code category for the code after 6).  DataOut is fixed into 4 byte byte[] such as: byte[] dataOut = new byte[4];  **2.This interface is not supported on TOSS device** | |

#### 3.6.14 Generate RSA key pair (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int generateRSAKeys(int pubKeyIndex, int pvtKeyIndex, int keysize, String pubExponent) | |
| Feature | Generate RSA public/private key pair. | |
| Parameter | pubKeyIndex [in] | Public key stored index, range 0-19 |
| pvtKeyIndex [in] | Private key stored index, range 0-19 |
| keysize [in] | The key size,512~65536,Unit: bit, such as 512,1024,etc |
| pubExponent [in] | The public exponent |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.The generated key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [removeRSAKey()](#RemoveRsaKey)  **3.This interface is deprecated, generate RSA key pair please refere to generateRSAKeypair()**  **4.This interface is not supported on TOSS device** | |

#### 3.6.15 Get RSA public key (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int getRSAPublicKey(int pubKeyIndex, byte[] dataOut) | |
| Feature | Get X509 encoded RSA public key data | |
| Parameter | pubKeyIndex [in] | The stored index of public key, range 0-19 |
| dataOut[out] | The out buffer, store public key data, ANS.1 X509 standard format encoding |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is deprecated**  **2.This interface is not supported on TOSS device** | |

#### 3.6.16 Encrypt data with RSA (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataEncryptRSA(String transformation, int keyIndex, byte[] dataIn, byte[] dataOut) | |
| Feature | Encrypt data with RSA | |
| Parameter | transformation [in] | The transformation used in cipher process. Refer to: [RSA transformation definition](#RSATransformationConstantDefinition) |
| keyIndex [in] | The encrypt key index, range 0-19 |
| dataIn [in] | Data will be encrypted |
| dataOut [out] | Out buffer, store encrypted data |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | 1.If transformation is RSA/None/NoPadding or RSA/ECB/NoPadding, dataIn length should equal to module length, dataOut length should greate than or equal to module length. Eg: key size is 2048, and module length is 256B, dataIn length should equal to 256B, and dataOut length should greate than or eaqual to 256B  2. transformation RSA/ECB/OAEPWithSHA-512AndMGF1Padding only usable for 2048 bits RSA key.  **3.This interface is deprecated**  **4.This interface is not supported on TOSS device** | |

#### 3.6.17 Decrypt data with RSA (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataDecryptRSA(String transformation, int keyIndex, byte[] dataIn, byte[] dataOut) | |
| Feature | Decrypt data with RSA | |
| Parameter | transformation [in] | The transformation used in cipher process. Refer to: [RSA transformation definition](#RSATransformationConstantDefinition) |
| keyIndex [in] | The decrypt key index, range 0-19 |
| dataIn [in] | Data will be decrypted |
| dataOut[out] | Out buffer, store decrypted data |
| Return | >=0: The valid length of dataOut  <0: Fail | |
| Comment | 1.If transformation is RSA/None/NoPadding or RSA/ECB/NoPadding, dataIn length should equal to module length, dataOut length should greate than or equal to module length. Eg: key size is 2048, and module length is 256B, dataIn length should equal to 256B, and dataOut length should greate than or eaqual to 256B  2. transformation RSA/ECB/OAEPWithSHA-512AndMGF1Padding only usable for 2048 bits RSA key.  **3.This interface is deprecated**  **4.This interface is not supported on TOSS device** | |

#### 3.6.18 Remove RSA key (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int removeRSAKey (int keyIndex) | |
| Feature | Remove stored RSA public or private key | |
| Parameter | keyIndex [in] | The stored index of RSA key, range 0-19 |
| Return | 0: Success  Other value: Fail | |
| Comment | **1.This interface is deprecated**  **2.This interface is not supported on TOSS device** | |

#### 3.6.19 Store certificate (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int storeCertificate(int certIndex, byte[] certData) | |
| Feature | Store certificate | |
| Parameter | certIndex [in] | The stored index of certificate, range 0-19 |
| certData [in] | The certificate data |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.The stored cert data is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  **2.This interface is deprecated**  **3.This interface is not supported on TOSS device** | |

#### 3.6.20 Get certificate (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int getCertificate(int certIndex, byte[] dataOut) | |
| Feature | Get stored certificate | |
| Parameter | certIndex [in] | The stored index of certificate, 0-19 |
| dataOut [out] | Out buffer, store certificate data |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is deprecated**  **2.This interface is not supported on TOSS device** | |

#### 3.6.21 Get initialized KSN

|  |  |  |
| --- | --- | --- |
| Prototype | int dukptGetInitKSN (byte[] outKSN) | |
| Feature | Get the initialized KSN | |
| Parameter | outKSN [out] | The initialized KSN |
| Return | >=0: The valid data length of outKSN  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.22 RSA algorithm signing data (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int signingRSA(String signAlg, int pvtKeyIndex, byte[] dataIn, byte[] dataOut) | |
| Feature | RSA signing data | |
| Parameter | signAlg [in] | The signing algorithm. Refer to: [RSA signature algoritm definition](#RSASignatureAlgorithmConstantDefinit) |
| pvtKeyIndex [in] | The RSA private key index, range 0-19 |
| dataIn [in] | The data to be signed |
| dataOut [out] | Buffer, store the signature data. |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | 1.If signAlg is NONEWithRSA, default padding mode of private encryption is PKCS1Padding , and the dataIn length should no more than moduleLen-11, and dataOut length should greate than or equal to module length. eg: if keySize is 2048, the moduleLen is 256B, dataIn length shoule less than or equal to 256-11=245B, dataOut length should greate than or equal to 256B  **2.This interface is deprecated**  **3.This interface is not supported on TOSS device** | |

#### 3.6.23 RSA algorithm verify signature (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int verifySignatureRSA(String signAlg, byte[] pubKey, byte[] srcData, byte[] signature) | |
| Feature | RSA verify sinature | |
| Parameter | signAlg [in] | The signing algorithm. Refer to: [RSA signature algoritm definition](#RSASignatureAlgorithmConstantDefinit) |
| pubKey [in] | The RSA public key, ANS.1 X509 DER encoded |
| srcData [in] | The source data before signed (raw data) |
| signature [in] | The signature data. |
| Return | 0: success  <0: Fail | |
| Comment | 1.If signAlg is NONEWithRSA, default padding mode of private encryption is PKCS1Padding , and the srcData length should no more than moduleLen-11, and signature length should be module length. eg: if keySize is 2048, the moduleLen is 256B, srcData length shoule less than or equal to 256-11=245B, signature length should be 256B  **2.This interface is deprecated**  **3.This interface is not supported on TOSS device** | |

#### 3.6.24 Inject plaintext key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectPlaintextKey(String targetPkgName, int keyType, byte[] keyValue, byte[] checkValue, int keyAlgType, int keyIndex) | |
| Feature | Save plaintext key | |
| Parameter | targetPkgName[in] | The target App(App which use the injected key) package name |
| keyType[in] | Key type. Refer to: [Aidl constants Key type constant definition](#KeyTypeConstantDefinition) |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition). |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. Client App should make sure keyIndex is unique for different key  2. The inject key App must has the same signature with the target App(App which use the injected key), or the injected key will not be accessible for target App.  3.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  4.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **5.This interface is not supported on TOSS device** | |

#### 3.6.25 Inject ciphertext key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectCiphertextKey(String targetPkgName,int keyType,byte[] keyValue,byte[] checkValue, int encryptIndex, int keyAlgType, int keyIndex) | |
| Feature | Save ciphertext key | |
| Parameter | targetPkgName[in] | The target App(App which use the injected key) package name |
| keyType[in] | Key Type: Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition). KEY\_TYPE\_KEK Type cannot be saved in ciphertext |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| encryptIndex[in] | Used to decrypt the key ciphertext index, note that here is the index of TMK，range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition). |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. Client App should make sure keyIndex is unique for different key  2. The inject key App must has the same signature with the target App(App which use the injected key), or the injected key will not be accessible for target App.  3.The App which call this interface should have the same signature with targetApp(which packge name is targetPkgName), or targetApp couldn’t access ininjected key  4. App shoule injected a plaintext key to targetApp by calling injectPlaintextKey() before call this interface, or SDK couldn’t not find depend key for this inject key.  5.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  6.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **7.This interface is not supported on TOSS device** | |

#### 3.6.26 Encrypt data (Based on the DUKPT, extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataEncryptDukptEx(int keySelect, int keyIndex, byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Dukpt key encrypt data | |
| Parameter | keySelect[in] | The Dukpt key select, refer to [Aidl constant dukpt key select](#DukptKeySelectConstantDefinition) |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 bytes vector for other encryption modes. |
| dataIn [in] | The original data |
| dataOut[out] | Encryption data |
| Return | 0: Success  <0: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg：byte[] dataIn = new byte[16];  dataOut length should be 16  3.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1024  **4.This interface is not supported on TOSS device** | |

#### 3.6.27 Decrypt data (Based on the DUKPT, extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataDecryptDukptEx(int keySelect, int keyIndex, byte[] dataIn, int encryptionMode, byte[] iv, byte[] dataOut) | |
| Feature | Dukpt Key decrypt data | |
| Parameter | keySelect[in] | The Dukpt key select, refer to [Aidl constant dukpt key select](#DukptKeySelectConstantDefinition) |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| encryptionMode[in] | Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) |
| iv[in] | The initial vector, the encrypted mode is the ECB space, and the 8 byte vector for other encryption modes. |
| dataIn[in] | Encryption data |
| dataOut[in] | Decryption results |
| Return | 0: Success  <0: Fail | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg: byte[] dataIn = new byte[16];  dataOut length should be 16  3.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1024  **4.This interface is not supported on TOSS device** | |

#### 3.6.28 Calculate mac (Based on the DUKPT, extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMacDukptEx(int keySelect, int keyIndex, int macType, byte[] dataIn, byte[] dataOut) | |
| Feature | Dukpt key calculate mac. | |
| Parameter | keySelect[in] | The Dukpt key select, refer to [Aidl constant dukpt key select](#DukptKeySelectConstantDefinition) |
| keyIndex [in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| macType [in] | Mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition) |
| dataIn [in] | The original data |
| dataOut[out] | Mac result. |
| Return | 0: Success  <0 Fail | |
| Comment | 1.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1016  **2.This interface is not supported on TOSS device** | |

#### 3.6.29 Verify mac (Based on the DUKPT, extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int verifyMacDukptEx(int keySelect, int keyIndex, int macType, byte[] dataIn, byte[] macData) | |
| Feature | Dukpt key verify mac | |
| Parameter | keySelect[in] | The Dukpt key select, refer to [Aidl constant dukpt key select](#DukptKeySelectConstantDefinition) |
| keyIndex [in] | Key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| macType [in] | Mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition) |
| dataIn [in] | The original data |
| macData [out] | Mac data |
| Return | 0: Success  <0 Fail | |
| Comment | 1.When keyIndex in 1100-1199(dukpt-extension key), the length of dataIn shoule <=1016  **2.This interface is not supported on TOSS device** | |

#### 3.6.30 Save TR31 key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveTR31Key(byte[] keyValue, int kbpkIndex, int keyIndex) | |
| Feature | Save a TR31 key | |
| Parameter | keyValue[in] | Key data |
| kbpkIndex[in] | KBPK index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK), [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | 0: Success  <0: Fail | |
| Comment | 1.Before save any TR31 key, KBPK should be saved firstly.  2.Currently, this interface only support save TR31 dukpt-IPEK, not support save TR31 dukpt-BDK  3.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  4.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **5.This interface is not supported on TOSS device** | |

#### 3.6.31 Save ciphertext key with decryption key is RSA private key (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int saveCiphertextKeyRSA(int keyType, byte[] keyValue, byte[] checkValue, int keyAlgType, int keyIndex, int encryptIndexRSA, String transformation) | |
| Feature | Save ciphertext key with the decryption key is RSA private key | |
| Parameter | keyType[in] | Key Type: Reference Appendix: [Aidl Constant key type Definition.](#KeyTypeConstantDefinition) KEY\_TYPE\_KEK Type cannot be saved in ciphertext |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant key algorithm type](#KeyAlgoritmTypeConstantDefinition). |
| keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| encryptIndexRSA[in] | RSA private key index, range 0-19 |
| Transformation[in] | RSA transformation, refer to appendix: [RSA transformation definition](#RSATransformationConstantDefinition) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is deprecated**  **4.This interface is not supported on TOSS device** | |

#### 3.6.32 Save RSA key (Deprecated)

|  |  |  |
| --- | --- | --- |
| Prototype | int saveRSAKey(int keyType, byte[] keyValue, int keyIndex) | |
| Feature | Save RSA key | |
| Parameter | keyType[in] | Key Type, 0-RSA public key, 1-RSA private key |
| keyValue[in] | Key data, keyType=0- ANS.1 X509 standard format encoding, keyType=1- ANS.1 PKCS#8 standard format encoding |
| keyIndex[in] | Key index, range 0-19 |
| Return | 0: Success  <0: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [removeRSAKey()](#RemoveRsaKey)  **3.This interface is deprecated**  **4.This interface is not supported on TOSS device** | |

#### 3.6.33 Delete key

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteKey(int keySystem, int keyIndex) | |
| Feature | Delete a stored key | |
| Parameter | keySystem[in] | Key system, refer to appendix: [Aidl constant class key system constants](#KeySystemTypeConstantDefinition) |
| keyIndex[in] | Key index, range is:  when keySystem is SEC\_MKSK the index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  when keySystem is SEC\_DUKPT, the index range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT)  when keySystem is SEC\_RSA\_KEY, the index range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| Return | 0: Success  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.34 Save dukpt-AES key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveKeyDukptAES(int dukptKeyType, int keyType, byte[] keyValue, byte[] checkValue, byte[] ksn, int encryptType, int keyIndex) | |
| Feature | Inject dukpt key | |
| Parameter | dukptKeyType[in] | Dukpt key type, refer to [Aidl constants DukptKeyType](#DukptKeyTpeConstantDefinition) |
| keyType[in] | Key type Contain:  BDK- base dispersed key: corresponding to Aidl constant class key type constant KEY\_TYPE\_DUPKT\_BDK.  IPEK- initial PIN encryption key: corresponding Aidl constant class key type constant KEY\_TYPE\_DUPKT\_IPEK. |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| ksn[in] | KSN(12 bytes, first 8 bytes is valid) |
| encryptType[in] | Key algorithm type, refer to: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition) |
| keyIndex[in] | Key index range: refer to [key system and key index range.SEC\_DUKPT.AES](#KeySystemAndKeyIndexRange_SEC_DUKPT_AES) |
| Return | 0: Success  <0: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.35 Calculate MAC(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMacEx (int keyIndex, int keyLen, int macAlgType, byte[] diversify, byte[] dataIn, byte[] dataOut) | |
| Feature | Calculate MAC | |
| Parameter | keyIndex[in] | Mac key index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| keyLen[in] | The key length, 3DES key allow use first 8/16/24 bytes to calculate Mac  0-the whole key, N- first N(8/16/24) bytes |
| macAlgType[in] | Mac encryption algorithm, refer to: Aidl constant.[MAC algorithm type](#MACAlgorithmConstantDefinition) |
| Diversify[in] | Diversify factor. Before calculate Mac, Mac key operate with diversify factor to generate new Mac key, and use the new key to calculate Mac. Length of diversify is 8B/16B |
| dataIn[in] | The original data |
| dataOut[out] | Mac result |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.Mac length depends on Mac key type, 3DES key return 8B Mac, SM4 return 16B Mac.  **2.This interface is not supported on TOSS device** | |

#### 3.6.36 Generate SM2 key pair

|  |  |  |
| --- | --- | --- |
| Prototype | int generateSM2Keypair(int pvkIndex, Bundle pubKey) | |
| Feature | Generate SM2 public/private key pair | |
| Parameter | pvkIndex[in] | The private key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| pubKey[out] | Contain following keys：  data: byte[] (64B), public key data  kcv: byte[] (5B), public key kcv  rfu: byte[] (10B), RFU data |
| Return | 0: Success  <0: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.37 Inject SM2 key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectSM2Key(int keyIndex, Bundle keyData) | |
| Feature | Inject a SM2 to HSM | |
| Parameter | keyIndex[in] | Key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| keyData[in] | Contain following keys：  data: byte[] (public key 64B/private key 32B), public/private key data  kcv: byte[] (5B), public/private key kcv  rfu: byte[] (10B), public/private key RFU data |
| Return | 0: Success  <0: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.38 SM2 signing data

|  |  |  |
| --- | --- | --- |
| Prototype | int sm2Sign(int pukIndex, int pvkIndex, byte[] userId, byte[] dataIn, byte[] dataOut) | |
| Feature | SM2 algorithm sign data | |
| Parameter | pukIndex[in] | The public key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| pvkIndex[in] | The private key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| userId[in] | The signer Id, length<=512B, recommend value is  0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38 |
| dataIn[in] | The data to be signed, no more than 1024B |
| dataOut[out] | Buffer, store the signature data. |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.39 SM2 verify signature

|  |  |  |
| --- | --- | --- |
| Prototype | int sm2VerifySign(int pukIndex, byte[] userId, byte[] srcData, byte[] signData) | |
| Feature | SM2 algorithm verify signature | |
| Parameter | pukIndex[in] | The public key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| userId[in] | The signer Id, length<=512B |
| srcData[in] | The source data before signed (raw data), length<=1024B |
| signData[in] | The signature data, length is 64B |
| Return | 0: Success  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.40 SM2 encrypt data

|  |  |  |
| --- | --- | --- |
| Prototype | int sm2EncryptData(int pukIndex, byte[] dataIn, byte[] dataOut) | |
| Feature | SM2 encrypt data | |
| Parameter | pukIndex[in] | The public key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| dataIn[in] | The data to be encrypted, 0B<length<=1024B |
| dataOut[out] | Buffer, store encrypted data |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.41 SM2 decrypt data

|  |  |  |
| --- | --- | --- |
| Prototype | int sm2DecryptData(int pvkIndex, byte[] dataIn, byte[] dataOut) | |
| Feature | SM2 decrypt data | |
|  | pvkIndex[in] | The private key index, range: refer to [key system and key index range.SEC\_SM2\_KEY](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| Parameter | dataIn[in] | The data to be decrypted, 96B<length<=1120B |
| dataOut[out] | Buffer, store decrypted data |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.42 Calculate hash

|  |  |  |
| --- | --- | --- |
| Prototype | int calcSecHash(int mode, byte[] dataIn, byte[] dataOut) | |
| Feature | Calculate hash | |
| Parameter | mode [in] | The hash type, refer to [HASH\_TYPE](#Hash_Type) |
| dataIn[in] | The data to be hashed, there’s no limit for data length |
| dataOut[out] | Buffer, store hash data |
| Return | >=0: The valid data length of dataOut  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.43 Verify Mac

|  |  |  |
| --- | --- | --- |
| Prototype | int verifyMac(int keyIndex, int macAlgType, byte[] dataIn, byte[] mac) | |
| Feature | Calculate hash | |
| Parameter | keyIndex [in] | The mac key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| macAlgType [in] | Mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition) |
| dataIn [in] | Raw input data |
| mac[in] | Mac data |
| Return | 0: Success  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.44 Generate RSA key pair(Only support 1024/2048 bit key)

|  |  |  |
| --- | --- | --- |
| Prototype | int generateRSAKeypair(int pvkIndex, int keySize, String pubExponent, byte[] dataOut) | |
| Feature | Generate RSA key pair(only support 1024/2048 bit key) | |
| Parameter | pvkIndex[in] | Private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| keySize[in] | Key size, 1024 or 2048 |
| pubExponent [in] | Public key exponent(HEX,03 or 010001) |
| dataOut [out] | Buffer, store output public key module |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.45 Inject RSA key(Only support 1024/2048 bit key)

|  |  |  |
| --- | --- | --- |
| Prototype | int injectRSAKey(int keyIndex, int keySize, String module, String exponent) | |
| Feature | Inject RSA key(only support 1024/2048 bit key) | |
| Parameter | keyIndex[in] | Public/Private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| keySize[in] | Key size, 1024 or 2048 |
| module [in] | Module (Hex) |
| exponent [in] | Exponent (Hex) |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.46 RSA public key encrypt/private key decrypt

|  |  |  |
| --- | --- | --- |
| Prototype | int rsaEncryptOrDecryptData(int keyIndex, int padding, byte[] dataIn, byte[] dataOut) | |
| Feature | RSA public key encrypt/private key decrypt | |
| Parameter | keyIndex[in] | Public/Private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| padding[in] | Padding mode, 0-NoPadding, 1-PKCS1Padding, 2-PKCS7Padding |
| dataIn[in] | Input data, less than or equal to 896B |
| dataOut[in] | Buffer, store output data |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | 1. This method do RSA operation with input data and output result to param **dataOut**.  2. If param **keyIndex** pointer to a private key:  (1) if **dataIn** **is** public key encrypted data(ciphertext), **dataOut** will be decrypted data(plaintext) (private decryption)  (2) if **dataIn** **is not** public key encrypted data(ciphertext), **dataOut** will be encrypted data(ciphertext) (private encryption is unusually).  3. If param **keyIndex** pointer to a public key:  (1) if **dataIn** **is** private key encrypted data(ciphertext), **dataOut** will be decrypted data(plaintext) (public key decryption. Normally this used on verify signature situation)  (2) if **dataIn** **is not** private key encrypted data(ciphertext), **dataOut** will be encrypted data(ciphertext)(public encryption)  4. The method could provide RSA operation with input data length less than 2048 bit(less than module length)  5. If input data already padded, param padding should be set as NOTHING\_PADDING  **6.This interface is not supported on TOSS device** | |

