**WIZARPOS International Co., Ltd.**

**EMV Kernel Interface**

version 4.15

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| version | Date | Description | who |
| 1.00 | 2013-12-20 | create | Michael Li |
| 4.01 | 2018-10-18 | Remove:  emv\_terminal\_param\_set  emv\_terminal\_param\_set2  Add:  emv\_terminal\_param\_set\_tlv | Michael Li |
| 4.02 | 2018-10-22 | Update emv\_aid\_param\_add | Michael Li |
| 4.03 | 2018-10-31 | Add Appendix | Michael Li |
| 4.04 | 2018-11-02 | Update error code | Michael Li |
| 4.05 | 2018-11-09 | Add： 4.16 emv\_get\_kernel\_checksum 4.17 emv\_get\_config\_checksum | Michael Li |
| 4.06 | 2018-12-10 | Add：4.18 emv\_set\_force\_aac | Michael Li |
| 4.07 | 2018-12-12 | 1. Add special application priority for US Common Debit AID(See 5.2:EF07, 5.5:EF06) 2. Add Appendix B | Michael Li |
| 4.08 | 2018-12-25 | Add NSICCS (Indonesia) Support | Michael Li |
| 4.09 | 2019-1-23 | Add emv\_get\_candidate\_list\_tlv | Michael Li |
| 4.10 | 2019-3-11 | Add emv\_set\_kernel\_attr | Michael Li |
| 4.11 | 2019-3-21 | Update emv\_get\_candidate\_list\_tlv | Michael Li |
| 4.12 | 2019-4-4 | Add new error code  ERROR\_APP\_UNSUPPORTED 39 | Michael Li |
| 4.13 | 2019-4-8 | Add：  emv\_generate\_pseudo\_track1；  emv\_generate\_pseudo\_track2 | Michael Li |
| 4.14 | 2019-4-9 | Add 9F6D, 9F6E in emv\_aidparam\_add | Michael Li |
| 4.15 | 2019-4-10 | Add interface :  emv\_get\_kernel\_id  emv\_get\_process\_type | Phill Feng |

1. IC Reader 5

1.1 open reader and wait card 5

1.2 close reader 5

1.3 get current card type 5

1.4 get card ATR 5

1.5 APDU command 5

2. store and set EMV data 6

2.1 check the existence of data for the tag 6

2.2 get the data for the tag 6

2.3 get the data for the tag list 6

2.4 set the data for the tag 7

3. EMV transaction processing 7

3.1 EMVKernel initialize 7

3.2 Initialize EMV transaction data 9

3.3 EMV processing function 9

4. Others functions 9

4.1 Get EMV Kernel version 9

4.2 Set transaction amount 9

4.3 Set other amount 9

4.4 Set transaction type 10

4.5 set emv kernel type 10

4.6 Is needed advice the transaction 10

4.7 Is needed sign the transaction 10

4.8 Set the transaction force online 10

4.9 Read transaction record from the card 11

4.10 Get candidate application list 11

4.11 Get candidate application list with TLV Format 11

4.12 Set the selected index for application selection 11

4.13 Set the result of cardholder ID check 11

4.14 Set the result of Online PIN 12

4.15 Set acceptance for Bypass PIN 12

4.16 Set the result of online authentication 12

4.17 Get Kernel checksum 12

4.18 Get Configuration checksum 13

4.19 Set the transaction Force AAC for first generate AC 13

4.20 Get Pseudo Track1 Data for Amex & Discover Contactless in MSD Mode 13

4.21 Get Pseudo Track2 Data for Amex & Discover Contactless in MSD Mode 13

4.22 Get Contactless Kernel Id 13

4.23 Get EMV Process Type 13

5. EMV parameters 14

5.1 Clear AID info 14

5.2 Add AID info 14

5.3 Clear CAPK info 16

5.4 Add CAPK info 16

5.5 Set EMV terminal parameters by TLV 16

5.6 Clear Exception File 17

5.7 Add Exception File 18

5.8 Clear Revoked Certicates 18

5.9 Add revoked Certificate 18

5.10 Set EMV Kernel additional attribute 18

Annex A：Tag List defined by MasterCard 19

A.1 Contactless Kernel ID 19

A.2 CVM Capability – CVM Required 19

A.3 CVM Capability – No CVM Required 19

A.4 Kernel Configuration 20

A.5 Mag-stripe CVM Capability – CVM Required 20

A.6 Mag-stripe CVM Capability – No CVM Required 20

Annex B：Tag List defined by American Expresspay 21

B.1 Contactless Reader Capabilities 21

B.2 Enhanced Contactless Reader Capabilities 21

Annex C: Self-defined Tag List 23

# 1. IC Reader

1.1 open reader and wait card

/\*

\* @param[in] reader : reader type : 0 all of readers

\* : 1 only contact reader

\* : 2 only contactless reader

\* return value : < 0 Fail

\* >= 0 Success

\* (If select open all of readers, any open success return success)