#### 3.6.47 Generate symmetric key

|  |  |  |
| --- | --- | --- |
| Prototype | int generateSymKey(int keyIndex, int keyType, int keyAlgType) | |
| Feature | Generate a symmetric key | |
| Parameter | keyIndex[in] | Key index , range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| keyType[in] | Key type. Refer to: [Aidl constants Key type constant definition](#KeyTypeConstantDefinition) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.48 Inject symmetric key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectSymKey(int keyIndex, int keyType, byte[] keyValue, byte[] checkValue, int keyAlgType) | |
| Feature | Inject a symmetric key | |
| Parameter | keyIndex[in] | Key index , range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| keyType[in] | Key type. Refer to: [Aidl constants Key type constant definition](#KeyTypeConstantDefinition) |
| keyValue[in] | Key data |
| checkValue[in] | Key check value. Refer to [Key check value param description](#KcvParamDescription) |
| keyAlgType[in] | For specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition).. |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.Only support inject symmetric plaintext MKSK key  2.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  3.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **4.This interface is not supported on TOSS device** | |

#### 3.6.49 Get key CheckValue(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int getKeyCheckValueEx(Bundle bundle, byte[] dataOut) | |
| Feature | Get key CheckValue | |
| Parameter | bundle[in] | Contains following keys:  keySystem: int, key system, refer to appendix: [Aidl constant class key system constants](#KeySystemTypeConstantDefinition)  keyIndex: int, key index, range is:  when keySystem is SEC\_MKSK the index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  when keySystem is SEC\_DUKPT, the index range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT)  kcvMode: int, kcv mode, refer to appendix: [kcv mode definition](#KcvModeDefinition)  targetAppPkgName: String, target app package name |
| dataOut[in] | Buffer, store kcv(4B) |
| Return | 0: Success  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.50 Delete key(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteKeyEx(Bundle bundle) | |
| Feature | Delete key | |
| Parameter | bundle[in] | Contains following keys:  keySystem: int, key system, efer to appendix: [Aidl constant class key system constants](#KeySystemTypeConstantDefinition)  keyIndex: int, key index, range is:  when keySystem is SEC\_MKSK the index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  when keySystem is SEC\_DUKPT, the index range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT)  when keySystem is SEC\_RSA\_KEY, the index range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY)  targetAppPkgName: String, target app package name |
| dataOut[in] | Buffer, store kcv(4B) |
| Return | 0: Success  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.51 Inject ciphertext key(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int injectCiphertextKeyEx(Bundle bundle) | |
| Feature | Delete key | |
| Parameter | bundle[in] | Contains following keys:  targetAppPkgName: String, target app package name  keyType: int, key type(eg: KEK/TMK/PIK/TDK/MAK/REC)  keyValue: byte[], key value  kcvMode: int, kcv mode, refer to [Aidl constant kcv mode definition](#KcvModeDefinition)  kcvMacType: int, kcv Mac type, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  kcvInData: byte[], the data used for calculate kcv.  checkValue: byte[], key check value, refer to [Key check value param description](#KcvParamDescription)  encryptIndex: int, the decrypt key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyAlgType: int, key algorithm type，1-3Des，2-AES，3-SM4  keyIndex: int, key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyLength：int, key length  dataMode: int, encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], initialize vector |
| dataOut[in] | Buffer, store kcv(4B) |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.52 Inject dukpt key(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int injectKeyDukptEx(Bundle bundle) | |
| Feature | Inject dukpt key | |
| Parameter | bundle[in] | Contains following keys:  targetAppPkgName: String, target app package name  keyValue: byte[], key value  kcvMode: int, kcv mode, refer to [Aidl constant kcv mode definition](#KcvModeDefinition)  kcvMacType: int, kcv Mac type, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  kcvInData: byte[], the data used for calculate kcv.  checkValue: byte[], key check value, Refer to [Key check value param description](#KcvParamDescription)  ksn：byte[], KSN  encryptIndex: int, the decrypt key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyAlgType: int, key algorithm type，1-3Des，2-AES，3-SM4  keyIndex: int, key index, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT)  isEncrypt：bool, is ciphertext key  keyLength：int, key length  dataMode: int, encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], initialize vector |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.53 Save plaintext or ciphertext key(extend method)

|  |  |  |
| --- | --- | --- |
| Prototype | int saveKeyEx(Bundle bundle) | |
| Feature | Save plaintext or ciphertext key | |
| Parameter | bundle[in] | Contains following keys:  keyType: int, key type, refer to [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition). KEY\_TYPE\_KEK Type cannot be saved in ciphertext  keyValue: byte[], key data  checkValue: byte[], key check value, refer to [Key check value param description](#KcvParamDescription)  kcvMode: int, kcv mode, refer to [Aidl constant kcv mode definition](#KcvModeDefinition)  kcvMacType: int, kcv Mac type, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  kcvInData: byte[], the data used for calculate kcv.  encryptIndex: int, Used to decrypt the key ciphertext index, note that here is the index of TMK, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyAlgType: int, for specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition)  keyIndex: int, key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  variantUsage: int, the usage of key variant  keyVariant: byte[], key variant  dataMode: int, encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], initialize vector  isEncrypt：bool, is ciphertext key |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.Param kcvMacType and kcvInData are Brazil-CKD special, not supported on other device types**  **4.This interface is not supported on TOSS device** | |

#### 3.6.54 Calculate MAC Extended

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMacExtended(Bundle bundle, byte[] dataOut) | |
| Feature | Calculate MAC | |
| Parameter | bundle [in] | Contain following keys：  keyIndex: int, mac key index, refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyLength: int，0-whole key，not 0-the first keyLength byte of the key macType: int, mac algorithm，refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  diversify: byte [], diversify dispersion factor (value is null, not supported)  dataIn: byt e[], the source data used to perform the mac calculation  iv: byte[], initialize vector |
| dataOut [out] | MAC result |
| Return | 0: Success  <0: Error code | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.55 Calculate Mac Dukpt extend

|  |  |  |
| --- | --- | --- |
| Prototype | int calcMacDukptExtended(Bundle bundle, byte[] dataOut) | |
| Feature | Calculate MAC with dukpt key | |
| Parameter | Bundle [in] | Contain following keys：  keySelect: int, dukpt key select, refer to [Aidl constant dukpt key select](#DukptKeySelectConstantDefinition)  keyIndex: int, range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT)  keyLength: int，0-whole key，not 0-the first keyLength byte of the key macType: int, mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  dataIn: byte[],the source data used to perform the mac calculation |
| dataOut [out] | MAC result |
| Return | 0: Success  <0: Error code | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.56 Read RSA key info

|  |  |  |
| --- | --- | --- |
| Prototype | int readRSAKey(int keyIndex, Bundle keyInfo) | |
| Feature | Read RSA key info | |
| Parameter | keyIndex [in] | RSA key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| keyInfo [out] | Output param, contains following keys:  modulus: byte[], RSA key module  exponent: byte[], RSA key exponent |
| Return | 0: Success  <0: Error code | |
| Comment |  | |

#### 3.6.57 Get key length(Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int getKeyLength(int keySystem, int keyIndex) | |
| Feature | Get key length | |
| Parameter | keySystem [in] | Key system, refer to appendix: [Aidl constant class key system constants](#KeySystemTypeConstantDefinition) |
| keyIndex [in] | Key index, range is:  when keySystem is SEC\_MKSK the index range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  when keySystem is SEC\_DUKPT, the index range: refer to [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | >=0: The key length  <0: Error code | |
| Comment | **1.This interface is Brazil-CKD special, not supported on other device types.**  **2.Currently, This interface is not supported on Brazil-CKD device type.** | |

#### 3.6.58 Write variable key(Brazil-CKD special)

|  |  |  |
| --- | --- | --- |
| Prototype | int writeKeyVariable(Bundle bundle) | |
| Feature | Save a variable key | |
| Parameter | bundle[in] | Input param, contains following keys:  keyType: int, key type, refer to [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition).  checkValue: byte[], key check value, refer to [Key check value param description](#KcvParamDescription)  kcvMode: int, kcv mode, refer to [Aidl constant kcv mode definition](#KcvModeDefinition)  kcvMacType：kcv Mac algorithm, refer to: [Aidl constant.MAC algorithm type](#MACAlgorithmConstantDefinition)  kcvInData: byte[], the data used for calculate kcv.  keyAlgType: int, for specifying the type of key that is currently saved, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition)  srcKeyIndex: int, the source key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  destKeyIndex：int, destination key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  xorData: byte[], the data used for XOR operation |
| Return | 0: Success  <0: Error code | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is Brazil-CKD special, not supported on other device types.**  **4.Currently, This interface is not supported on Brazil-CKD device type.** | |

#### 3.6.59 Key IO control

|  |  |  |
| --- | --- | --- |
| Prototype | int secKeyIoControl(int keyIndex, int ctrCode, byte[] dataIn, byte[] dataOut) | |
| Feature | Key IO control | |
| Parameter | keyIndex[in] | The key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| ctrCode[in] | The key control code, refer to [Key control code](#KeyControlCode) |
| dataIn[in] | The input data |
| dataOut[out] | The output data |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.60 Calculate APACS Mac

|  |  |  |
| --- | --- | --- |
| Prototype | int apacsMac(int initMakIndex, int makIndex, int pikIndex, int ctrCode, byte[] dataIn, byte[] dataOut) | |
| Feature | Calculate APACS Mac | |
| Parameter | initMakIndex[in] | The initialize Mac key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| makIndex[in] | The derived Mac keyindex, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| pikIndex[in] | The derived PIN key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| ctrCode[in] | The key control code, refer to [Key control code](#KeyControlCode) |
| dataIn[in] | The input data |
| dataOut[out] | The output data |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.61 Save ciphertext key, depend key is RSA private key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveCiphertextKeyUnderRSA(int keyIndex, int rsaKeyIndex, int keyType, int keyAlgType, byte[] checkValue, byte[] keyData) | |
| Feature | Save MKSK ciphertext key, the depend key is RAS private key | |
| Parameter | keyIndex[in] | KeyIndex, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| rsaKeyIndex [in] | RSA private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| keyType [in] | Key type, refer to [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition) |
| keyAlgType [in] | Key algorithm type, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition) |
| checkValue [in] | key check value, refer to [Key check value param description](#KcvParamDescription) |
| keyData [in] | key data |
| Return | 0: Success  <0: Error code | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.62 Inject ciphertext key, depend key is RSA private key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectCiphertextKeyUnderRSA(String targetPkgName, int keyIndex, int rsaKeyIndex, int keyType, int keyAlgType, byte[] checkValue, byte[] keyData) | |
| Feature | Inject MKSK ciphertext key, depend key is RSA private key | |
| Parameter | targetPkgName[in] | Target APP package name |
| keyIndex[in] | KeyIndex, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK) |
| rsaKeyIndex [in] | RSA private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY) |
| keyType [in] | Key type, refer to [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition) |
| keyAlgType [in] | Key algorithm type, refer to appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition) |
| checkValue [in] | Key check value, refer to [Key check value param description](#KcvParamDescription) |
| keyData [out] | key data |
| Return | 0: Success  <0: Error code | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.63 Generate symmetric key(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int generateSymKeyEx(Bundle bundle) | |
| Feature | Generate symmetric key | |
| Parameter | bundle[in] | Contains following keys:  keyIndex: int, key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyType: int, key Type: Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition). KEY\_TYPE\_KEK Type cannot be saved in ciphertext keyValue:  keyAlgType: int, for specifying the type of key that is currently saved, refer to Appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition)  keyLength: int, key length(3DES-16/24 bytes, AES-16/24/32 bytes) |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.64 Inject symmetric key(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | int injectSymKeyEx(Bundle bundle) | |
| Feature | Inject a symmetric key | |
| Parameter | bundle[in] | Contains following keys:  keyIndex: int, key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyType: int, key Type: Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition). KEY\_TYPE\_KEK Type cannot be saved in ciphertext keyValue: keyValue: byte[], key data  checkValue: byte[], key check value, refer to [Key check value param description](#KcvParamDescription)  keyAlgType: int, for specifying the type of key that is currently saved, refer to Appendix: [Aidl constant Key algorithm type](#KeyAlgoritmTypeConstantDefinition)  encryptIndex1: int, the decrypt key index1, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK), [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY), [key system and key index range. SEC\_DEVICE\_CERT](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY)  encryptIndex2: int, the decrypt key index2(GOWF algorithm has 2 decrypt keys), range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  dataMode: int, encryption mode, refer to Appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], initialize vector  injectMode: int, refer to Appendix: [Aidl constant Inject symmetric key mode constant](#InjectSymmetricKeyMode), as follows:  (1) If value is INJECT\_OAEP\_MODE(0x05), then **keyValue** is the ciphertext value encrypted by a device certificate's public key. When encrypting, the padding mode is PKCS1\_OAEP\_PADDING (0x04), and the depend key **encryptIndex1** points to a device certificate's private key corresponding to the device certificate's public key  (2) If value is INJECT\_PKCS1\_MODE(0x06), then **keyValue** is a ciphertext value encrypted by a RSA public key. When encrypting, the padding mode is PKCS1\_PADDING(0x01), and the depend key **encryptIndex1** points to an RSA private key corresponding to the RSA public key  (3) If value is other than INJECT\_OAEP\_MODE or INJECT\_PKCS1\_MODE, the saved key is a plaintext key or a key derived from specified algorithm, refer to the instructions in the interface comment column |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.Support inject following type key:  (1) Plaintext key  (2) Ciphertext key(the injected key is encrypted by a asymmetric key), decrypt mode refer to keyInfo fields(only support RSA encrypt MKSK)  (3) 0x80 OWF2 algorithm derive and save key  (4) 0x81 OWF3 algorithm derive and save key  (5) 0x82 GOWF algorithm derive and save key  (6) 0x83 ENC algorithm derive and save key  2.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  3.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **4.This interface is not supported on TOSS device** | |

#### 3.6.65 Inject device cert ciphertext private key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectDeviceCertPrivateKey(Bundle bundle) | |
| Feature | Inject device cert ciphertext private key | |
| Parameter | bundle[in] | Input param, contains following keys:  targetAppPkgName: String, target APP package name  certIndex: int, cert index, range: 9001~9008  mode: int, mode, 4-ECB mode(used for inject ciphertext private key)  isEncrypt: boolean, is ciphertext key  encryptIndex: int, the depend key for ciphertext private key, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  certData: byte[], device cert data  pvkData: byte[], ciphertext private key data |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.This interface only support inject ciphertext device cert private key, that is, param key **isEncrypt** shoule be true, or inject will fail  2.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  **3.This interface is not supported on TOSS device** | |

#### 3.6.66 Generate RSA key pair(extended method , only support 1024/2048 bit key)

|  |  |  |
| --- | --- | --- |
| Prototype | int generateRSAKeypairEx(Bundle bundle, byte[] dataOut) | |
| Feature | Generate RSA key pair(only support 1024/2048 bit key) | |
| Parameter | bundle[in] | Contains following keys:  keyType: int, key Type, value is 0 or KEY\_TYPE\_RSA\_KPK or KEY\_TYPE\_RSA\_KEK, Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition).  pvkIndex: int, private key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY)  keySize: int, key size, 1024 or 2048  pubExponent: String, public key exponent(HEX,03 or 010001) |
| dataOut[out] | Buffer, store output public key module |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.Currently, this interface is only supported on X30TR device type**  **4.This interface is not supported on TOSS device** | |

#### 3.6.67 Inject RSA key(extended method ,only support 1024/2048 bit key)

|  |  |  |
| --- | --- | --- |
| Prototype | int injectRSAKeyEx(Bundle bundle) | |
| Feature | Inject RSA key(only support 1024/2048 bit key) | |
| Parameter | bundle[in] | Contains following keys:  keyType: int, key Type, value is 0 or KEY\_TYPE\_RSA\_KPK or KEY\_TYPE\_RSA\_KEK, Reference Appendix: [Aidl Constant Key Type Definition](#KeyTypeConstantDefinition).  keyIndex: int, private key/public key index, range: refer to [key system and key index range.SEC\_RSA\_KEY](#KeySystemAndKeyIndexRange_SEC_RSA_KEY)  keySize: int, key size, 1024 or 2048  module: String, module(Hex)  exponent: String, key exponent(Hex,03 or 010001) |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.Currently, this interface is only supported on X30TR device type**  **4.This interface is not supported on TOSS device** | |

#### 3.6.68 Save device cert (TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | int setDeviceCertificate(int certIndex, byte[] certData) | |
| Feature | Save device certificate | |
| Parameter | certIndex [in] | Device cert index, range: refer to [key system and key index range.SEC\_DEVICE\_KEY](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY) |
| certData [in] | Cert data |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  **2.This interface is TOSS special, not supported on other device types.** | |

#### 3.6.69 Save base key

|  |  |  |
| --- | --- | --- |
| Prototype | int saveBaseKey(int destinationIndex, byte[] keyData) | |
| Feature | Save base key | |
| Parameter | destinationIndex [in] | Base key index, range: 1~200 |
| keyData[in] | Key data, length is 512B |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **3.This interface is not supported on TOSS device** | |

#### 3.6.70 Encrypt data (extend method)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataEncryptEx(Bundle bundle, byte[] dataOut) | |
| Feature | Encrypt data | |
| Parameter | bundle[in] | Input param, contains following keys:  keyIndex: int, Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT)  keyLength: int, key length, 0-the whole key, >0-the first **keyLength** bytes of key, <0-invalid value  dataIn: byte[], the input data  encryptionMode: int, Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], The initial vector, if **encryptionMode** ECB, iv set as null, otherwise iv should be 8 or 16 bytes. |
| dataOut [out] | The ciphertext data |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg: byte[] dataIn = new byte[16];  dataOut length should be 16  3.Param **keyLength** only applies to TDES key, not to other keys(eg: SM4, AES,etc)  **4.This interface is not supported on TOSS device** | |

#### 3.6.71 Decrypt data(extend method)

|  |  |  |
| --- | --- | --- |
| Prototype | int dataDecryptEx(Bundle bundle, byte[] dataOut) | |
| Feature | Decrypt data | |
| Parameter | bundle[in] | Input param, contains following keys:  keyIndex: int, Key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_DUKPT)  keyLength: int, key length, 0-the whole key, >0-the first **keyLength** bytes of key, <0-invalid value  dataIn: byte[], the input data  encryptionMode: int, Encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition)  iv: byte[], The initial vector, if **encryptionMode** ECB, iv set as null, otherwise iv should be 8 or 16 bytes. |
| dataOut [out] | The plaintext data |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.If encryptionMode is DATA\_MODE\_OFB or DATA\_MODE\_CFB, dataIn length could not be multiply of 8 or 16, otherwise dataIn length should be multiply of 8 or 16  2.The dataOut length should be equal to dataIn length.  eg: byte[] dataIn = new byte[16];  dataOut length should be 16  3.Param **keyLength** only applies to TDES key, not to other keys(eg: SM4, AES,etc)  **4.This interface is not supported on TOSS device** | |

#### 3.6.72 Query key mapping records(return records of caller or targetApp)

|  |  |  |
| --- | --- | --- |
| Prototype | int queryKeyMappingRecordList(Bundle bundle, List<Bundle> list) | |
| Feature | Query key mapping records in SDK | |
| Parameter | bundle[in] | Input param, contains following keys:  targetPkgName: String, target App package name, this param is optional, meaning is:  (1) If not set this param, SDK query interface caller’s key mapping records  (2) If set this param, if interface caller has the same App signature with target App, return target App’s key mapping records, if interface caller has different App signature with target App, return empty list. |
| list[out] | Output param, each item in list is a key mapping record, item contains following keys:  pkgName: String, the package name of the APP to which the key belongs  signature: String, the package signatuer of the APP to which the key belongs, Hex format  keySystem: String，the key sytem，range: SEC\_MKSK, SEC\_DUKPT, SEC\_RSA\_KEY, SEC\_SM2\_KEY, SEC\_ECC\_KEY, SEC\_CERT, SEC\_DEVICE\_CERT, SEC\_MKSK\_NOLOST, SEC\_RSA\_KEY\_NOLOST, SEC\_ECC\_KEY\_NOLOST, SEC\_CERT\_NOLOST, SEC\_UNKNOWN  keyIndexRaw：int，the raw index of the key, refer to: [Key system and key index range](#KeySystemAndKeyIndexRange)  keyIndexMapped: int, the mapped index of the key (the key index in SP)，refer to: [Key system and key index range](#KeySystemAndKeyIndexRange)  keyType: String, the key type, range: BASE\_KEY, KEK, TMK, PIK, MAK, TDK, REC, DUPKT\_BDK, DUPKT\_IPEK, KBPK, TADK, RSA\_PUK, RSA\_PVK, RSA\_PUK\_KPK, RSA\_PVK\_KPK, SM2\_PUK, SM2\_PVK, ECC\_PUK, ECC\_PVK, RSA\_CERT, DEVICE\_CERT\_PVK, UNKNOWN  keyAlgType: String, the algorithm type of the key，range: ALG\_3DES, ALG\_AES, ALG\_SM4, ALG\_UNKNOWN  checkValue: String, the key check value of the key, Hex format, the kcv mode is KCV\_MODE\_CHK0  injectFlag: String, the injection flag of the key, range：null, injected, occupied  keySize: int, for symmetric key, this is the key length, for asymmetric key, this is the keys’binary length. If SDK query key size failure, this param set as -1  ksn: String, the dukpt key ksn, Hex format. If not a dukpt key, this value is null |
| Return | 0: Success  <0: Error code | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.73 Query key mapping records(return all records in devices)

|  |  |  |
| --- | --- | --- |
| Prototype | int queryKeyMappingRecordListWL(List<Bundle> list) | |
| Feature | Query key mapping records in SDK | |
| Parameter | list[out] | Output param, each item in list is a key mapping record, item contains following keys:  pkgName: String, the package name of the APP to which the key belongs  signature: String, the package signatuer of the APP to which the key belongs, Hex format  keySystem: String，the key sytem，range: SEC\_MKSK, SEC\_DUKPT, SEC\_RSA\_KEY, SEC\_SM2\_KEY, SEC\_ECC\_KEY, SEC\_CERT, SEC\_DEVICE\_CERT, SEC\_MKSK\_NOLOST, SEC\_RSA\_KEY\_NOLOST, SEC\_ECC\_KEY\_NOLOST, SEC\_CERT\_NOLOST, SEC\_UNKNOWN  keyIndexRaw：int，the raw index of the key, refer to: [Key system and key index range](#KeySystemAndKeyIndexRange)  keyIndexMapped: int, the mapped index of the key (the key index in SP)，refer to: [Key system and key index range](#KeySystemAndKeyIndexRange)  keyType: String, the key type, range: BASE\_KEY, KEK, TMK, PIK, MAK, TDK, REC, DUPKT\_BDK, DUPKT\_IPEK, KBPK, TADK, RSA\_PUK, RSA\_PVK, RSA\_PUK\_KPK, RSA\_PVK\_KPK, SM2\_PUK, SM2\_PVK, ECC\_PUK, ECC\_PVK, RSA\_CERT, DEVICE\_CERT\_PVK, UNKNOWN  keyAlgType: String, the algorithm type of the key，range: ALG\_3DES, ALG\_AES, ALG\_SM4, ALG\_UNKNOWN  checkValue: String, the key check value of the key, Hex format, the kcv mode is KCV\_MODE\_CHK0  injectFlag: String, the injection flag of the key, range：null, injected, occupied  keySize: int, for symmetric key, this is the key length, for asymmetric key, this is the keys’binary length. If SDK query key size failure, this param set as -1  ksn: String, the dukpt key ksn, Hex format. If not a dukpt key, this value is null |
| Return | 0: Success  <0: Error code | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.74 Read SM2 public key

|  |  |  |
| --- | --- | --- |
| Prototype | int readSM2Key(int keyIndex, Bundle keyInfo) | |
| Feature | Read SM2 public key data | |
| Parameter | keyIndex[in] | SM2 public key index, range: refer to [key system and key index range.SEC\_SM2](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| keyInfo[in] | Output param, contains following keys:  keyData: byte[], SM2 public key data, length is 64B |
| Return | 0: Success  <0: Fail | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.75 Calculate SM3 hash with Z(ID)

|  |  |  |
| --- | --- | --- |
| Prototype | int calcSM3HashWithID(int keyIndex, byte[] userId, byte[] dataIn, byte[] dataOut) | |
| Feature | Calculate SM3 hash with Z ID | |
| Parameter | keyIndex[in] | SM2 public key index, range: refer to [key system and key index range.SEC\_SM2](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| userId[in] | ID, length no more than 32B |
| dataIn[in] | Input data, length no more than 896B |
| dataOut[out] | Buffer, store hash data(32B) |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.76 Calculate SM2 signature with Z(ID) SM3 hash

|  |  |  |
| --- | --- | --- |
| Prototype | int sm2SingleSign(int keyIndex, byte[] hash, byte[] dataOut) | |
| Feature | Calculate SM3 hash with Z ID | |
| Parameter | keyIndex[in] | SM2 private key index, range: refer to [key system and key index range.SEC\_SM2](#KeySystemAndKeyIndexRange_SEC_SM2_KEY) |
| hash[in] | Z(ID) SM3 hash(the returned hash value of calcSM3HashWithID()) , length is 32B |
| dataOut[out] | Buffer, store SM2 signature data(64B) |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | **1.This interface is not supported on TOSS device** | |

#### 3.6.77 Inject TR31 key

|  |  |  |
| --- | --- | --- |
| Prototype | int injectTR31Key(Bundle bundle) | |
| Feature | Inject a TR31 key to targetApp | |
| Parameter | bundle[in] | Input param, contains following keys:  targetPkgName: String, the target App(App which use the injected key) package name  keyValue: byte[], the TR31 key value  kbpkIndex: int, the KBPK index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK)  keyIndex: int, the key index, range: refer to [key system and key index range.SEC\_MKSK](#KeySystemAndKeyIndexRange_SEC_MKSK), [key system and key index range.SEC\_DUKPT](#KeySystemAndKeyIndexRange_SEC_DUKPT) |
| Return | 0: Success  <0: Fail | |
| Comment | 1.Before save any TR31 key, KBPK should be saved firstly.  2.Currently, this interface only support save TR31 dukpt-IPEK, not support save TR31 dukpt-BDK  3.The stored key is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  4.Client App could delete the stored key by calling interface [deleteKey()](#DeleteKey) or [deleteKeyEx()](#DeleteKey_Extended)  **5.This interface is not supported on TOSS device** | |

### 3.7 EMV operation module

#### EMV methods

##### 3.7.1.1 Add AID parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int addAid(AidV2 aid) | |
| Feature | Add or update one AID | |
| Parameter | aid [in] | Refer to [AidV2](#AidV2) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. The SDK has some built-in AID parameters by default, refer to [EMV default AIDS](#_7.1.1_EMV_default) | |

##### 3.7.1.2 Delete AID parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteAid(String tag9F06Value) | |
| Feature | Delete one AID according to the value of tag 9F06，if you want to delete all AIDs，set tag9F06Value as null. | |
| Parameter | tag9F06Value [in] | The value of tag 9F06 (hex format). |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. The SDK has some built-in AID parameters by default, refer to [EMV default AIDS](#_7.1.1_EMV_default) | |

##### 3.7.1.3 Add CAPK parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int addCapk(CapkV2 capk) | |
| Feature | Add or update one CAPK | |
| Parameter | capk [in] | Refer to [CapkV2](#CapkV2) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. The SDK has some built-in CAPK parameters by default, refer to [EMV default CAPKs](#_7.1.1_EMV_default_1) | |

##### 3.7.1.4 Delete CAPK parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteCapk(String tag9F06Value, String tag9F22Value) | |
| Feature | Delete one CAPK according to the value of tag 9F06 and value of tag 9F22, if you want to delete all CAPKs, set tag9F06Value as null. | |
| Parameter | tag9F06Value [in] | The value of tag 9F06 (hex format) |
| tag9F22Value [in] | The value of tag 9F22 (hex format) |
| Return | 0: Success  Other value: Fail | |
| Comment | 1. The SDK has some built-in CAPK parameters by default, refer to [EMV default CAPKs](#_7.1.1_EMV_default_1) | |

##### 3.7.1.5 Set terminal parameter

|  |  |  |
| --- | --- | --- |
| Prototype | int setTerminalParam(EmvTermParamV2 termParam) | |
| Feature | Set terminal parameter. | |
| Parameter | termParam [in] | Reference to [EMVTermParamV2](#EmvTermParamV2) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.6 Check whether AID and CAPK exist or not

|  |  |
| --- | --- |
| Prototype | int isExistCapkAndAid() |
| Feature | Determine the existence of CAPK and AID. |
| Parameter | None |
| Return | -1: Both AID and CAPK not exist.  0: Both AID and CAPK exist.  1: Only exist AID  2: Only exist CAPK |
| Comment | None |

##### 3.7.1.7 Initialize EMV process

|  |  |
| --- | --- |
| Prototype | int initEmvProcess() |
| Feature | Initialize EMV process |
| Parameter | None |
| Return | 0: Success  Other value: Fail |
| Comment | None |

##### 3.7.1.8 Strat transaction process

|  |  |  |
| --- | --- | --- |
| Prototype | void transactProcess(EMVTransDataV2 transData, EMVListenerV2 listener) | |
| Feature | Start EMV process | |
| Parameter | transData [in] | Refer to [EMVTransDataV2](#EMVTransDataV2) |
| listener [in] | Refer to [EMVListenerV2](#EMVListenerV2callbackmethods) |
| Return | None | |
| Comment | None | |

##### 3.7.1.9 Read one TLV data from kernel

|  |  |  |
| --- | --- | --- |
| Prototype | int getTlv(int opCode,String tag, byte[] outData) | |
| Feature | Read one TLV data from kernel | |
| Parameter | opCode[in] | TLV operation code, refer to [TLV\_OPERATITON\_TYPE](#EMV_TLV_operation_type) |
| tag [in] | The tag to read (hex format), eg:”95” |
| outData[out] | Buffer, store the read data. |
| Return | >=0: The valid data length of outData  Other value: Fail | |
| Comment | 1. If opened sred, this interface will not return account sensitive tlv data, refer to [SERD description](#SREDDescription) | |

##### 3.7.1.10 Read TLV data list from kernel

|  |  |  |
| --- | --- | --- |
| Prototype | int getTlvList(int opCode,String[] tags, byte[] outData) | |
| Feature | Read TLV data list from kernel | |
| Parameter | opCode[in] | TLV operation code, refer to [TLV\_OPERATITON\_TYPE](#EMV_TLV_operation_type) |
| tags [in] | The tag list to read (hex format), eg:{“95”,”9F2A”} |
| outData [out] | Buffer , store the read data. |
| Return | >=0: The valid data length of outData  Other value: Fail | |
| Comment | 1.If there is no data has been read for a special tag, outData do not contains this tag, eg: tags={“95”,”9F2A”}, if no data has been read for tag“95”, outData may like as “9F2A03A00001”,which don not contains tag “95”.  2. If opened sred, this interface will not return account sensitive tlv data, refer to [SERD description](#SREDDescription) | |

##### 3.7.1.11 Set TLV data to kernel

|  |  |  |
| --- | --- | --- |
| Prototype | void setTlv(int opCode,String tag, String hexValue) | |
| Feature | Set TLV data to kernel | |
| Parameter | opCode[in] | TLV operation code, refer to [TLV\_OPERATITON\_TYPE](#EMV_TLV_operation_type) |
| tag [in] | The tag to set (hex format) |
| hexValue [in] | The value corresponding to this tag (hex format) |
| Return | None | |
| Comment | None | |

##### 3.7.1.12 Set TLV data list to kernel

|  |  |  |
| --- | --- | --- |
| Prototype | void setTlvList(int opCode,String[] tags, String[] hexValues) | |
| Feature | Set TLV data to kernel | |
| Parameter | opCode[in] | TLV operation code, refer to [TLV\_OPERATITON\_TYPE](#EMV_TLV_operation_type) |
| tags [in] | The tag list to set (hex format), eg:{“9F1A”,”9F33”} |
| hexValues [in] | The values corresponding to this tags list (hex format) eg:{“0156”, “E0F8C8”} |
| Return | None | |
| Comment | None | |

##### 3.7.1.13 Import App select result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importAppSelect (int selectIndex) | |
| Feature | Import app select result to EMV procedure | |
| Parameter | selectIndex [in] | The selected app index, start from 0. |
| Return | None | |
| Comment | None | |

##### 3.7.1.14 Import app final select result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importAppFinalSelectStatus(int status) | |
| Feature | Import app final select result to EMV procedure. | |
| Parameter | status [in] | Final select result, 0:Success, 1:Fail |
| Return | None | |
| Comment | None | |

##### 3.7.1.15 Import card confirm result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importCardNoStatus(int status) | |
| Feature | Import card confirm result to EMV procedure. | |
| Parameter | status [in] | Confirm result, 0:Success, 1:Fail |
| Return | None | |
| Comment | None | |

##### 3.7.1.16 Import certificate authorize result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importCertStatus(int status) | |
| Feature | Import certificate authorize result to EMV procedure | |
| Parameter | status [in] | Authorize result, 0: Success, 1: Fail |
| Return | None | |
| Comment | None | |

##### 3.7.1.17 Import PIN input result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importPinInputStatus(int pinType, int inputResult) | |
| Feature | Import PIN input result to EMV procedure | |
| Parameter | pinType [in] | PIN type, 0: online PIN, 1: offline PIN |
| inputResult [in] | PIN input results. 0: Success, 1: PIN cancels, 2: PIN skipped, 3: PINPAD fails, 4: input PIN timeout |
| Return | None | |
| Comment | None | |

##### 3.7.1.18 Import online process result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | int importOnlineProcStatus(int status, String[] tags, String[] hexValues, byte[] outData) | |
| Feature | Import online process result to EMV procedure. | |
| Parameter | status [in] | Online result, 0-Online approval, 1-Online denial, 2- Online failed |
| tags[in] | Online data which need to import to kernel,eg: { "71", "72", "91", "8A", "89" } |
| hexValues[in] | Values corresponding to each tag in parameter **tags** |
| outData[out] | Result data of kernel process. |
| Return | >=0: The valid data length in **outData**  <0: Error code | |
| Comment | None | |

##### 3.7.1.19 Import signature result to EMV

|  |  |  |
| --- | --- | --- |
| Prototype | void importSignatureStatus(int status) | |
| Feature | Import signature result to EMV procedure. | |
| Parameter | status [in] | process result, 0:Success, 1:Fail |
| Return | None | |
| Comment | None | |

##### 3.7.1.20 Read transaction log

|  |  |  |
| --- | --- | --- |
| Prototype | int readTransLog(int logType, List<String> infoOut) | |
| Feature | Read card transaction log | |
| Parameter | logType [in] | log type,0:transaction log, 1: trap log |
| infoOut [out] | Buffer, store the read logs. |
| Return | 0: Success  Other value: Fail | |
| Comment |  | |

##### 3.7.1.21 Abort transact process

|  |  |
| --- | --- |
| Prototype | void abortTransactProcess() |
| Feature | Abort the emv transact process |
| Parameter | None |
| Return | None |
| Comment | **Note: If EMV process not started, call this method has no effect, if EMV process already started, call this method can interrupt the process and result in EMVListenerV2.onTransResult() called** |

##### 3.7.1.22 Import data exchange status to emv

|  |  |  |
| --- | --- | --- |
| Prototype | void importDataExchangeStatus(int status) | |
| Feature | Import data exchange status to emv procedure. | |
| Parameter | status[in] | Data exchange result, 0:Success, 1:Fail |
| Return | None | |
| Comment | None | |

##### 3.7.1.23 Start transaction process(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void transactProcessEx(Bundle transData, EMVListenerV2 listener) | |
| Feature | Start EMV process, refer to **transactProcess(EMVTransDataV2 transData, EMVListenerV2 listener)** | |
| Parameter | transData [in] | Emv transaction data, similar to [EMVTransDataV2](#EMVTransDataV2), this param can be set with following code:  Bundle bundle = new Bundle();  bundle.putString("amount", amount); //transaction amount  bundle.putString("transType", transType);//transaction type  bundle.putInt("flowType", flowType);//flow type  bundle.putInt("cardType", cardType);//card type  bundle.putString("cashbackAmount",cashbackAmount);//cashback amount  bundle.putInt("emvAuthLevel",level);//auth level,0-Normal,1-EC PBOC |
| listener[in] | Refer to [EMVListenerV2](#EMVListenerV2callbackmethods) |
| Return | None | |
| Comment | None | |

##### 3.7.1.24 Query electronic cash balance

|  |  |  |
| --- | --- | --- |
| Prototype | int queryECBalance(Bundle bundle) | |
| Feature | Query electronic cash balance | |
| Parameter | bundle[out] | Contains the following data:  9F51: String, Application currency code (Hex format)  9F79: long, Electronic cash balance |
| Return | None | |
| Comment | None | |

##### 3.7.1.25 Add DRL LimitSet

|  |  |  |
| --- | --- | --- |
| Prototype | int addDrlLimitSet(DrlV2 drl) | |
| Feature | Query electronic cash balance | |
| Parameter | drl[in] | DRL LimitSet, refer to [DrlV2](#DrlV2) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.26 Delete DRL LimitSet

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteDrlLimitSet(String programId) | |
| Feature | Delete one DRL LimitSet by programId, if want to delete all DRL LimitSets, set programId as null | |
| Parameter | programId [in] | The program ID |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.27 Set terminal param(extended method)

|  |  |  |
| --- | --- | --- |
| Prototype | void setTermParamEx(Bundle bundle) | |
| Feature | Set terminal param | |
| Parameter | bundle[out] | Contains the following data:  supportDRL: boolean, is support DRl function  downloadAidParam: boolean, is support download AidParam  downloadAidParamAll：boolean，is download AidParamAll  downloadPreParamEP：boolean，is download PreParamEP  optOnlineRes：boolean，is optimize Online Result  ledLightingDuration：int，Led Lighting Duration  contactlessManualSelApp：boolean, Contactless transaction select App manually  contactlessManualSelAppGeneral：boolean, Contactless transaction select App manually(Generally version)  contactlessManualSelAppGeneralEx: boolean, Contactless transaction select App manually(Generally version) extend  supportEP: boolean，Is supportEP  importScriptData：boolean，Online denial import script data  quickChip: booean, should follow the quick chip process  noSignatureOrPINThreshold: int, the threshold for PIN input in quick chip mode, if amount excess this value, input PIN is needed  dpasV2Support：boolean, support discoverV2.0  dpasDeferredAuthSupport：boolean, support discover Deferred Auth  dpasDataStorageSupport：boolean，support discover DataStorage  dpasExtendedLoggingSupport：boolean，support discover ExtendedLogging  dpasTearingRecoverySupport：boolean，support discover TearingRecovery  dpasContactlessSpeedupSupport: boolean, is support DPAS contactless speedup preocess  jcbContactlessSpeedupSupport: boolean, is support JCB contactless speedup preocess  AEContactlessSpeedupSupport: boolean, is support AE contactless speedup preocess  AEOnlineProcessSupport: boolean, is support AE kernel execute the process after online process  supportPOI: Boolean, set the switch of EP POI  CertifiedEP: Boolean, set whether run Certified EP process or not  AutoRun: Boolean, set the switch of Auto run  KernelsForCertEP: long, set which kernel types support Certified EP  SupportAE4.1: boolean, is support AE 4.1 specification |
| Return | None | |
| Comment | None | |

##### 3.7.1.28 Query All Aid or Capk list

|  |  |  |
| --- | --- | --- |
| Prototype | int queryAidCapkList(int type, List<String> list) | |
| Feature | Query all Aid or Capk list in SDK | |
| Parameter | type[in] | The query type, 0-query all Aids 1-query all capks |
| list[out] | The queried Aid or Capk list |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.29 EMV transact pre process

|  |  |
| --- | --- |
| Prototype | int transactPreProcess() |
| Feature | EMV pre process |
| Parameter | None |
| Return | 0: Success  Other value: Fail |
| Comment | None |

##### 3.7.1.30 Add RevocationList

|  |  |  |
| --- | --- | --- |
| Prototype | int addRevocList(RevocListV2 revocList) | |
| Feature | Add or update a RevocationList | |
| Parameter | revocList[in] | A revocationList, refer to Aidl constant. [RevocListV2](#RevocListV2) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.31 delete RevocationList

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteRevocList(RevocListV2 revocList) | |
| Feature | Delete one or all RevocationLists | |
| Parameter | revocList[in] | The RevocationList to be delete, refer to Aidl constant. [RevocListV2](#RevocListV2).  If set this param as null, all RevocationLists will be deleted |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.32 Set system time

|  |  |  |
| --- | --- | --- |
| Prototype | int sysSetTime(long timeStamp) | |
| Feature | Set system time | |
| Parameter | timeStamp[in] | The timestamp to be set, unit: ms |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.33 Get system time

|  |  |  |
| --- | --- | --- |
| Prototype | int sysGetTime(byte[] outData) | |
| Feature | Get system time, unit: s | |
| Parameter | outData[out] | Buffer, store the got system time, this value is a yyyyMMddHHmmss  string converted byte array, eg: hexStringToBytes(“20191130142020”) |
| Return | >=0: The valid data length in **outData**  <0: Fail | |
| Comment | None | |

##### 3.7.1.34 Clear data

|  |  |  |
| --- | --- | --- |
| Prototype | int clearData(int opCode) | |
| Feature | Clear emv data | |
| Parameter | opCode[in] | Operation code, 0-clear all data, 1-clear terminal data, 2-clear card data.  Refer to Aidl contact. [EMV clear data constant definition](#EMVCleanDataConstantDefinition) |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.35 Set account data security param

|  |  |  |
| --- | --- | --- |
| Prototype | int setAccountDataSecParam(Bundle bundle) | |
| Feature | Set account data secure param | |
| Parameter | bundle[in] | encKeySystem: int, key system, refer to [KEY\_SYSTEM\_CONSTANTS](#KeySystemTypeConstantDefinition). Only support SEC\_MKSK, SEC\_DUKPT, SEC\_RSA  encKeyIndex: int**,** the encryption key index, refer to [key system and key index range](#KeySystemAndKeyIndexRange)  encKeyAlgType：int, the algorithm of encryption key. refer to [Key algorithm type constant](#KeyAlgoritmTypeConstantDefinition). Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encMode: int, the encryption mode, refer to appendix: [Aidl constant encryption mode constant.](#EncryptionModeConstantDefinition) Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encIv: byte[], IV(initial vector), if **encMode** is ECB，value is **null**, otherwise value is 8 or 16 bytes data. Only valid when **encKeySystem** is SEC\_MKSK, SEC\_DUKPT  encPaddingMode: byte, The padding mode on encryption. If **encKeySystem** is SEC\_MKSK or SEC\_DUKPT, and encryption key is DES/AES/SM4, and track data length is not a multiply of 8 or 16, then padding the track data at tail with **encPaddingMode** to a multiple of 8 or 16 in length, if **encKeySystem** is SEC\_RSA, padding mode refer to [RSA Padding Mode](#RSAPaddingMode), mode NOTHING\_PADDING is unsupported  encMaskStart: int, 0~6, how much characters are clear text in front of PAN  encMaskEnd: int, 0~4, how much characters are clear text in rear of PAN  encMaskWord: char, 0 or non-digit character，The mask character for **EncMaskStart**~**encMaskWord** PAN, default is ‘**\***’  panAppendContent: String, the appended data to track2 before RSA encrypt track data (TID)  panAppendMode: int, the concatenate mode of **panAppendContent(TID)** and track2 before RSA encryption, 0- TID+track2, 1-track2+TID  sred: boolean, the default is true |
| Return | 0: Success  Other value: Fail | |
| Comment | 1.If param bundle not contains key sred or contains key **sred** and corresponding value is true, sred will be turn-on, refer to [SRED description](#_7.9_SRED_description)  2.If pram bundle contains key **sred** and corresponding value is false, **sred** will be turn-off, meanwhile other keys except sred will be ignored | |

##### 3.7.1.36 Get account security data

|  |  |  |
| --- | --- | --- |
| Prototype | int getAccountSecData(int opCode, String[] tags, Bundle bundle) | |
| Feature | Get account security data | |
| Parameter | opCode[in] | TLV operation code, refer to [TLV\_OPERATITON\_TYPE](#EMV_TLV_operation_type) |
| tags[in] | The tag list to read (hex format), eg: {“95”,”9F2A”} |
| bundle[out] | Contains the following key:  Tag+Enc: String, eg: {“5AEnc, “57Enc”}, tag encryption data  panMask: String, PAN masking data, only exist for tag 5A/57  If the encryption key is RSA key, output data contains following key:  appendedPanEncBytes: byte[], the ciphertext data of TID+track2 or track2+TID  appendedPanEnc: String, the ciphertext data of TID+track2 or track2+TID, Hex format |
| Return | 0: Success  Other value: Fail | |
| Comment | None | |

##### 3.7.1.37 Import terminal risk management result

|  |  |  |
| --- | --- | --- |
| Prototype | void importTermRiskManagementStatus(int status) | |
| Feature | Import terminal risk management result | |
| Parameter | status[in] | Terminal risk management result, 0-success, other value-failed |
| Return | None | |
| Comment | None | |

##### 3.7.1.38 Import pre first GAC result

|  |  |  |
| --- | --- | --- |
| Prototype | void importPreFirstGenACStatus(int status) | |
| Feature | Import pre first GAC result | |
| Parameter | status[in] | Pre first GAC result, 0-success, other value-failed |
| Return | None | |
| Comment | None | |

##### 3.7.1.39 Import DataStorage

|  |  |  |
| --- | --- | --- |
| Prototype | void importDataStorage(String[] tags, String[] hexValues) | |
| Feature | Import DataStorage | |
| Parameter | tags[in] | DataStorage tag list |
| hexValues[in] | DataStorage value list |
| Return | None | |
| Comment | None | |

##### 3.7.1.40 add EMV data listener

|  |  |  |
| --- | --- | --- |
| Prototype | void addEMVDataListener(EMVDataListenerV2 listener) | |
| Feature | Add EMV data listener | |
| Parameter | listener [in] | EMV data listener |
| Return | None | |
| Comment | None | |

##### 3.7.1.41 Add DET data

|  |  |  |
| --- | --- | --- |
| Prototype | int addDETData(byte[] data) | |
| Feature | Add DET data | |
| Parameter | data [in] | DET data |
| Return | 0: Success  Other value: Fail | |
| Comment |  | |

##### 3.7.1.42 Data input/output process

|  |  |  |
| --- | --- | --- |
| Prototype | int dataInputOutputProcess(int mode, int procType, byte[] inData, byte[] outData) | |
| Feature | Data input/out process | |
| Parameter | mode[in] | Input/output mode, range: 0-input, 1-output |
| procType [in] | Process type |
| inData[in] | Input data |
| outData[out] | Output data |
| Return | >=0: The valid data length in **outData**  <0: Fail | |
| Comment | 1.This interface only used for SDK internal test, Client App should not call this interface | |

##### 3.7.1.43 Import PIN input result for TOSS(TOSS special)

|  |  |  |
| --- | --- | --- |
| Prototype | void importPinInputStatusForToss(byte[] pinValue, int inputResult) | |
| Feature | Import PIN input result on TOSS device | |
| Parameter | pinValue [in] | PIN value |
| inputResult [in] | PIN input results. 0: Success, 1: PIN cancels, 2: PIN skipped, 3: PINPAD fails, 4: input PIN timeout |
| Return | None | |
| Comment | **1.This interface is TOSS special, not supported on other device types.** | |

#### 3.7.2 EMVListenerV2 callback methods

**Note: This interface is the interface passed between AIDLs and must be implemented in accordance with the Aidl interface when the callback is passed.**

##### 3.7.2.1 Request App select

|  |  |  |
| --- | --- | --- |
| Prototype | void onWaitAppSelect(List<EMVCandidateV2> candList, boolean isFirstSelect) | |
| Feature | If card has more than one app, request to select one app. Refer to [EMVCandidateV2](#EMVCandidateV2) | |
| Parameter | candList [in] | The app list for selecting |
| isFirstSelect[in] | Is first time to select |
| Return | None | |
| Comment |  | |

##### 3.7.2.2 Request app final select

|  |  |  |
| --- | --- | --- |
| Prototype | void onAppFinalSelect (String tag9F06Value) | |
| Feature | Notify client the emv app final select | |
| Parameter | tag9F06Value [in] | The value of tag 9F06 |
| Return | None | |
| Comment | None | |

##### 3.7.2.3 Request Confirm Card number

|  |  |  |
| --- | --- | --- |
| Prototype | void onConfirmCardNo(String cardNO) | |
| Feature | Request confirm card number. | |
| Parameter | cardNO[in] | The card number to be confirmed |
| Return | None | |
| Comment | 1. If opened sred, param **cardNO** is masked data,refer to [SERD description](#SREDDescription) | |

##### 3.7.2.4 Request confirm certificate info

|  |  |  |
| --- | --- | --- |
| Prototype | void onCertVerfiy(int certType, String certInfo) | |
| Feature | Request confirm certificate info if EMV process required | |
| Parameter | certType[in] | Certificate type |
| certInfo[in] | Certificate info |
| Return | None | |
| Comment | None | |

##### 3.7.2.5 Request input PIN

|  |  |  |
| --- | --- | --- |
| Prototype | void onRequestShowPinPad(int pinType, int remainTimes) | |
| Feature | Request input PIN | |
| Parameter | pinType[in] | PIN type, 0:online PIN，1: offline PIN |
| remainTimes[in] | The remain try times of offline PIN. If current is online PIN, this value is always -1, if it is the first time to try input PIN, this value is -1 too. |
| Return | None | |
| Comment | None | |

##### 3.7.2.6 Requset signature

|  |  |
| --- | --- |
| Prototype | void onRequestSignature() |
| Feature | Request user signature |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.7 Requset online process

|  |  |
| --- | --- |
| Prototype | void onOnlineProc() |
| Feature | Request online process |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.8 Card data exchange complete

|  |  |
| --- | --- |
| Prototype | void onCardDataExchangeComplete() |
| Feature | The kernel and card data exchange complete |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.9 EMV procedure result

|  |  |  |
| --- | --- | --- |
| Prototype | void onTransResult(int code, String desc) | |
| Feature | Send EMV procedure result to client. This method is mutex with onConfirmationCodeVerified(), that is, in an EMV procedure, only one of the 2 methods will be called. | |
| Parameter | code [in] | The result code:  1-Offline approval, 2-Offline decline, 3-reserved, 4- try again, 5-online approval, 6-online decline  <0-Error code. |
| desc[in] | The result message corresponding to result code. |
| Return | None | |
| Comment | None | |

##### 3.7.2.10 Confirmation code verified

|  |  |
| --- | --- |
| Prototype | void onConfirmationCodeVerified() |
| Feature | The confirmation code has been verified. This method is mutex with onTransResult(), that is, in an EMV procedure, only one of the 2 methods will be called.  **At this moment, only PayPass Contactless may call this method.** |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.11 Request data exchage

|  |  |  |
| --- | --- | --- |
| Prototype | void onRequestDataExchange(String cardNo) | |
| Feature | Request user exchange data with emv procedure(MIR use this method) | |
| Parameter | cardNo[in] | The card number |
| Return | None | |
| Comment | None | |

##### 3.7.2.12 Request terminal risk management

|  |  |
| --- | --- |
| Prototype | void onTermRiskManagement() |
| Feature | Request terminal risk management |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.13 Request pre first GAC

|  |  |
| --- | --- |
| Prototype | void onPreFirstGenAC() |
| Feature | Request pre first GAC |
| Parameter | None |
| Return | None |
| Comment | None |

##### 3.7.2.13 Request DataStorage Process

|  |  |  |
| --- | --- | --- |
| Prototype | void onDataStorageProc(String[] containerID, String[] containerContent) | |
| Feature | Request data Storage Process | |
| Parameter | containerID[in] | Container id |
| Parameter | containerContent[in] | Container Content |
| Return | None | |
| Comment | None | |

### 3.8 Tax operation module (Unsupported for TOSS)

#### 3.8.1 Tax data exchange

|  |  |  |
| --- | --- | --- |
| Prototype | int taxDataExchange(byte[] taxSend, byte[] taxRecv) | |
| Feature | Exchange tax data | |
| Parameter | taxSend[in] | Read/write operation send data |
| taxRecv[out] | Read/write operation receive data |
| Return | None | |
| Comment | Packet format for read: Header(2B)+CMD(1B)+Expect receive data length(2B,LSB)  Packet format for write: Header(2B)+CMD(1B)+Data field len(2B,LSB)+Data field(LEN B)  Receive packet format for read/write: Header(2B)+Response code(1B)+Data field len(2B,LSB)+Data field(LEN B)  The max response data length of read/write is 1030B, following is an example of data exchange (Hex format):  Read-send>>：1B 1D 09 00 06  Read-receive<<：1B 1D 00 00 06 04 01 00 02 EE DB  Write-send>>：1B 1D 08 00 0B 04 06 00 11 11 03 02 0A 15 39 18  Write-receive<<：1B 1D 00 00 06 04 01 00 02 EE DB | |

### 3.9 Device certificate manager module (Unsupported for TOSS)

#### 3.9.1 Get device certificate and private key state

|  |  |  |
| --- | --- | --- |
| Prototype | int getDevKeyState(int certIndex) | |
| Feature | Get device certificate and private key state | |
| Parameter | certIndex [in] | Certificate index, range: refer to [key system and key index range. SEC\_DEVICE\_CERT](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY) |
| Return | 0: Success  < 0: Error code. | |
| Comment | None | |

#### 3.9.2 Generate device certificate public and private key

|  |  |  |
| --- | --- | --- |
| Prototype | int genDevKey(int certIndex, int mode, byte[] dataOut) | |
| Feature | Generate device certificate public and private key | |
| Parameter | certIndex [in] | Certificate index, range: refer to [key system and key index range. SEC\_DEVICE\_CERT](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY) |
| mode[in] | Mode, refer to: [Aidl constant.](#MACAlgorithmConstantDefinition) Generate [device](#GenerateDeviceCertificateModeDefinition) certificate mode |
| dataOut[out] | Buffer, store public key module |
| Return | >=0: The valid data length of dataOut  < 0: Error code. | |
| Comment | 1.The stored cert is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteDevCertAndPrivateKey) | |

#### 3.9.3 Save device certificate

|  |  |  |
| --- | --- | --- |
| Prototype | int saveDevCert(int certIndex, byte[] certData) | |
| Feature | Save device certificate | |
| Parameter | certIndex [in] | Certificate index, range: refer to [key system and key index range. SEC\_DEVICE\_CERT](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY) |
| certData [in] | Certificate data, length<=1920 |
| Return | 0: Success  < 0: Error code. | |
| Comment | 1.The stored cert is persistent for permanently, and will no lost while restart App or device reboot or device shutdown.  2.Client App could delete the stored key by calling interface [deleteKey()](#DeleteDevCertAndPrivateKey) | |

#### 3.9.4 Delete device certificate and private key

|  |  |  |
| --- | --- | --- |
| Prototype | int deleteKey(int certIndex) | |
| Feature | Delete device certificate and private key | |
| Parameter | certIndex [in] | Certificate index, range: refer to key system and key index range. SEC\_DEVICE\_CERT |
| Return | 0: Success  < 0: Error code. | |
| Comment | None | |

#### 3.9.5 Get device cert with speicified package name

|  |  |  |
| --- | --- | --- |
| Prototype | int getDeviceCertificateEx(Bundle bundle, byte[] dataOut) | |
| Feature | Get device cert with speicified package name | |
| Parameter | certIndex [in] | Input param, contains following keys:  targetPkgName: String, target App package nam, not null  certIndex: int, cert index, range: refer to [key system and key index range. SEC\_DEVICE\_CERT](#KeySystemAndKeyIndexRange_SEC_DEVICE_KEY) |
|  | dataOut[out] | Buffer, store device cert data |
| Return | >=0: The valid data length of dataOut  < 0: Error code | |
| Comment | None | |

### 3.10 HCE manager module (Unsupported for TOSS)

#### 3.10.1 Open HCE

|  |  |  |
| --- | --- | --- |
| Prototype | int hceOpen(int cardType) | |
| Feature | Open HCE | |
| Parameter | cardType[in] | Card type: 2-NFC tag2 card, 4-NFC FORUM T4T card |
| Return | 0: Success  < 0: Error code | |
| Comment | None | |

#### 3.10.2 Open HCE (overload method)

|  |  |  |
| --- | --- | --- |
| Prototype | int hceOpen(int cardType, byte[] param) | |
| Feature | Open HCE | |
| Parameter | cardType[in] | Card type: 2-NFC tag2 card, 4-NFC FORUM T4T card |
| param[in] | Card type relevant param, length<=255B, can be null |
| Return | 0: Success  < 0: Error code | |
| Comment | None | |

#### 3.10.3 HCE write NDEF data

|  |  |  |
| --- | --- | --- |
| Prototype | int hceNdefWrite(NdefMessage msg) | |
| Feature | HCE write NDEF data | |
| Parameter | msg[in] | The NDEF data, currently only support NdefMessage format data. For tag2, msg.length<=399B, and for t4t , msg.length<=1024B |
| Return | 0: Success  < 0: Error code | |
| Comment | None | |

#### 3.10.4 HCE write data

|  |  |  |
| --- | --- | --- |
| Prototype | int hceWrite(byte[] msg) | |
| Feature | HCE write data | |
| Parameter | msg[in] | The data to be written, currently only support NdefMessage format data. For tag2, msg.length<=399B, and for t4t , msg.length<=1024B |
| Return | 0: Success  < 0: Error code | |
| Comment | None | |

#### 3.10.5 HCE read NDEF data

|  |  |  |
| --- | --- | --- |
| Prototype | NdefMessage hceNdefRead() | |
| Feature | HCE read NDEF data | |
| Parameter | [in] |  |
| Return | Success: NdefMessage object  Failure: null | |
| Comment | None | |

#### 3.10.6 HCE read data

|  |  |  |
| --- | --- | --- |
| Prototype | int hceRead(byte[] outData) | |
| Feature | HCE read data | |
| Parameter | outData[out] | Buffer, store read data, length>=1024B, currently only return NdefMessage format data |
| Return | >=0: The valid data length of outData  < 0: Error code | |
| Comment | None | |

#### 3.10.7 Close HCE

|  |  |  |
| --- | --- | --- |
| Prototype | int hceClose() | |
| Feature | Close HCE and deactive emulated card | |
| Parameter | [in] |  |
| Return | 0: Success  < 0: Error code | |
| Comment | None | |

## 4. Error Code Definition

|  |  |
| --- | --- |
| Error code | Error description |
| -100 | Incorrect number of parameters or length |
| -101 | Unsupported command |
| -1000 | System parameter error |
| -1001 | Feature not supported |
| -1002 | Initialization failed |
| -1003 | System time year error |
| -1004 | System time month error |
| -1005 | System time day error |
| -1006 | System time hour error |
| -1007 | System time minute error |
| -1008 | System time second error |
| -1009 | Hardware failure |
| -1010 | Buffer length error |
| -2000 | Card parameter error |
| -2001 | Have no card |
| -2002 | Multiple cards |
| -2032 | Mifare card refuse CMD |
| -2033 | Mifare card response data size not expected number |
| -2034 | Mifare card not authenticate password |
| -2035 | Mifare card authenticate failed |
| -2036 | Mifare card response data error |
| -2037 | Mifare card param illegal |
| -2038 | Mifare Plus calculate CMAC error |
| -2039 | Mifare Plus CMAC error |
| -2040 | Mifare Plus AES decrypt failed |
| -2041 | Mifare Plus AES encrypt failed |
| -2401 | HCE param error |
| -2402 | HCE module init failed |
| -2403 | HCE module not open |
| -2404 | HCE function not supported |
| -2405 | HCE t4t operation failed |
| -2406 | HCE NDEF data unchanged |
| -2407 | HCE status error |
| -2408 | HCE t2t operation failed |
| -2100 | Magnetic card data decoding |
| -2500 | Module detection failed |
| -2501 | Drive core data structure error |
| -2502 | Module is not powered |
| -2503 | Carrier is not turned on |
| -2520 | Communication timeout |
| -2521 | Internal FIFO operation failed |
| -2522 | Communication frame error |
| -2523 | Communication character check error |
| -2524 | Communication conflict |
| -2525 | Signal in communication does not comply with the protocol |
| -2526 | CRC check error in communication |
| -2527 | M1 card password authentication error |
| -2528 | Mifare authentication parameters are incorrect |
| -2529 | Card exists |
| -2530 | The card refused command |
| -2540 | The number of data in the A card communication response does not match the expected |
| -2541 | A card communication replies to the first character of the WUPA/REQA command is illegal |
| -2542 | Card number checksum error of A card communication response |
| -2543 | The first character of the card number answered by the A card communication is wrong. |
| -2544 | A card communication response ATS TL byte is illegal |
| -2545 | The A0 T0 byte of the A card communication response is illegal. |
| -2546 | The A1 TA1 byte of the ATS communication is illegal. |
| -2547 | The ATB TB1 byte of the A-card communication response is illegal. |
| -2548 | A card communication response ATS TC1 byte is illegal |
| -2549 | Type–A not support ISO14443–4, activation process aborted |
| -2550 | The number of data in the B card communication response does not match the expected |
| -2551 | The first character of the B card communication response WUBP/REQB command is not 0x50 |
| -2552 | The fourth bit of the protocol type byte in ATQB is not ‘0’ |
| -2553 | The B-card communication responds to the difference in channel coding and setting in the ATTRIB command. |
| -2554 | B card communication response HLTB command response non 0x00 error |
| -2560 | Retransmitted to the limit if the correct reception is received |
| -2561 | Block type coding error |
| -2562 | I block PCB error or subsequent data length error |
| -2563 | PICC uses I block response link block |
| -2564 | Received I block serial number is incorrect |
| -2565 | R block PCB error or subsequent data length error |
| -2566 | PICC response NAK block |
| -2567 | The received R block serial number is incorrect. |
| -2568 | S block PCB error or subsequent data length error |
| -2569 | S block non-S-WTX request sent by PICC |
| -2570 | WTX parameter error requested (=0) |
| -2571 | Card return data exceeds FSD |
| -2580 | Read ID card GUID error |
| -2581 | User canceled |
| -2582 | MSR or IC interrupted |
| -2800 | Verify error |
| -2801 | Communication timeout |
| -2802 | Module is not powered |
| -2803 | ATR error |
| -2804 | Communication error |
| -2805 | PPS error |
| -2806 | T0 param error |
| -2807 | T0 probe byte error |
| -2808 | T1 param error |
| -2809 | T1 LRC error |
| -2810 | T1 block number error |
| -2904 | invalid reset |
| -2909 | verify PSC error |
| -2911 | Nack |
| -3000 | Security parameter error |
| -3001 | Root key error |
| -3002 | Security system is locked |
| -3003 | Security file read and write error |
| -3004 | Key index error |
| -3005 | Key verification error |
| -3006 | No PIN entry |
| -3007 | PIN input canceled |
| -3008 | PIN input timeout |
| -3009 | PIN input interval is too short |
| -3010 | KCV mode error |
| -3011 | KCV check error |
| -3012 | KCV ODD check error |
| -3013 | No matching key |
| -3014 | Error of key type |
| -3015 | Key length error |
| -3016 | Key exponent length error |
| -3017 | Destination key index error |
| -3018 | Source key index error |
| -3019 | Source key type error |
| -3020 | Group index error |
| -3022 | No KCV |
| -3023 | DUKPT overflow |
| -3024 | DUKPT key type error |
| -3025 | DUKPT KSN needs to add 1 |
| -3026 | Try to use the key outside the scope of the key. |
| -3027 | Incorrect use of the key, restricting only decrypted keys to encrypt data, such as calculate the Mac with the master key |
| -3028 | Feature not yet supported |
| -3029 | Feature key attribute does not match |
| -3030 | Not certified |
| -3031 | TR31 key distribute encryption key error |
| -3032 | TR31 key distribute MAC key error |
| -3033 | CMAC algorithm error |
| -3034 | Data length error |
| -3035 | Algorithm block error |
| -3036 | Des algorithm exception |
| -3037 | Aes algorithm exception |
| -3038 | Sm4 algorithm exception |
| -3039 | Sm2 algorithm exception |
| -3040 | Sm3 algorithm exception |
| -3041 | Rsa algorithm exception |
| -3042 | hash algorithm exception |
| -3046 | POS public key exception |
| -3047 | PAN timeout |
| -3048 | Times exceed limit |
| -3049 | Password error |
| -3050 | key invalid |
| -3051 | Not set new password |
| -3052 | Request sensitive service |
| -3061 | PIN/PAN anti-exhausting |
| -3062 | The same key exists |
| -3063 | TLV param error |
| -3064 | ECC algorithm error |
| -3065 | RND key error |
| -3081 | AP key read error |
| -3082 | AP key write error |
| -3083 | AP key verify error |
| -3084 | AP key lost |
| -3085 | AP key open failed |
| -3086 | AP key self-check failed |
| -3087 | AP key write mode not supported |
| -3088 | AP key invalid |
| -3089 | AP key access timeout |
| -3090 | AP key delete file failed |
| -3091 | AP key other error |
| -4000 | Transaction refuse |
| -4001 | Please use another interface |
| -4002 | Transaction termination |
| -4003 | See phone |
| -4005 | Final select data error |
| -4006 | (MIR)Data error |
| -4008 | Use another card |
| -4009 | (JCB)Try again |
| -4100 | Transaction termination (command sending and receiving error) |
| -4101 | Transaction termination (command receiving timeout) |
| -4102 | Transaction termination (command receiving timeout) |
| -4103 | Transaction terminated (status code error) |
| -4104 | Transaction termination (card is locked) |
| -4105 | Transaction terminated (application locked) |
| -4106 | Transaction termination (no application on the terminal) |
| -4107 | Transaction termination (the terminal and the card have no common support application) |
| -4108 | Transaction terminated (card return data error) |
| -4109 | Transaction termination (duplicate data element returned by the card) |
| -4110 | Transaction termination (transaction is not accepted) |
| -4111 | Transaction termination (card expired) |
| -4112 | Preprocessing Parameters list is empty |
| -4113 | Communication timeout in the APDU command |
| -4114 | Transaction terminated (L1 transmission error) |
| -4115 | Error in the APDU command |
| -4116 | Transaction terminated (L2 mandatory data error) |
| -4117 | Transaction terminated (L2 card authentication failed (offline data authentication failed)) |
| -4118 | Transaction terminated (L2 status word error) |
| -4119 | Transaction terminated (L2 data parsing failed) |
| -4120 | Transaction terminated (L2 transaction amount exceeds the contactless transaction limit) |
| -4121 | Transaction terminated (L2 card data error) |
| -4122 | mag not support (Paypass) |
| -4123 | Transaction terminated (L2 card without PPSE) |
| -4124 | Transaction terminated (L2 PPSE processing error) |
| -4125 | Candidate list is empty |
| -4126 | Transaction terminated (L2 IDS read error) |
| -4127 | Transaction terminated (L2 IDS write error) |
| -4128 | Transaction terminated (L2 IDS data error) |
| -4129 | Transaction terminated (L2 IDS has no matching AC) |
| -4130 | Transaction terminated (L2 terminal data error) |
| -4131 | L3 Timeout |
| -4132 | Transaction terminated (L3 cancel) |
| -4133 | Transaction terminated (L3 transaction amount does not exist) |
| -4134 | Transaction terminated (re-presentation of the card) |
| -4135 | Transaction terminated (using other cards (with Data Record)) |
| -4136 | Transaction terminated (using other cards) |
| -4137 | Transaction terminated (GPO response error) |
| -4138 | Transaction terminated (final selection of card data error) |
| -4139 | Transaction terminated (L3 no DET data) |
| -4140 | Kernel not support |
| -4141 | Amount exceeds Reader Contactless Transaction Limit |
| -4142 | Amount is zero |
| -4144 | Insert, Swipe or Try Another Card |
| -4500 | Invalid parameter |
| -4501 | Check code error when downloading public key |
| -4502 | Terminal Parameters data does not exist |
| -4503 | Terminal Parameters data error |
| -4504 | Transaction log does not exist |
| -4505 | Transaction log data error |
| -4506 | EMV data does not exist |
| -4507 | PBOC LOG format does not exist |
| -4825 | Two present card |
| -4854 | Complete command with empty |
| -4855 | Complete command with ODOL |
| -4856 | Complete command after reselect app |
| -4857 | Read record command after reselect app |
| -7001 | Printer error |
| -7002 | Low battery voltage |
| -7003 | Out of paper |
| -7004 | Temperature is too high |
| -7005 | Printer data error |
| -7006 | Invalid print parameters |
| -7007 | Device not open or device operation error |
| -7008 | Print buffer overflow |
| -7009 | Printer not supported |
| -7010 | Printer function not supported |
| -7011 | Printer no platen |
| -8001 | Write data fail for tax control module |
| -8002 | Read data fail for tax control module |
| -10100 | Serial port closed |
| -10101 | Serial port overtime |
| -10102 | LRC check error |
| -10103 | SP Out of sequence |
| -10104 | SP is initializing |
| -10105 | SP is rebooting |
| -10106 | SP is reconnecting. |
| -10107 | SP is busy |
| -10108 | SP is sleep |
| -10200 | Read OS file package error |
| -10201 | SP is updating |
| -10202 | Connect SP failed |
| -10203 | Failed to open upgrade file |
| -10204 | Packet timeout |
| -10205 | Packet processing error |
| -10206 | Upgrade string is too long |
| -10207 | Upgrade unsuccessful |
| -10208 | Did not get the sdk version number of this machine |
| -10209 | The version is the same as the target upgrade version. |
| -10210 | Query default information failed |
| -10211 | Firmware version does not allow downgrade |
| -10212 | Upgrade cancelled |
| -10300 | Input parameter error |
| -10301 | The length of the response packet data area is illegal. |
| -10302 | Answer packet data parsing error |
| -10400 | The kernel has been rebooted |
| -11000 | BASE error start |
| -11001 | Operation not permitted |
| -11002 | No such file or directory |
| -11003 | No such process |
| -11004 | Interrupted system call |
| -11005 | I/O error |
| -11006 | No such device or address |
| -11007 | Argument list too long |
| -11008 | Exec format error |
| -11009 | Bad file number |
| -11010 | No child processes |
| -11011 | Try again |
| -11012 | Out of memory |
| -11013 | Permission denied |
| -11014 | Bad address |
| -11015 | Block device required |
| -11016 | Device or resource busy |
| -11017 | File exists |
| -11018 | Cross-device link |
| -11019 | No such device |
| -11020 | Not a directory |
| -11021 | Is a directory |
| -11022 | Invalid argument |
| -11023 | File table overflow |
| -11024 | Too many open files |
| -11025 | Not a typewriter |
| -11026 | Text file busy |
| -11027 | File too large |
| -11028 | No space left on device |
| -11029 | Read-only file system |
| -11030 | Illegal seek |
| -11031 | Too many links |
| -11032 | Broken pipe |
| -11033 | Math argument out of domain of function |
| -11034 | Math result not representable |
| -11107 | Communication not connected |
| -11301 | ACK response packet param error |
| -11302 | SP ACK data overflow |
| -11401 | CMD packet data length overflow |
| -11402 | CMD package verify error, no info field |
| -11403 | SP receive buffer is full, no info field |
| -11404 | SP receive data timeout, no info field |
| -11406 | CMD packet sequence error |
| -11600 | CMD packet param error |
| -11601 | Unsupported CMD packet |
| -11700 | Firmware update failed |
| -11701 | Firmware size exceed designed |
| -11702 | Firmware signature verify failed |
| -11703 | Firmware boot name error |
| -11704 | Firmware update CMD error |
| -11705 | Firmware update access flash error |
| -11706 | Get device code error |
| -11707 | SE chip type error |
| -16000 | Not enough space for LITESO |
| -16001 | LITESO length error |
| -16002 | Amount of installed LITESO exceed system max value |
| -16003 | LITESO signature error |
| -16004 | LITESO fingerprint error |
| -16005 | LITESO write flash error |
| -16006 | LITESO info error |
| -16007 | LITESO no rights to access file |
| -16008 | LITESO not exist |
| -20000 | Function not supported |
| -20001 | Repeated call |
| -20002 | Firmware is being upgraded |
| -20003 | Parameter error |
| -20004 | Thread was aborted |
| -20005 | Firmware upgrade failed |
| -20006 | Firmware verification failed |
| -30001 | Failure to read card (unknown cause card failure, recommended to re-read the card operation) |
| -30002 | Unknown card type |
| -30003 | Failure of NFC check card |
| -30004 | Failure of IC check card |
| -30005 | Read card timeout |
| -30013 | This card is a chip card and can not fallback. |
| -30014 | Create candidate list timeout |
| -30015 | Card interaction failure |
| -30016 | Wrong card interaction parameters |
| -40002 | Key length error |
| -40003 | The check value error |
| -40004 | Store key fail |
| -40005 | Calculate MAC error |
| -40006 | Encryption error |
| -40007 | Return array data length error |
| -40008 | The MAC algorithm type not support |
| -40009 | Length of check value error |
| -40010 | Key index error |
| -40011 | Decrypt error |
| -40012 | Length of key error |
| -40013 | Get random error |
| -40014 | Key does not exist |
| -40016 | Verify signature fail |
| -40017 | Failed to get alarm information code |
| -40018 | Key partition has run out |
| -40019 | Inject BDK error |
| -40020 | Transformation not supported |
| -40021 | Key not saved |
| -50002 | Transaction preprocess fail |
| -50003 | Transaction process fail |
| -50004 | EMV kernel process fail |
| -50005 | PAN format error |
| -50006 | Call PINPAD fail |
| -50007 | None kernel data |
| -50008 | PINPAD parameter error |
| -50009 | EMV process not finish |
| -50010 | The transaction type not support |
| -50011 | Checking card information fail or timeout. |
| -50012 | CVM error |
| -50013 | Database operation fail |
| -50014 | No matching CAPK |
| -50015 | Save terminal parameter error |
| -50016 | No matching AID |
| -50017 | Check card fail |
| -50018 | Call interface order error |
| -50019 | Transaction data invalid |
| -50020 | PIN entry cancel |
| -50021 | PIN entry error |
| -50022 | The index of app select error |
| -50023 | Cert verify error |
| -50024 | Online process error |
| -50025 | App final select timeout |
| -50026 | App final select error |
| -50027 | Signature error |
| -50028 | Unknown CVM type |
| -50029 | Data exchange error |
| -50030 | Data exchange timeout |
| -50031 | Terminal risk management timeout |
| -50032 | Terminal risk management error |
| -50033 | Pre first GAC called timeout |
| -50034 | Pre first GAC called error |
| -60001 | Input PIN timeout |
| -60002 | Keyboard failed to activate password |
| -60003 | PinPadType type error (when the incoming keyboard type is not 1 and 2, the error is returned) |
| -60004 | Getting PinBlock failed |
| -60005 | PIN status query thread is interrupted |
| -70001 | Miss permisstion **com.sunmi.perm.MSR** |
| -70002 | Miss permission **com.sunmi.perm.ICC** |
| -70003 | Miss permission **com.sunmi.perm. CONTACTLESS\_CARD** |
| -70004 | Miss permission **com.sunmi.perm. PINPAD** |
| -70005 | Miss permission **com.sunmi.perm. SECURITY** |
| -70006 | Miss permission **com.sunmi.perm. LED** |
| -80001 | PinPad ongoing |
| -90001 | No device searched |

## 5. Entity class

### 5.1 EmvTermParamV2 – Terminal parameter entity class

**Class member variable declaration:**

|  |
| --- |
| public String ifDsn = "3030303030393035"; // IFD sn  public String terminalType = "22"; // Terminal type  public String countryCode = "0156"; // Terminal country code  public boolean forceOnline = false; // Merchant force online(1-online)  public boolean getDataPIN = true; // Whether to read the number of retries before the password is detected?  public boolean surportPSESel = true; // Support PSE select  public boolean useTermAIPFlg = true; // Is risk management based on card AIP?  public boolean termAIP = true; // Whether the terminal enforces risk management?  public boolean bypassAllFlg; // After processing a PIN in byPass mode, whether other PINs are also handled in bypass mode?  public boolean bypassPin = true; // Support bypass PIN?  public boolean batchCapture; // Whether to batch capture data?  public boolean ectSiFlg = true; // Does EC Terminal Support Indicator exist?  public boolean ectSiVal = true; //Whether to support the electronic cash terminal support indicator?  public boolean ectTlFlg = true; //Does EC Terminal Transaction Limit exist?  public String ectTlVal = "100000"; // Electronic cash terminal transaction limit, unit cent  public String capability = "E0F8C8"; // Terminal capability  public String addCapability = "0300C00000";// Terminal extended capability  public boolean scriptMode; // scriptMode  public boolean adviceFlag = true; // adviceFlag  public boolean isSupportSM = true; // Support SM?  public boolean isSupportTransLog = true; // Support transaction LOG?  public boolean isSupportMultiLang = true; // Support multiple language?  public boolean isSupportExceptFile = true; // Support exception file?  public boolean isSupportAccountSelect = true; // Support account select?  public String TTQ = "26000080"; // Terminal transaction attribute (For contactless)  public boolean IsReadLogInCard; // Is it an application selection process for reading in-card transaction records?  private byte[] reserved = new byte[3]; // Reserved byte value must be 0 |

### 5.2 AidV2 – AID entity class

|  |
| --- |
| public byte[] aid; //AID  public byte[] cvmLmt = new byte[6]; //ContactLess CVM Limit  public byte[] termClssLmt = new byte[6]; //ContactLess Trans Limit  public byte[] termClssOfflineFloorLmt = new byte[6]; //ContactLess Floor Limit  public byte[] termOfflineFloorLmt = new byte[6]; //Terminal electronic cash transaction limit  public byte selFlag; //ASI  public byte targetPer;// Random target percentage  public byte maxTargetPer; // Offset the maximum target percentage randomly selected  public byte[] floorLimit; //Terminal Floor Limit  public byte randTransSel; // Whether to make random trading choices  public byte velocityCheck;// Whether to perform frequency detection  public byte[] threshold = new byte[4];// Bias randomly selected thresholds  public byte[] TACDenial = new byte[5];// TAC-Denial  public byte[] TACOnline = new byte[5];// TAC-Online  public byte[] TACDefault = new byte[5]; //TAC-Default  public byte[] AcquierId = new byte[6];// Acquirer ID  public byte[] dDOL; //Default DDOL  public byte[] tDOL; // Default TDOL  public byte[] version = new byte[2]; //Application version  public byte rMDLen; // Risk management data length  public byte[] riskManData = new byte[8]; // Risk management data  public byte[] merchName = new byte[128]; //Merchant Name  public byte[] merchCateCode = new byte[2];// Merchant type code  public byte[] merchId = new byte[16]; //Merchant ID  public byte[] termId = new byte[8]; //Teminal ID  public byte[] referCurrCode = {0x01, 0x56}; //Reference currency code  public byte referCurrExp; //Reference currency code exponent  public byte[] referCurrCon = new byte[4]; // Conversion Coefficient of Reference Currency Code and Transaction currency Code(currently, this field is ignored)  public byte clsStatusCheck; //Contactless status check  public byte zeroCheck; //zero amount check  public byte kernelType;//Kernel type(DFC10A)  public byte paramType;//Param type(DFC10B,0-default,1-contact,2-contactless)  public byte[] ttq = new byte[4]; //terminal transaction attribute 9F66  public byte[] kernelID; //kernel ID DFC10C  public byte extSelectSupFlg;//extend select flag DFC10D(00-not support，01support) |

### 5.3 CapkV2 – CAPK entity class

|  |
| --- |
| public byte[] rid = new byte[5];// CAPK RID  public byte index; //CAPK KeyID  public byte hashInd;// CAPK hash algorithm indicator  public byte arithInd; //CAPK RSA algorithm indicator  public byte[] modul; // CAPK modulus  public byte[] exponent; // CAPK Exponent  public byte[] expDate = new byte[3];// CAPK Expiry Date(YYMMDD)  public byte[] checkSum = new byte[20]; // CAPK CheckSum |

### 5.4 EMVTransDataV2 – Transaction data entity class

|  |
| --- |
| public String amount; // Transaction amount(unit: cent)，must have parameter，cannot be null or "" , when amount="0" means check the balance.  public String transType = "00"; // Transaction type default padding with "00".  public int flowType = 01; // flow type，0x01:Standard flow；0x02:Simple flow；0x03 qPass  public int cardType = 2; //Card type, 2:IC 4:NFC |

### 5.5 EMVCandidateV2 – EMV Application Candidate

|  |
| --- |
| public short index;//Index,corresponding to the priority list  public String aid;//Card Aid  public String appPreName;/App prefer name  public String appLabel;//App label  public String issDiscrData;//Data of tag 'BF0C’  public byte priority;//Priority flag  public String appName;//Local app name  public byte kernelType;//The kernel type of contactless App |

### 5.6 PinPadConfigV2 – Pinpad configuration entity class

|  |
| --- |
| private int pinpadType; //PinPad type:  0-SDK built-in normal PinPad(defualt),  1-Client customized normal PinPad,  2-SDK built-in blind PinPad,  3-SDK built-in rnib auth blind PinPad,  4-SDK built-in rnib auth normal PinPad,  5-Client customized blind PinPad  private int pinType = 0; // Pin type, 0-Online PIN，1-Offline PIN  private boolean isOrderNumKey = false; // true: Normal Pinpad; false: Random Pinpad  private byte[] pan; // Ascii format to byte. eg. “123456”.getbytes("us ascii")  private int pinkeyIndex; // Pin Key Index  private int maxInput = 6; // Maximum password input (max 12 numbers)  private int minInput = 0; // Minimum password input  private int timeout = 60000; // Time out/millisecond  private boolean isSupportbypass = true;//support bypasspin?  private int pinblockFormat = 0; //pinblock format, refer to [PinBlock format](#PinBlock_format)  private int algorithmType = 0; //Encrypted Pin algorithm type 0-3DES (returns 8 bytes), 1-SM4 (returns 16 bytes)  private int keySystem = 0; // The key system to which Pik currently belongs is 0-SEC\_MKSK, 1-SEC\_DUKPT. Reference Appendix: Aidl Constant Key System Constant |
| **Note:**  **1. Currently, pinPayType not support type 3 and 4** |

### 5.7 PinPadTextConfigV2 – Pinpad showing text configuration entity class

|  |
| --- |
| public String confirm;//The **confirm** key text  public String inputPin; //The input online PIN text  public String inputOfflinePin;//The input offline PIN text  public String reinputOfflinePinFormat;//The reinput offline PIN(show remain times)text |

### 5.8 PinPadDataV2 – Pinpad layout entity class

|  |
| --- |
| public int numX;//The x-axis coordinate(absolute coordinate) of the first key on the PinPad keyboard  public int numY;//The y-axis coordinate(absolute coordinate) of the first key on the PinPad keyboard  public int numH;//The height value of the first key on the PinPad keyboard  public int numW;//The width value of the first key on the PinPad keyboard  public int lineW;//The divider line width (if no divider line, set as 0)  public int cancelX;//The x-axis coordinate(absolute coordinate) of the cancel key on the PinPad keyboard  public int cancelY;//The y-axis coordinate(absolute coordinate) of the cancel key on the PinPad keyboard  public int cancelH; //The height value of the cancel key on the PinPad keyboard  public int cancelW; //The width value of the cancel key on the PinPad keyboard  public int rows; //The row count of keys  public int clos; //The column count of keys  public byte[] keyMap = new byte[64]; // The mapping of Keyboard’keys and values |

### 5.9 PinPadDataV2Ex – Pinpad layout entity class(extended)

|  |
| --- |
| public int numX;//The x-axis coordinate(absolute coordinate) of the first key on the PinPad keyboard  public int numY;//The y-axis coordinate(absolute coordinate) of the first key on the PinPad keyboard  public int numH;//The height value of the first key on the PinPad keyboard  public int numW;//The width value of the first key on the PinPad keyboard  public int lineW;//The divider line width (if no divider line, set as 0)  public int cancelX;//The x-axis coordinate(absolute coordinate) of the cancel key on the PinPad keyboard  public int cancelY;//The y-axis coordinate(absolute coordinate) of the cancel key on the PinPad keyboard  public int cancelH; //The height value of the cancel key on the PinPad keyboard  public int cancelW; //The width value of the cancel key on the PinPad keyboard  public int enterX; //The x-axis coordinate(absolute coordinate) of the Enter key on the PinPad keyboard  public int enterY; //The y-axis coordinate(absolute coordinate) of the Enter key on the PinPad keyboard  public int enterH; //The height value of the Enter key on the PinPad keyboard  public int enterW; //The width value of the Enter key on the PinPad keyboard  public int clearX; //The x-axis coordinate(absolute coordinate) of the Clear key on the PinPad keyboard  public int clearY; //The y-axis coordinate(absolute coordinate) of the Clear key on the PinPad keyboard  public int clearH; //The height value of the Clear key on the PinPad keyboard  public int clearW; //The width value of the Clear key on the PinPad keyboard  public int rows; //The row count of keys  public int clos; //The column count of keys  public byte[] keyMap = new byte[64]; // The mapping of Keyboard’keys and values |

### 5.9 DrlV2 – DRL LimitSet entity class

|  |
| --- |
| public boolean isDefaultLmt = false; //Is default limitSet  public boolean statusCheck = false; //Is enable status check  public byte zeroCheck = 1; //Is enable zero amount check, 0-online,1-not allow,2-disable  public byte[] programID; //Program ID  public byte[] cvmLmt = new byte[6]; // ContactLess CVM Limit  public byte[] termClssLmt = new byte[6]; // ContactLess Trans Limit  public byte[] termClssFloorLmt = new byte[6];//Terminal Contactless floor limit  public byte[] termFloorLmt = new byte[6];//Terminal floor limit  public boolean cvmLmtActivate = true;//Is enable CVM limit check  public boolean termClssLmtActivate = false; //Is enable terminal contactless limit check  public byte termClssFloorLmtActivate = 1;//Is enable terminal contactless floor limit,0-disable,1-enable,2-enable but contactless floor limit not exist |

### 5.10 RevocListV2 – RevocationList entity class

|  |
| --- |
| public byte[] rid= new byte[5];// Application registration service provider ID  public byte index;//Key index  public byte[] sn= new byte[3];//Serial number  public byte[] reserved = new byte[3];//Reserved bytes, all bytes should be 0 |

## 6. Access permission

### 6.1 Permission location

The appropriate permissions should be declared in the AndroidManifest file before using each interface.

### 6.2 Permission definition

#### 6.2.1 Magnetic stripe card permission

<uses-permission android:name="com.sunmi.perm.MSR"/>

#### 6.2.2 Contact IC card permission

<uses-permission android:name="com.sunmi.perm.ICC"/>

#### 6.2.3 Contactless IC card permission

<uses-permission android:name="com.sunmi.perm.CONTACTLESS\_CARD"/>

#### 6.2.4 Pinpad permission

<uses-permission android:name="com.sunmi.perm.PINPAD"/>

#### 6.2.5 Security permission

<uses-permission android:name="com.sunmi.perm.SECURITY"/>

#### 6.2.6 LED permission

<uses-permission android:name="com.sunmi.perm.LED" />

#### 6.2.7 Printer permission (not supported)

<uses-permission android:name="com.sunmi.perm.PRINTER" />

#### 6.2.8 Serial port permission (not supported)

<uses-permission android:name="com.sunmi.perm.SERIAL"/>

#### 6.2.9 Customer display permission (not supported)

<uses-permission android:name="com.sunmi.perm.CUSTOMER\_DISPLAY"/>

#### 6.2.10 ID card permission (not supported)

<uses-permission android:name="com.sunmi.perm.IDCard"/>

#### 6.2.11 Money box permission (not supported)

<uses-permission android:name="com.sunmi.perm.MONEYBOX"/>

#### 6.2.12 Finger print permission (not supported)

<uses-permission android:name="com.sunmi.perm.FINGERPRINT"/>

## 7. Appendix

### 7.1 Aidl constants class（com.sunmi.pay.hardware.aidl.AidlConstants）

#### 7.1.1 Card type constant definition

|  |
| --- |
| // Magnetic  public static final int MAGNETIC= 1<<0;  // IC  public static final int IC = 1<<1;  // RFC  public static final int NFC = 1<<2;  // Mifare  public static final int MIFARE= 1<<3;  // PSAM,slot 0  public static final int PSAM0= 1<<4;  // Felica  public static final int FELICA= 1<<5;  // SAM1  public static final int SAM1= 1<<6;  // Mifare plus  public static final int MIFARE\_PLUS= 1<<7;  // Mifare desfire  public static final int MIFARE\_DESFIRE= 1<<8;  // AT24C01  public static final int AT24C01= 1<<9;  // AT24C02  public static final int AT24C02= 1<<10;  // AT24C04  public static final int AT24C04= 1<<11;  // AT24C08  public static final int AT24C08= 1<<12;  // AT24C16  public static final int AT24C16= 1<<13;  // AT24C32  public static final int AT24C32= 1<<14;  // AT24C64  public static final int AT24C64= 1<<15;  // AT24C128  public static final int AT24C128= 1<<16;  // AT24C256  public static final int AT24C256= 1<<17;  // AT24C512  public static final int AT24C512= 1<<18;  // SLE4442  public static final int SLE4442= 1<<19;  // SLE4428  public static final int SLE4428= 1<<20;  // AT88SC1608  public static final int AT88SC1608= 1<<21;  // CTX512B  public static final int CTX512B= 1<<22;  // SAM2  public static final int SAM2= 1<<23;  // SAM3  public static final int SAM3= 1<<24;  // SRI  public static final int SRI= 1<<25;  // SAM4  public static final int SAM4= 1<<26;  // SAM5  public static final int SAM5= 1<<27;  // ISO15693  public static final int ISO15693= 1<<28; |

#### 7.1.2 Key type constant definition

|  |
| --- |
| // KEK (Key encrypt key)  public final static int KEY\_TYPE\_KEK = 0x01;  // TMK (Terminal master key)  public final static int KEY\_TYPE\_TMK = 0x02;  // PIK (PIN key)  public final static int KEY\_TYPE\_PIK = 0x03;  // MAK (Mac key)  public final static int KEY\_TYPE\_MAK = 0x04;  // TDK (Track data key)  public final static int KEY\_TYPE\_TDK = 0x05;  // Reserved  public final static int KEY\_TYPE\_REC = 0x06;  // Dupkt BDK (Base derived key)  public static final int KEY\_TYPE\_DUPKT\_BDK = 0x07;  // Dukpt IPEK (Initial PIN encryption key)  public static final int KEY\_TYPE\_DUPKT\_IPEK = 0x08;  // TR31 KBPK (Key block protection key)  public static final int KEY\_TYPE\_KBPK = 0x09;  // Account data key  public static final int KEY\_TYPE\_TADK =0x0A;  // RSA KPK  public static final int KEY\_TYPE\_RSA\_KPK = 0x0B;  // RSA KEK  public static final int KEY\_TYPE\_RSA\_KEK = 0x0C; |

#### 7.1.3 Key algorithm type constant definition

|  |
| --- |
| // Encryption type: 3DES or DES  public final static int KEY\_ALG\_TYPE\_3DES = 0x01;  // Encryption type: AES  public final static int KEY\_ALG\_TYPE\_AES = 0x02;  // Encryption type: SM4  public final static int KEY\_ALG\_TYPE\_SM4 = 0x03; |

#### 7.1.4 MAC algorithm constant definition

|  |
| --- |
| // ISO9797-1-ALG1  public static final int MAC\_ALG\_ISO\_9797\_1\_MAC\_ALG1 = 1001;  // ISO9797-1-ALG3  public static final int MAC\_ALG\_ISO\_9797\_1\_MAC\_ALG3 = 1003;  // Same as ISO9797-1-ALG3  public static final int MAC\_ALG\_ISO\_16609\_MAC\_ALG1 = 2000;  // ECB algorithm.  public static final int MAC\_ALG\_FAST\_MODE = 3000;  // X9\_19 algorithm mac  public static final int MAC\_ALG\_X9\_19 = 3001;  // CBC UnionPay algorithm  public static final int MAC\_ALG\_CBC = 3002;  // SM4 mac  public static final int MAC\_ALG\_CUP\_SM4\_MAC\_ALG1 = 3003;  public static final int MAC\_ALG\_CUP\_SM4\_MAC\_ALG2 = 3004;  public static final int MAC\_ALG\_X9\_19\_DEA = 3005;  // HMAC-SHA1  public static final int MAC\_ALG\_HMAC\_SHA1 = 3006;  // HMAC-SHA256  public static final int MAC\_ALG\_HMAC\_SHA256 = 3007;  // CMAC  public static final int MAC\_ALG\_CMAC = 3008;  // FAST\_MODE internation standard mac  public static final int MAC\_ALG\_FAST\_MODE\_INTERNATIONAL = 30000;  // CBC international standard mac public static final int MAC\_ALG\_CBC\_INTERNATIONAL = 30001; |

#### 7.1.5 DukptKeyTpe constant definition

|  |
| --- |
| public static final int DUKPT\_KEY\_TYPE\_2TDEA = 1;  public static final int DUKPT\_KEY\_TYPE\_3TDEA = 2;  public static final int DUKPT\_KEY\_TYPE\_AES128 = 3;  public static final int DUKPT\_KEY\_TYPE\_AES192 = 4;  public static final int DUKPT\_KEY\_TYPE\_AES256 = 5; |

#### 7.1.6 Key system type constant definition

|  |
| --- |
| public static final int SEC\_MKSK = 0x00;  public static final int SEC\_DUKPT = 0x01;  public static final int SEC\_RSA\_KEY = 0x02;  public static final int SEC\_SM2\_KEY = 0x03; |

#### 7.1.7 Encryption mode constant definition

|  |
| --- |
| public static final int DATA\_MODE\_ECB = 0;  public static final int DATA\_MODE\_CBC = 1;  public static final int DATA\_MODE\_OFB = 2;  public static final int DATA\_MODE\_CFB = 3; |

#### 7.1.8 Dukpt key select constant definition

|  |
| --- |
| // DUKPT PIN key  public static final int DUKPT\_KEY\_SELECT\_KEY\_PIN = 0;  // DUKPT request and response Mac key  public static final int DUKPT\_KEY\_SELECT\_KEY\_MAC\_BOTH = 1;  // DUKPT response Mac key  public static final int DUKPT\_KEY\_SELECT\_KEY\_MAC\_RSP = 2;  // DUKPT request and response data key  public static final int DUKPT\_KEY\_SELECT\_KEY\_DATA\_BOTH = 3;  // DUKPT response data key  public static final int DUKPT\_KEY\_SELECT\_KEY\_DATA\_RSP = 4;  // DUKPT calculate Mac key（dukpt-aes）  public static final int DUKPT\_KEY\_SELECT\_KEY\_MAC\_GEN = 5;  // DUKPT data encrypt key（dukpt-aes）  public static final int DUKPT\_KEY\_SELECT\_KEY\_DATA\_ENC = 6;  // DUKPT key encryption key（dukpt-aes）  public static final int DUKPT\_KEY\_SELECT\_KEY\_KEY\_ENC\_KEY = 7;  // DUKPT ipek key,aquire service use（dukpt-aes）  public static final int DUKPT\_KEY\_SELECT\_KEY\_DERIVATION = 8;  // DUKPT bdk key,aquire service use（dukpt-aes）  public static final int DUKPT\_KEY\_SELECT\_KEY\_DERIVATION\_INIT = 9; |

#### 7.1.9 RSA transformation constant definition

|  |
| --- |
| public static final String RSA\_TRANSFORMATION\_1 = "RSA/None/NoPadding";  public static final String RSA\_TRANSFORMATION\_2 = "RSA/None/PKCS1Padding";  public static final String RSA\_TRANSFORMATION\_3 = "RSA/ECB/NoPadding";  public static final String RSA\_TRANSFORMATION\_4 = "RSA/ECB/PKCS1Padding";  public static final String RSA\_TRANSFORMATION\_5 = "RSA/ECB/OAEPWithSHA-1AndMGF1Padding";  public static final String RSA\_TRANSFORMATION\_6 = "RSA/ECB/OAEPWithSHA-256AndMGF1Padding";  public static final String RSA\_TRANSFORMATION\_7 = "RSA/ECB/OAEPWithSHA-512AndMGF1Padding"; |

#### 7.1.10 RSA signature algorithm constant definition

|  |
| --- |
| public static final String RSA\_SIGN\_ALG\_1 = "NONEwithRSA";  public static final String RSA\_SIGN\_ALG\_2 = "MD5withRSA";  public static final String RSA\_SIGN\_ALG\_3 = "SHA1withRSA";  public static final String RSA\_SIGN\_ALG\_4 = "SHA256withRSA";  public static final String RSA\_SIGN\_ALG\_5 = "SHA512withRSA";  Note: If signature algorithm io NONEWithRSA, the input data length should less than key module length |

#### 7.1.11 RSA padding mode

|  |
| --- |
| public static final int NOTHING\_PADDING = 0;  public static final int PKCS1\_PADDING = 1;  public static final int PKCS7\_PADDING = 2;  public static final int PKCS5\_PADDING = 3;  public static final int PKCS1\_OAEP\_PADDING = 4;  public static final int PKCS1\_V1\_5\_SHA512 = 5;  public static final int PADDING\_OAEP\_SHA1 = 6; |

#### 7.1.12 Hash type constant definition

|  |
| --- |
| public static final int HASH\_SHA\_TYPE\_1 = 0x00;  public static final int HASH\_SHA\_TYPE\_224 = 0x01;  public static final int HASH\_SHA\_TYPE\_256 = 0x02;  public static final int HASH\_SHA\_TYPE\_384 = 0x03;  public static final int HASH\_SHA\_TYPE\_512 = 0x04;  public static final int HASH\_SM3\_TYPE = 0x05; |

#### 7.1.13 Key control code

|  |
| --- |
| public static final int KEY\_CTRL\_PANPARA = 0x41;  public static final int KEY\_CTRL\_AUTHPARA = 0x42;  public static final int KEY\_CTRL\_APACSMAC = 0x43; |

#### 7.1.14 Inject symmetric key mode definition

|  |
| --- |
| //OAEP mode inject key, depend key is a device cert private key  public static final int INJECT\_OAEP\_MODE = 0x05;  //PKCS1 mode inject key, depend key is a rsa private key  public static final int INJECT\_PKCS1\_MODE = 0x06;  // OWF2 algorithm derive and store key  public static final int INJECT\_DERIVER\_OWF2 = 0x80;  // OWF3 algorithm derive and store key  public static final int INJECT\_DERIVER\_OWF3 = 0x81;  // GOWF algorithm derive and store key  public static final int INJECT\_DERIVER\_GOWF = 0x82;  public static final int INJECT\_DERIVER\_ENC = 0x83; |

#### 7.1.15 Generate device certificate mode definition

|  |
| --- |
| public static final int CERT\_GENERATE\_RSA2048\_E65537\_PVK\_PUK = 0x82; |

#### 7.1.16 Exchange ECC key mode definition

|  |
| --- |
| public static final int SEC\_ECDH\_STEP1\_MODE = 2;  public static final int SEC\_ECDH\_STEP2\_MODE = 3; |

#### 7.1.17 ECC curve param definition

|  |
| --- |
| public static final String SEC\_ECC\_PARAM\_P256 = "P-256";  public static final String SEC\_ECC\_PARAM\_P384 = "P-384";  public static final String SEC\_ECC\_PARAM\_P521 = "P-521"; |

#### 7.1.18 kcv mode definition

|  |
| --- |
| public static final int KCV\_MODE\_NOCHK = 0;  public static final int KCV\_MODE\_CHK0 = 1;  public static final int KCV\_MODE\_CHKFIX = 2;  public static final int KCV\_MODE\_CHKMAC = 3;  public static final int KCV\_MODE\_CHKCMAC = 4; |

#### 7.1.19 Certificate type constant definition

|  |
| --- |
| // ID card  public static final int IDCARD = 536911872;  // Certificate of officers  public static final int ARMYCARD = 536911873;  // Passport  public static final int PASSPORT = 536911874;  // Arrival card  public static final int ARRIVALCARD = 536911875;  // Temporary ID card  public static final int TEMPIDCARD = 536911876;  // Other certificate  public static final int OTHERCARD = 536911877; |

#### 7.1.20 EMV module related constant definition

|  |
| --- |
| // Force online  public static final int FORCE\_ONLINE = 0;  // No online  public static final int NO\_ONLINE = 1;  // CAPK and AID are not exist.  public static final int EXIST\_ALL\_NOT = -1;  // CAPK and AID are exist.  public static final int EXIST\_ALL= 0;  // CAPK is not exist  public static final int EXIST\_CAPK\_NOT = 1;  // AID is not exist.  public static final int EXIST\_AID\_NOT = 2;  // Transaction complete.  public static final int EMV\_RESULT\_FINISHED = 0x9000;  // Transaction abort.  public static final int EMV\_RESULT\_TERMINATION = 0x9001;  // Get PINBLOCK failure.  public static final int EMV\_ERROR\_PINBLOCK = 0x9002;  // Transaction is not supported.  public static final int EMV\_UNSUPPORTED\_TRANS = 0X9003; |

#### 7.1.21 EMV FlowType constant definition

|  |
| --- |
| // Sandard authorization flow  public static final int TYPE\_EMV\_STANDARD = 0x01;  // Simple flow, finished as soon as read card number  public static final int TYPE\_EMV\_BRIEF = 0x02;  // QPASS flow-Skip input PIN in contactless transaction  public static final int TYPE\_NFC\_SKIP\_CVM = 0x03;  // Contactless speedup flow  public static final int TYPE\_NFC\_SPEEDUP = 0x04; |

#### 7.1.22 EMV clean data constant definition

|  |
| --- |
| // Clear all data  public static final int OP\_CLEAR\_DATA\_ALL = 0;  // Clear terminal data  public static final int OP\_CLEAR\_DATA\_TERMINAL = 1;  // Clear card data  public static final int OP\_CLEAR\_DATA\_CARD = 2; |

#### 7.1.23 EMV TLV operation type constant definition

|  |
| --- |
| // Normal  public static final int OP\_NORMAL = 0;  // PayPass  public static final int OP\_PAYPASS = 1;  // PayWave  public static final int OP\_PAYWAVE = 2;  // MIR  public static final int OP\_MIR = 3;  // PAGO  public static final int OP\_PAGO = 4;  // JCB  public static final int OP\_JCB = 5;  // PURE  public static final int OP\_PURE = 6;  // AE  public static final int OP\_AE = 7;  // FLASH  public static final int OP\_FLASH = 8;  // DPAS  public static final int OP\_DPAS = 9;  // RUPAY  public static final int OP\_RUPAY = 10;  // EFTPOS  public static final int OP\_EFTPOS = 11;  // AID RELEVANT  public static final int OP\_AID\_RELEVANT = 101;  // CPACE  public static final int OP\_CPACE = 13;  // Add customized tag  public static final int OP\_ADD\_SELF\_DEFINE\_TAG = 102;  // Delete customized tag  public static final int OP\_DEL\_SELF\_DEFINE\_TAG = 103; |

#### 7.1.24 EMV kernel type definition

|  |
| --- |
| // EMV(Contact)  public static final int EMV = 0;  // QPBOC  public static final int QPBOC = 1;  // PAYPASS  public static final int PAYPASS = 2;  // PAYWAVE  public static final int PAYWAVE = 3;  // AE  public static final int AE = 4;  // DISCOVER  public static final int DISCOVER = 5;  // JCB  public static final int JCB = 6;  // FLASH  public static final int FLASH = 7;  // MIR  public static final int MIR = 8;  // MCCS  public static final int MCCS = 9;  // RUPAY  public static final int RUPAY = 10;  // PAGO  public static final int PAGO = 11;  // EFTPOS  public static final int EFTPOS = 12;  // SAMSUNGPAY  public static final int SAMSUNGPAY = 13;  // CPACE  public static final int CPACE = 15; |

#### 7.1.25 EMV param type definition

|  |
| --- |
| // CONTACT/CONTACTLESS(default)  public static final int DEFAULT = 0;  // CONTACT  public static final int CONTACT = 1;  // CONTACTLESS  public static final int CONTACTLESS = 2; |

#### 7.1.26 EMV transaction result code definition

|  |
| --- |
| // Success  public static final int SUCCESS = 0;  // offline approval  public static final int OFFLINE\_APPROVAL = 1;  // offline denied  public static final int OFFLINE\_DECLINE = 2;  // Reserved  public static final int RESERVE = 3;  // Retap  public static final int TRY\_AGAIN = 4;  // online approval  public static final int ONLINE\_APPROVAL = 5;  // online denied  public static final int ONLINE\_DECLINE = 6; |

#### 7.1.27 EMV online result definition

|  |
| --- |
| // online approval  public static final int ONLINE\_APPROVAL = 0;  // online decline  public static final int ONLINE\_DECLINE = 1;  // online fail  public static final int ONLINE\_FAIL = 2;  // online approval and 2nd tap card for RUPAY  public static final int ONLINE\_APPROVAL\_2\_TAP = 3;  // onlie decline and 2nd tap card for RUPAY  public static final int ONLINE\_DECLINE\_2\_TAP = 4;  // online fail and 2nd tap card for RUPAY  public static final int ONLINE\_FAIL\_2\_TAP = 5;  // offline decline for RUPAY  public static final int OFFLINE\_DECLINE = 6;  // online fail and Full online for RUPAY  public static final int ONLINE\_FAIL\_FULL\_ONLINE = 7;  // online success and no script for RUPAY  public static final int ONLINE\_SUCCESS\_NO\_SCRIPT = 8;  // Only for AE: the Online PIN capture fails, when ARC (Tag 8A) is Request Online PIN from the Issuer. (only for SDK internal)  public static final int ONLINE\_FAIL\_ONLINEPIN = 9; |

#### 7.1.28 AID function behavior constant definition

|  |
| --- |
| // Add or update an AID.  public static final int ACTION\_AID\_ADD = 0x00;  // Delete all AIDs.  public static final int ACTION\_AID\_DEL = 0x01; |

#### 7.1.29 CAPK function behavior constant definition

|  |
| --- |
| // Add or update a CAPK.  public static final int ACTION\_CAPK\_ADD = 0x00;  // Delete all CAPKs.  public static final int ACTION\_CAPK\_DEL = 0x01; |

#### 7.1.30 System parameter constant definition

|  |
| --- |
| // Hardware version  public static final String HARDWARE\_VERSION = "HardwareVersion";  // Firmware version  public static final String FIRMWARE\_VERSION = "FirmwareVersion";  // SN  public static final String SN = "SN";  // PN  public static final String PN = "PN";  // TUSN  public static final String TUSN = "TUSN";  // Device code(eg.W6900(4G))  public static final String DEVICE\_CODE = "DeviceCode";  // Device model(eg.P1N)  public static final String DEVICE\_MODEL = "DeviceModel";  // Reserved. (value is Json format- extensible)  public static final String RESERVED = "Reserved";  // PIN input mode  public static final String PINPAD\_MODE = "PinPadMode";  // Contactless param for A card  public static final String PCD\_PARAM\_A = "PCD\_PARAM\_A";  // Contactless param for B card  public static final String PCD\_PARAM\_B = "PCD\_PARAM\_B";  // Contactless param for Felica card  public static final String PCD\_PARAM\_C = "PCD\_PARAM\_C";  // Key same check flag  public static final String SEC\_MODE = "SecMode";  // IC driver version  public static final String PCD\_IFM\_VERSION = "PCD\_IFMVersion";  // SAM slots count  public static final String SAM\_COUNT = "SAM";  //version of libicsam.a library  public static final String IFM\_LIB\_VERSION = "IfmLibVersion";  // version number of manufacturer's magnetic stripe card decoding library  public static final String MSR\_VERSION = "MsrVersion";  //Version of [sunmiposapi.jar]  public static final String POSAPI\_VERSION = "posapiVersion";  //Is support detect RTC battery voltage, 0-not support, 1-supported  public static final String RTC\_BAT\_VOL\_DET = "RTCBATVOLDET";  //Is support SRED, “0”-unsupported, “1”-supported  public static final String SRED = "sred";  //PCIPTS version  public static final String PCI\_PTS\_VERSION = "PCIPTSVersion";  //RNIB auth version  public static final String RNIB\_VERSION = "RNIBVersion";  // EMV version  public static final String EMV\_VERSION = "EMVVersion";  // Paypass version  public static final String PAYPASS\_VERSION = "PaypassVersion";  // Paywave version  public static final String PAYWAVE\_VERSION = "PaywaveVersion";  // QPBOC version  public static final String QPBOC\_VERSION = "QPBOCVersion";  // Entry version  public static final String ENTRY\_VERSION = "EntryVersion";  // Mir version  public static final String MIR\_VERSION = "MirVersion";  // JCB version  public static final String JCB\_VERSION = "JCBVersion";  // Pago version  public static final String PAGO\_VERSION = "PAGOVersion";  // PURE version  public static final String JCB\_VERSION = "PUREVersion";  // AE version  public static final String PAGO\_VERSION = "AEVersion";  //FLASH version  public static final String FLASH\_VERSION = "FLASHVersion";  // DPA Sversion  public static final String DPAS\_VERSION = "DPASVersion";  // APEMV version  public static final String APEMV\_VERSION = "APEMVVersion";  // EFTPOS version  public static final String EFTPOS\_VERSION = "EFTPOSVersion";  // EMVBase version  public static final String EMVBASE\_VERSION = "EMVBaseVersion";  // KernelDirect version  public static final String KD\_VERSION = "KDVersion";  // RUPAY version  public static final String RUPAY\_VERSION = "RUPAYVersion";  // SAMSUNGPAY version  public static final String SAMSUNGPAY\_VERSION = "SAMSUNGPAYVersion";  // CPACE version  public static final String CPACE\_VERSION = "CPACEVersion";  // EMV kernel checksum  public static final String EMV\_KERNEL\_CHECKSUM = "EmvKernelCheckSum";  // Pure release date  public static final String PURE\_RELEASE\_DATE = "PUREReleaseDate";  // EFTPOS release date  public static final String EFTPOS\_RELEASE\_DATE = "EFTPOSReleaseDate";  // EMV release date  public static final String EMV\_RELEASE\_DATE = "EMVReleaseDate";  // Paypass release date  public static final String PAYPASS\_RELEASE\_DATE = "PaypassReleaseDate";  // Paywave release date  public static final String PAYWAVE\_RELEASE\_DATE = "PaywaveReleaseDate";  // QPBOC release date  public static final String QPBOC\_RELEASE\_DATE = "QPBOCReleaseDate";  // Entry release date  public static final String ENTRY\_RELEASE\_DATE = "EntryReleaseDate";  // Mir release date  public static final String MIR\_RELEASE\_DATE = "MirReleaseDate";  // JCB release date  public static final String JCB\_RELEASE\_DATE = "JCBReleaseDate";  // Pago release date  public static final String PAGO\_RELEASE\_DATE = "PAGOReleaseDate";  // AE release date  public static final String AE\_RELEASE\_DATE = "AEReleaseDate";  // FLASH release date  public static final String FLASH\_RELEASE\_DATE = "FLASHReleaseDate";  // DPAS release date  public static final String DPAS\_RELEASE\_DATE = "DPASReleaseDate";  // EMVBase release date  public static final String EMVBASE\_RELEASE\_DATE = "EMVBaseReleaseDate";  // KernelDirect release date  public static final String KD\_RELEASE\_DATE = "KDReleaseDate";  // Rupay release date  public static final String RUPAY\_RELEASE\_DATE = "RUPAYReleaseDate";  // SAMSUNGPAY release date  public static final String SAMSUNGPAY\_RELEASE\_DATE = "SAMSUNGPAYReleaseDate";  // CPACE release date  public static final String CPACE\_RELEASE\_DATE = "CPACEReleaseDate"; |

#### 7.1.31 Led constant definition

|  |
| --- |
| // Red light  public static final int RED\_LIGHT = 1;  // Green light  public static final int GREEN\_LIGHT = 2;  // Yellow light  public static final int YELLOW\_LIGHT = 3;  // Blue light  public static final int BLUE\_LIGHT = 4;  // White light  public static final int WHITE\_LIGHT = 5;  // Red corner light (supported on P3\_MIX, P3)  public static final int CORNER\_RED\_LIGHT = 6;  // Green corner light (supported on P3\_MIX, P3)  public static final int CORNER\_GREEN\_LIGHT = 7  // Blue corner light (supported on P3\_MIX, P3)  public static final int CORNER\_BLUE\_LIGHT = 8;  // Yellow indicator light (supported on P3\_MIX)  public static final int INDICATOR\_YELLOW\_LIGHT = 9; |

#### 7.1.32 PinPad mode constant definition

|  |
| --- |
| // Normal  public static final String MODE\_NORMAL = "Normal";  // MeiTuan  public static final String MODE\_MEITUAN = "MeiTuan";  // Silent  public static final String MODE\_SILENT = "Silent"; |

#### 7.1.33 PinBlock format constant definition

|  |
| --- |
| // support DES/TDES/SM4 key, input PAN is formatted 12 bytes ASCII charactors  public static final int SEC\_PIN\_BLK\_ISO\_FMT0 = 0;  // support DES/TDES key, input PAN meaningless  public static final int SEC\_PIN\_BLK\_ISO\_FMT1 = 1;  // unsupported pinBlock type  public static final int SEC\_PIN\_BLK\_ISO\_FMT2 = 2;  // support DES/TDES key, input PAN is formatted 12 bytes ASCII charactors  public static final int SEC\_PIN\_BLK\_ISO\_FMT3 = 3;  support DES/TDES key, input PAN is 6 bytes serial number  public static final int SEC\_PIN\_BLK\_EPS = 4;  // unsupported pinBlock type  public static final int SEC\_PIN\_BLK\_IBM\_3621 = 5;  // unsupported pinBlock type  public static final int SEC\_PIN\_BLK\_IBM\_3624 = 6;  // only support AES key, input PAN is formatted 12-19 bytes ASCII charactors  public static final int SEC\_PIN\_BLK\_ISO\_FMT4 = 7; |

#### 7.1.34 Keyboard beep mode constant definiton

|  |
| --- |
| //Open keyboard sound  public static final String MODE\_ON = "ON";  //Close keyboard sound  public static final String MODE\_OFF = "OFF"; |

#### 7.1.35 Card exist status

|  |
| --- |
| // Card absent  public static final int CARD\_ABSENT = 0x01;  // Card present  public static final int CARD\_PRESENT = 0x02; |

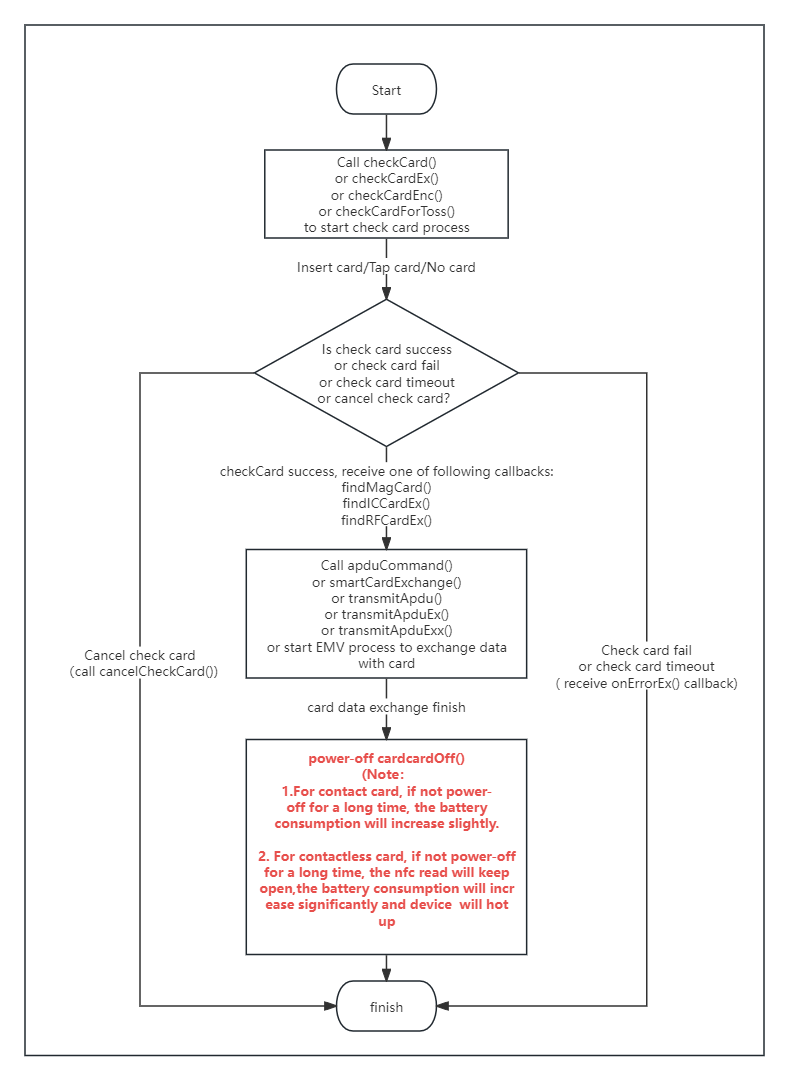
### 7.2 PAN data interception

1. Starting from the penultimate character at the end of the card number (Field 2), from right to left, the consecutive 12 characters are PAN data, PAN data participate in PIN encryption (calculate PinBlock) or decryption.
2. For example, the card number is 6225882145611077, PAN data after interception is 588214561107, convert to byte array is (Java code):

byte[] pan= "588214561107".getBytes ("US-ASCII");

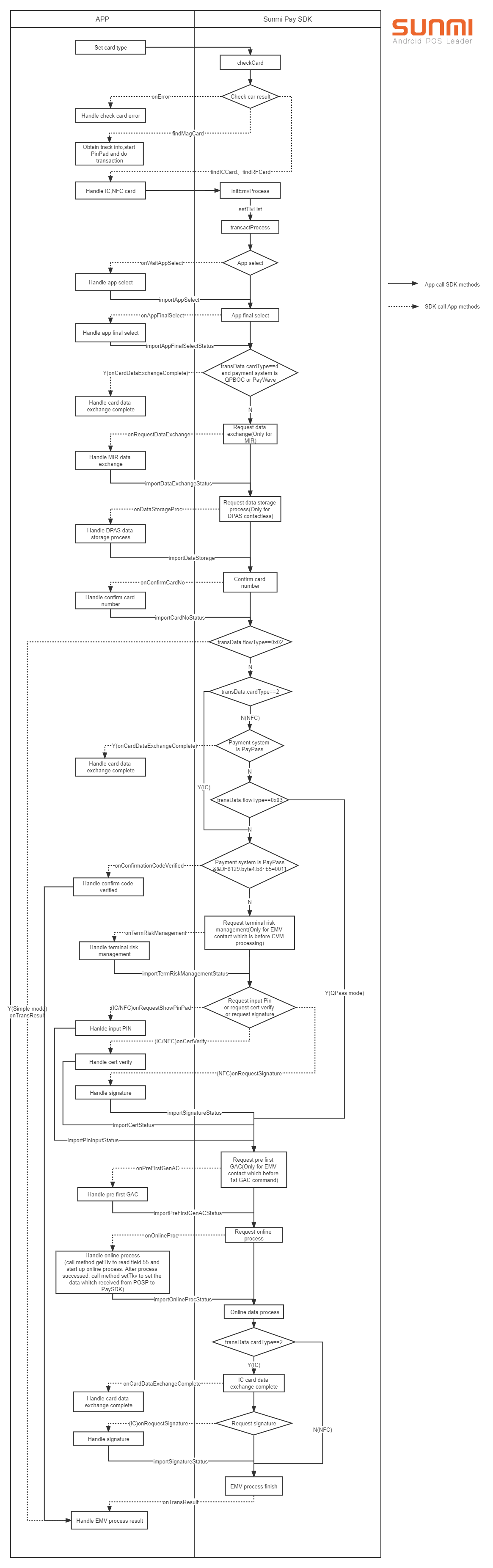
1. All the PAN data that passed to SDK should intercepted in this way. If the pinblock result is incorrect, client should check whether the passed PAN data is in accordance with the above rules or not, or check whether the input PIN number is correct or not.

### 7.3 Read card flow chart

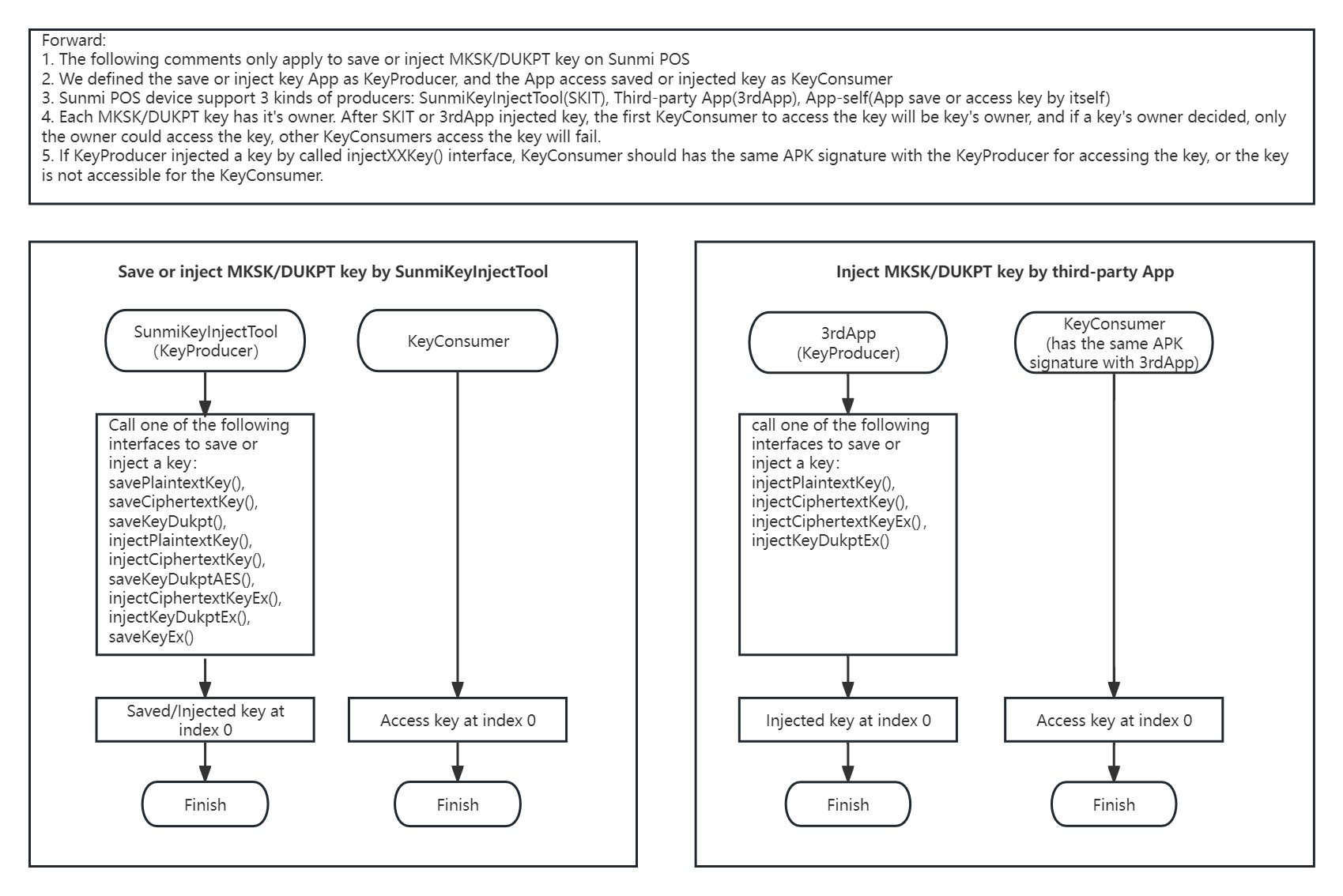


### 7.4 EMV trans flow chart





### 7.5 Key inject/access flow chart



### 7.6 Key system and key index range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key system | Key algorithm type | Is open key partition | Is Brazil CKD | Normal Key index range |
| SEC\_MKSK | -- | Yes | -- | 0-199 |
| -- | No | -- | 10001-12000 |
| SEC\_DUKPT | 3DES | Yes | Yes | 0-99 |
| No | 0-9, 1100-1199 |
| No | -- | 20001-21000 |
| AES | Yes | Yes | 100-199 |
| No | 10-19, 2100-2199 |
| No | -- | 30001-31000 |
| SEC\_RSA\_KEY | -- | Yes | -- | 0-19, 20-39 |
| -- | No | -- | 0-19, 35001-35200 |
| SEC\_SM2\_KEY | -- | -- | -- | 0-9 |
| -- | -- |  | 0-9 |
| SEC\_ECC\_KEY | -- | Yes | -- | -- |
| -- | No | -- | -- |
| SEC\_DEVICE\_CERT | -- | Yes | -- | 9001-9008 |
| -- | No | -- | 9001-9008 |
| Note：  1.SM2 key not support key partition  2.For interface generateRSAKeypair() and injectRSAKey(), no matter open or close key partition, key index range is always 0-19，and the saved key not support key partition.  3. For interface generateRSAKeypairEx() and injectRSAKeyEx(), when key partition is open, key index range is 20-39, and when key partition is closed, key index range is 35001-35200  4. For other interfaces which use RSA for calculation, key index range refer to the SEC\_RSA\_KEY section in this table | | | | |

### 7.7 Key partition description

1. Sunmi Pay SDK support key partition and open it by default, client could open or close key parttion according to their requirements. **Switch key partition may cause previously stored key lost. It’s recommended not to switch key partition, just use SDK default value.**
2. For the key index range after open or close key partition, please refer to section [7.4 key system and key index range](#KeySystemAndKeyIndexRange)
3. After open key partition, SDK isolate key for each client APP, APP A saved key only accessible for itself, for other APP, it is not accessible.
4. After close key partition, SDK not isolate key for each client APP, APP A saved key is accessible for all client APPs, exist key overrided or modified risk.
5. Currently, key partition is not applicable to SM2 key system, client APP A saved SM2 key is accessible for all client APPs, exist key overrided or modified risk.

### 7.8 Key check value param description

1. If kcv is null, the kcv mode is **KCV\_ MODE\_ NOCHK**, SDK will not check the input kcv
2. If kcv is not null and not set param **kcvMode**, the default kcv mode is **KCV\_ MODE\_ CHK0**, kcv will be generate as following rules:
3. DES/TDES key encrypt 8 bytes 0x00 data, the first 4 bytes of ciphertext is kcv
4. AES key encrypt 16 bytes 0x00, the first 4 bytes of ciphertext is kcv
5. SM4 key encrypt 16 bytes 0x00, the whole 16 bytes ciphertext is kcv
6. If kcv is not null and set param **kcvMode**, kcv is calculated by kcvMode
7. If input kcv length great than 4 or 16, the kcv will be intercepted
8. For DES/TDES, the input kcv length<=8B, for AES/SM4, input kcv length<=16B

### 7.9 SRED description

1. SRED (Secure Reading and Exchange of Data) is one of the security requirements in the PCI PTS POI specification, which ensures that the terminal immediately encrypts and desensitizes card account related data after reading it, and maintains an encrypted state during transmission to reduce the risk of data leakage. Refer to chapters B10 and B23 of the specifications: <https://docs-prv.pcisecuritystandards.org/PTS/Derived%20Test%20Requirements/PCI_PTS_POI_DTRs_v6.2.pdf>

2. For Sunmi terminals, turning on/off SRED has following effects:

(1) After enabling SRED, client application are not allowed to obtain account data in plaintext through the SDK interface, including checkCard(), checkCardEx(), getTlv(), getTlvList(), onConfirmCardNo(), etc. Only interface checkCardEnc() and getAccountSecData() are allowed to obtain account ciphertext or truncate data.

(2) After disabling SRED, allow client application to obtain complete account data in plaintext through the SDK interface, including checkCard(), checkCardEx(), getTlv(), getTlvList(), onConfirmCardNo(), etc.

3. When calling interface EMVOptV2. setAccountDatasecParam(), if parameter **bundle** does not contain the keyword **sred** or contains **sred** and set it’s value as true, sred function will be opened. Example code:

Bundle bundle = new Bundle();

bundle.putInt("encKeySystem", Security.SEC\_MKSK);

bundle.putInt("encKeyIndex", KEY\_INDEX\_MKSK);

bundle.putInt("encMode", Security.DATA\_MODE\_CBC);

bundle.putByteArray("encIv", new byte[16]);

// Set sred as true or not set keyword sred will turn on the sred function

bundle.putBoolean("sred", true);

int code = MyApplication.app.emvOptV2.setAccountDataSecParam(bundle);

Log.e(TAG, "setAccountDataSecParam() with mksk ,code:" + code);

3. Start from v5.0.34, SRED function switches from global mode to application related mode，App1 turn-on sred only affect App1 itself, not affect other Apps.

### 7.10 EMV default data

#### 7.1.1 EMV default AIDs

|  |  |  |
| --- | --- | --- |
| NO. | AID | Applicable to card brand |
| 1 | 9F0608A000000333010100DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 2 | 9F0608A000000333010101DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 3 | 9F0608A000000333010102DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 4 | 9F0608A000000333010103DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 5 | 9F0608A000000333010107DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 6 | 9F0607A0000000033010DF0101009F08020140DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180100DF2006000200000000DF2106000200000000DFC10B0100 | Visa |
| 7 | 9F0607A0000000032010DF0101009F08020140DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180100DF2006000200000000DF2106000200000000DFC10B0100 | Visa |
| 8 | 9F0607A0000000031010DF0101009F08020140DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180100DF2006000200000000DF2106000200000000DFC10B0100 | Visa |
| 9 | 9F0607A0000000043060DF0101009F08020002DF1105FC5058A000DF1205F85058F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | MasterCard |
| 10 | 9F0607A0000000041010DF0101009F08020002DF1105FC5080A000DF1205F85080F800DF130504000000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180100DF2006000200000000DF2106000200000000DFC10B0100 | MasterCard |
| 11 | 9F0608A000000333010106DF0101009F08020020DF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180101DF2006000200000000DF2106000200000000DFC10B0100 | UnionPay |
| 12 | 9F0607A0000000651010DF0101009F08020200DF1105FC6024A800DF1205FC60ACF800DF130500100000009F1B0400000000DF150400000000DF160199DF170199DF14039F3704DF180100DF2006000200000000DF2106000200000000DFC10B0100 | JCB |
| 13 | 9F0607A0000000031010DF0101009F0902008CDF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160100DF170100DF180101DF2106000000005001DF2006000000006000DF1906000000005000DFC10B0100 | Visa |
| 14 | 9F0607A0000000032010DF0101009F0902008CDF1105D84000A800DF1205D84004F800DF130500100000009F1B0400000000DF150400000000DF160100DF170100DF180101DF2106000000010000DF2006000000100000DF1906000000000010DFC10B0100 | Visa |
| 15 | 9F0607A0000000041010DF0101009F09020002DF1105FC50A8A000DF1205F850A8F800DF130504000000009F1B0400000000DF150400000000DF160100DF170100DF180101DF2106000000010000DF2006000000100000DF1906000000000010DFC10B0100 | MasterCard |
| 16 | 9F0607A0000000043060DF0101009F09020002DF1105FC50A8A000DF1205F850A8F800DF130504000000009F1B0400000000DF150400000000DF160100DF170100DF180101DF2106000000010000DF2006000000100000DF1906000000000010DFC10B0100 | MasterCard |
| 17 | 9F0607A0000000046000DF0101009F09020002DF1105FC50A8A000DF1205F850A8F800DF130504000000009F1B0400000000DF150400000000DF160100DF170100DF180101DF2106000000010000DF2006000000100000DF1906000000000010DFC10B0100 | MasterCard |
| 18 | 9F0607A00000000410109F09020000DF11050000000000DF12050000000000DF130500000000009F1B0400000000DF150400000000DF160100DF170100DF140b9F37049F47018F019F32019F3C0206439F3D0102DF1906000000000000DF2006999999999999DF21060000000030009F4E009F150211119F160c746573746D65726368616E74DFC10B0100 | MasterCard |
| 19 | 9F0607A00000000310109F09020000DF11050000000000DF12050000000000DF130500000000009F1B0400000000DF150400000000DF160100DF170100DF14039F37049F3C0206439F3D0102DF1906000000000000DF2006999999999999DF21060000000000009F4E009F150211119F160c746573746D65726368616E74DFC10B0100 | Visa |

#### 7.1.1 EMV default CAPKs

|  |  |  |
| --- | --- | --- |
| NO. | CAPK | Applicable to card brand |
| 1 | 9F0605A0000003339F220102DF0503211231DF060101DF070101DF028190A3767ABD1B6AA69D7F3FBF28C092DE9ED1E658BA5F0909AF7A1CCD907373B7210FDEB16287BA8E78E1529F443976FD27F991EC67D95E5F4E96B127CAB2396A94D6E45CDA44CA4C4867570D6B07542F8D4BF9FF97975DB9891515E66F525D2B3CBEB6D662BFB6C3F338E93B02142BFC44173A3764C56AADD202075B26DC2F9F7D7AE74BD7D00FD05EE430032663D27A57DF040103DF031403BB335A8549A03B87AB089D006F60852E4B8060 | UnionPay |
| 2 | 9F0605A0000003339F220103DF0503221231DF060101DF070101DF0281B0B0627DEE87864F9C18C13B9A1F025448BF13C58380C91F4CEBA9F9BCB214FF8414E9B59D6ABA10F941C7331768F47B2127907D857FA39AAF8CE02045DD01619D689EE731C551159BE7EB2D51A372FF56B556E5CB2FDE36E23073A44CA215D6C26CA68847B388E39520E0026E62294B557D6470440CA0AEFC9438C923AEC9B2098D6D3A1AF5E8B1DE36F4B53040109D89B77CAFAF70C26C601ABDF59EEC0FDC8A99089140CD2E817E335175B03B7AA33DDF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | UnionPay |
| 3 || UnionPay |
| 4 | 9F0605A0000000039F220147DF028190F562B594C911C0310AEFB002BDFE01CC8C2F1351CAEDD12C5210F4C3507BC106E01BBF94362392F3A66496079CBC8EAD09D5D942B195CA12A8A1AABAEF1D5545AF1693484764DEB625100E15F630510F3E2FB9E00D81F86D4C6F099307182E30F67FBE9C3D91F557BA5AAA1ACFBB625ABFA0176D4CEF3DFE1892BA1EC49F57B5DB6C53147F1DC9FD844298E238A94FEBDF040103DF0503201231 | Visa |
| 5 | 9F0605A0000000039F220101DF028180C696034213D7D8546984579D1D0F0EA519CFF8DEFFC429354CF3A871A6F7183F1228DA5C7470C055387100CB935A712C4E2864DF5D64BA93FE7E63E71F25B1E5F5298575EBE1C63AA617706917911DC2A75AC28B251C7EF40F2365912490B939BCA2124A30A28F54402C34AECA331AB67E1E79B285DD5771B5D9FF79EA630B75DF040103DF0503201231 | Visa |
| 6 | 9F0605A0000000039F220107DF028190A89F25A56FA6DA258C8CA8B40427D927B4A1EB4D7EA326BBB12F97DED70AE5E4480FC9C5E8A972177110A1CC318D06D2F8F5C4844AC5FA79A4DC470BB11ED635699C17081B90F1B984F12E92C1C529276D8AF8EC7F28492097D8CD5BECEA16FE4088F6CFAB4A1B42328A1B996F9278B0B7E3311CA5EF856C2F888474B83612A82E4E00D0CD4069A6783140433D50725FDF040103DF0503201231 | Visa |
| 7 | 9F0605A0000000039F220108DF0281B0D9FD6ED75D51D0E30664BD157023EAA1FFA871E4DA65672B863D255E81E137A51DE4F72BCC9E44ACE12127F87E263D3AF9DD9CF35CA4A7B01E907000BA85D24954C2FCA3074825DDD4C0C8F186CB020F683E02F2DEAD3969133F06F7845166ACEB57CA0FC2603445469811D293BFEFBAFAB57631B3DD91E796BF850A25012F1AE38F05AA5C4D6D03B1DC2E568612785938BBC9B3CD3A910C1DA55A5A9218ACE0F7A21287752682F15832A678D6E1ED0BDF040103DF0503201231 | Visa |
| 8 | 9F0605A0000000039F220105DF0260D0135CE8A4436C7F9D5CC66547E30EA402F98105B71722E24BC08DCC80AB7E71EC23B8CE6A1DC6AC2A8CF55543D74A8AE7B388F9B174B7F0D756C22CBB5974F9016A56B601CCA64C71F04B78E86C501B193A5556D5389ECE4DEA258AB97F52A3DF060101DF070101DF040103DF031486DF041E7995023552A79E2623E49180C0CD957ADF0503151231 | Visa |
| 9 | 9F0605A0000000049F2201FADF0503202001DF060101DF070101DF028190A90FCD55AA2D5D9963E35ED0F440177699832F49C6BAB15CDAE5794BE93F934D4462D5D12762E48C38BA83D8445DEAA74195A301A102B2F114EADA0D180EE5E7A5C73E0C4E11F67A43DDAB5D55683B1474CC0627F44B8D3088A492FFAADAD4F42422D0E7013536C3C49AD3D0FAE96459B0F6B1B6056538A3D6D44640F94467B108867DEC40FAAECD740C00E2B7A8852DDF040103DF03145BED4068D96EA16D2D77E03D6036FC7A160EA99C | MasterCard |
| 10 | 9F0605A0000000049F2201F1DF0503202001DF060101DF070101DF0281b0A0DCF4BDE19C3546B4B6F0414D174DDE294AABBB828C5A834D73AAE27C99B0B053A90278007239B6459FF0BBCD7B4B9C6C50AC02CE91368DA1BD21AAEADBC65347337D89B68F5C99A09D05BE02DD1F8C5BA20E2F13FB2A27C41D3F85CAD5CF6668E75851EC66EDBF98851FD4E42C44C1D59F5984703B27D5B9F21B8FA0D93279FBBF69E090642909C9EA27F898959541AA6757F5F624104F6E1D3A9532F2A6E51515AEAD1B43B3D7835088A2FAFA7BE7DF040103DF0314D8E68DA167AB5A85D8C3D55ECB9B0517A1A5B4BB | MasterCard |
| 11 | | MasterCard |
| 12 | 9F0605A0000000049F220147DF0503202001DF060101DF070101DF028190F562B594C911C0310AEFB002BDFE01CC8C2F1351CAEDD12C5210F4C3507BC106E01BBF94362392F3A66496079CBC8EAD09D5D942B195CA12A8A1AABAEF1D5545AF1693484764DEB625100E15F630510F3E2FB9E00D81F86D4C6F099307182E30F67FBE9C3D91F557BA5AAA1ACFBB625ABFA0176D4CEF3DFE1892BA1EC49F57B5DB6C53147F1DC9FD844298E238A94FEBDF040103DF03146304B43BAF1342F914ABD4B55EE817CEF1FA557F | MasterCard |
| 13 | | MasterCard |
| 14 | | MasterCard |
| 15 | 9F0605A0000000049F220105DF0503202001DF060101DF070101DF0281b0B8048ABC30C90D976336543E3FD7091C8FE4800DF820ED55E7E94813ED00555B573FECA3D84AF6131A651D66CFF4284FB13B635EDD0EE40176D8BF04B7FD1C7BACF9AC7327DFAA8AA72D10DB3B8E70B2DDD811CB4196525EA386ACC33C0D9D4575916469C4E4F53E8E1C912CC618CB22DDE7C3568E90022E6BBA770202E4522A2DD623D180E215BD1D1507FE3DC90CA310D27B3EFCCD8F83DE3052CAD1E48938C68D095AAC91B5F37E28BB49EC7ED597DF040103DF0314EBFA0D5D06D8CE702DA3EAE890701D45E274C845 | MasterCard |
| 16 | 9F0605A0000000039F220195DF0503202001DF060101DF070101DF028190BE9E1FA5E9A803852999C4AB432DB28600DCD9DAB76DFAAA47355A0FE37B1508AC6BF38860D3C6C2E5B12A3CAAF2A7005A7241EBAA7771112C74CF9A0634652FBCA0E5980C54A64761EA101A114E0F0B5572ADD57D010B7C9C887E104CA4EE1272DA66D997B9A90B5A6D624AB6C57E73C8F919000EB5F684898EF8C3DBEFB330C62660BED88EA78E909AFF05F6DA627BDF040103DF0314EE1511CEC71020A9B90443B37B1D5F6E703030F6 | Visa |
| 17 | 9F0605A0000000039F220199DF0503202001DF060101DF070101DF028180AB79FCC9520896967E776E64444E5DCDD6E13611874F3985722520425295EEA4BD0C2781DE7F31CD3D041F565F747306EED62954B17EDABA3A6C5B85A1DE1BEB9A34141AF38FCF8279C9DEA0D5A6710D08DB4124F041945587E20359BAB47B7575AD94262D4B25F264AF33DEDCF28E09615E937DE32EDC03C54445FE7E382777DF040103DF03144ABFFD6B1C51212D05552E431C5B17007D2F5E6D | Visa |
| 18 | 9F0605A0000000039F220150DF0503202001DF060101DF070101DF028180D11197590057B84196C2F4D11A8F3C05408F422A35D702F90106EA5B019BB28AE607AA9CDEBCD0D81A38D48C7EBB0062D287369EC0C42124246AC30D80CD602AB7238D51084DED4698162C59D25EAC1E66255B4DB2352526EF0982C3B8AD3D1CCE85B01DB5788E75E09F44BE7361366DEF9D1E1317B05E5D0FF5290F88A0DB47DF040103DF03145765295089960938BAAA4431506E424295F98BD0 | Visa |
| 19 | 9F0605A0000000039F220192DF0503202001DF060101DF070101DF0281b0996AF56F569187D09293C14810450ED8EE3357397B18A2458EFAA92DA3B6DF6514EC060195318FD43BE9B8F0CC669E3F844057CBDDF8BDA191BB64473BC8DC9A730DB8F6B4EDE3924186FFD9B8C7735789C23A36BA0B8AF65372EB57EA5D89E7D14E9C7B6B557460F10885DA16AC923F15AF3758F0F03EBD3C5C2C949CBA306DB44E6A2C076C5F67E281D7EF56785DC4D75945E491F01918800A9E2DC66F60080566CE0DAF8D17EAD46AD8E30A247C9FDF040103DF0314429C954A3859CEF91295F663C963E582ED6EB253 | Visa |
| 20 | | Visa |
| 21 | 9F0605A0000005249F220104DF0503211231DF060101DF070101DF0803000000DF028180A1F5E1C9BD8650BD43AB6EE56B891EF7459C0A24FA84F9127D1A6C79D4930F6DB1852E2510F18B61CD354DB83A356BD190B88AB8DF04284D02A4204A7B6CB7C5551977A9B36379CA3DE1A08E69F301C95CC1C20506959275F41723DD5D2925290579E5A95B0DF6323FC8E9273D6F849198C4996209166D9BFC973C361CC826E1DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 22 || Rupay |
| 23 || Rupay |
| 24 | 9F0605A0000005249F220103DF0503211231DF060101DF070101DF0803000000DF02818094EA62F6D58320E354C022ADDCF0559D8CF206CD92E869564905CE21D720F971B7AEA374830EBE1757115A85E088D41C6B77CF5EC821F30B1D890417BF2FA31E5908DED5FA677F8C7B184AD09028FDDE96B6A6109850AA800175EABCDBBB684A96C2EB6379DFEA08D32FE2331FE103233AD58DCDB1E6E077CB9F24EAEC5C25AFDF0403010001DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 25 | 9F0605A0000005249F2201F1DF0503211231DF060101DF070101DF0803000000DF0281B0A4DC71056B6607EFD116625AB0506D11DEEB4BAED6475AEF11702C90604BA5D7F2F632236474F0C79E3FBE160A6ABAC126730BD6853ECA412F38CD16DD48129CD53D91F1BB9196F2465C3014FCE2CA702C41472ED0609BD238052FE9C07F38DE7268DF1A0083E4DE20814B5BBFA9ADC33916A049155951648821A05C20CCFD7E8BC141EF3E29A3F306325B13017EDC38D62E03B57A371DFC578274DC78C3FBD6C5E60A0AF2901CAF3B0DD6975EFB5421DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 26 | 9F0605A0000005249F220106DF0503211231DF060101DF070101DF0803000000DF028180A1F5E1C9BD8650BD43AB6EE56B891EF7459C0A24FA84F9127D1A6C79D4930F6DB1852E2510F18B61CD354DB83A356BD190B88AB8DF04284D02A4204A7B6CB7C5551977A9B36379CA3DE1A08E69F301C95CC1C20506959275F41723DD5D2925290579E5A95B0DF6323FC8E9273D6F849198C4996209166D9BFC973C361CC826E1DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 27 || Rupay |
| 28 | 9F0605A0000005249F220100DF0503211231DF060101DF070101DF0803000000DF02819098F0C770F23864C2E766DF02D1E833DFF4FFE92D696E1642F0A88C5694C6479D16DB1537BFE29E4FDC6E6E8AFD1B0EB7EA0124723C333179BF19E93F10658B2F776E829E87DAEDA9C94A8B3382199A350C077977C97AFF08FD11310AC950A72C3CA5002EF513FCCC286E646E3C5387535D509514B3B326E1234F9CB48C36DDD44B416D23654034A66F403BA511C5EFA3DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 29 | 9F0605A0000005249F2201F3DF0503211231DF060101DF070101DF0803000000DF02819098F0C770F23864C2E766DF02D1E833DFF4FFE92D696E1642F0A88C5694C6479D16DB1537BFE29E4FDC6E6E8AFD1B0EB7EA0124723C333179BF19E93F10658B2F776E829E87DAEDA9C94A8B3382199A350C077977C97AFF08FD11310AC950A72C3CA5002EF513FCCC286E646E3C5387535D509514B3B326E1234F9CB48C36DDD44B416D23654034A66F403BA511C5EFA3DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 30 || Rupay |
| 31 || Rupay |
| 32 | 9F0605A0000005249F2201F7DF0503211231DF060101DF070101DF0803000000DF028180924D9576F8FB29F7E086265004EFB5897123F4FC6264E7AA61A53A352D83EFEC14B895101E8F9A00DF895FC780F13CFB5E43471E56BD51B7A6DC48044FA9BEE87032ACBBFB256E9B2559EF6A922F760AEDA1720818A954D6B0DA61F0E101371649898B8E18DCDEAA4BC7867D600A21D6CD462ACDE99F95672D52FECE228DE493DF0403010001DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 33 | 9F0605A0000005249F2201F8DF0503211231DF060101DF070101DF0803000000DF028180A1F5E1C9BD8650BD43AB6EE56B891EF7459C0A24FA84F9127D1A6C79D4930F6DB1852E2510F18B61CD354DB83A356BD190B88AB8DF04284D02A4204A7B6CB7C5551977A9B36379CA3DE1A08E69F301C95CC1C20506959275F41723DD5D2925290579E5A95B0DF6323FC8E9273D6F849198C4996209166D9BFC973C361CC826E1DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 34 || Rupay |
| 35 | 9F0605A0000005249F2201FADF0503211231DF060101DF070101DF0803000000DF0281A09C6BE5ADB10B4BE3DCE2099B4B210672B89656EBA091204F613ECC623BEDC9C6D77B660E8BAEEA7F7CE30F1B153879A4E36459343D1FE47ACDBD41FCD710030C2BA1D9461597982C6E1BDD08554B726F5EFF7913CE59E79E357295C321E26D0B8BE270A9442345C753E2AA2ACFC9D30850602FE6CAC00C6DDF6B8D9D9B4879B2826B042A07F0E5AE526A3D3C4D22C72B9EAA52EED8893866F866387AC05A1399DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |
| 36 | 9F0605A0000005249F2201FEDF0503211231DF060101DF070101DF0803000000DF02819098F0C770F23864C2E766DF02D1E833DFF4FFE92D696E1642F0A88C5694C6479D16DB1537BFE29E4FDC6E6E8AFD1B0EB7EA0124723C333179BF19E93F10658B2F776E829E87DAEDA9C94A8B3382199A350C077977C97AFF08FD11310AC950A72C3CA5002EF513FCCC286E646E3C5387535D509514B3B326E1234F9CB48C36DDD44B416D23654034A66F403BA511C5EFA3DF040103DF031487F0CD7C0E86F38F89A66F8C47071A8B88586F26 | Rupay |