\*/

**int** **open\_reader**(**int** reader)

1.2 close reader

/\*

\* @param[in] reader： reader type : 0 all of readers

\* : 1 only contact reader

\* : 2 only contactless reader

\*/

**void** **close\_reader**(**int** reader)

1.3 get current card type

/\*

\* return value : 1 contact card

\* : 2 contactless card

\* : -1 no card

\*/

**int** **get\_card\_type**(**void**)

1.4 get card ATR

/\*

\* @param[out] pATR : the value of ATR

\* return value : the length of ATR

\*/

**int** **get\_card\_atr**(unsigned char \*pATR)

1.5 APDU command

/\*

\* @param[in] cmd :APDU command

\* @param[in] cmdLength : the length of APDU command

\* @param[out] respData : the value of card response

\* @param[in] respDataLength : accepted max length of card reasponse

\* return value : >= 0 :the length of card response

\* < 0 :Fail

\*/

**int transmit\_card(** unsigned char **\*cmd,**

**int cmdLength,**

unsigned char **\*respData,**

**int respDataLength)**

# 2. store and set EMV data

2.1 check the existence of data for the tag

/\*

\* @param[in] tag : tag name

\* return value : < 0 the data not exist

\* >= 0 the length of data

\*/

**int** emv\_is\_tag\_present(**int** tag)

2.2 get the data for the tag

/\*

\* @param[in] tag : tag name

\* @param[out] data : the value of the data

\* @param[in] dataLength : accepted max length of the data

\* return value : < 0 : Fail

\* >= 0: the length of the data

\*/

**int** emv\_get\_tag\_data(**int** tag, unsigned char \*data, **int** dataLength)

2.3 get the data for the tag list

/\*

\* @param[in] tagNames : the list of the tags

\* @param[in] tagCount : the count of the tags

\* @param[out] pTagsValue : the values of the data（TLV format）

\* @param[in] pTagsValueLength : accepted max length of the data

\* return value : < 0 : Fail

\* : >= 0: the length of the data

\*/

int emv\_get\_tag\_list\_data(**int** \*tagNames, **int** tagCount,

unsigned char \*pTagsValue,

**int** pTagsValueLength);

2.4 set the data for the tag

/\*

\* @param[in] tag : tag name

\* @param[in] data : the value of the data

\* @param[in] length : the length of the data

\* return value : < 0 : Fail

\* : >= 0 : the tag的长度

\*/

**int** emv\_set\_tag\_data(**int** tag, unsigned char \*data, **int** length)

# 3. EMV transaction processing

3.1 EMVKernel initialize

typedef struct

{

// callback function for card event

CARD\_EVENT\_OCCURED pCafdEventOccured;

// callback function for EVM processing

EMV\_PROCESS\_CALLBACK pEVMProcessCallback;

}EMV\_INIT\_DATA;

**void** **emv\_kernel\_initialize**(unsigned char \*pInitData)

1）typedef void (\*CARD\_EVENT\_OCCURED) (int eventType)

// any card event occured, this function will be revoked

// @param[in] eventType : SMART\_CARD\_EVENT\_INSERT\_CARD = 0;

// : SMART\_CARD\_EVENT\_REMOVE\_CARD = 1;

//: SMART\_CARD\_EVENT\_POWERON\_ERROR = 9;

// :SMART\_CARD\_EVENT\_CONTALESS\_HAVE\_MORE\_CARD = 10;

2）typedef void (\*EMV\_PROCESS\_CALLBACK)(unsigned char \*pData);

// callback function for EVM processing，pData have 2 bytes

// unsigned char status = pData[0]；

// unsigned char desc = pData[1]；

\* status：

\* STATUS\_ERROR = 0; //ERROR

\* STATUS\_CONTINUE = 1; // not completed, need to continue

\* STATUS\_COMPLETION = 2; // completed

\* desc

\* when status = STATUS\_COMPLETION，desc means：

\* APPROVE\_OFFLINE = 1; //Transaction approved Offline

\* APPROVE\_ONLINE = 2; //Transaction approved Online

\* DECLINE\_OFFLINE = 3; //Transaction declined Offline

\* DECLINE\_ONLINE = 4; //Transaction declined Online

\*

\* when status = STATUS\_ERROR，desc means：

\* SUCCESS = 0; //SUCCESS

\* ***ERROR\_NO\_APP*** = 1; //No Supported Application Selected

\* ***ERROR\_CARD\_BLOCKED*** = 2; //card return 6A81 when Application Select

\* ***ERROR\_APP\_SELECT*** = 3; //Error when Application Select

\* ***ERROR\_INIT\_APP*** = 4; //Error when Initialize Application Data

\* ***ERROR\_EXPIRED\_CARD*** = 5; // Card Expired

\* ***ERROR\_APP\_DATA*** = 6; //Error when Read Application Data

\* ***ERROR\_DATA\_INVALID*** = 7; // have invalid data

\* ***ERROR\_DATA\_AUTH*** = 8; // Fail in offline authentication

\* ***ERROR\_GEN\_AC*** = 9; //Generate AC error when Transaction Process

\* ***ERROR\_PROCESS\_CMD*** = 10; //Process Command ERROR

\* ***ERROR\_SERVICE\_NOT\_ALLOWED*** = 11; //Service not Allowed

\* ***ERROR\_PINENTERY\_TIMEOUT*** = 12; //PIN Entry timeout

\* ***ERROR\_OFFLINE\_VERIFY*** = 13; //Check Offline PIN Error when Cardholder Verify

\* ***ERROR\_NEED\_ADVICE*** = 14; //Communication Error with Host, but the card need advice, halted the transaction

\* ***ERROR\_USER\_CANCELLED*** = 15;

\* ***ERROR\_AMOUNT\_OVER\_LIMIT*** = 16; // amount over limit

\* ***ERROR\_AMOUNT\_ZERO*** = 17; // amount can not be zero

\* ***ERROR\_OTHER\_CARD*** = 18； // Please try other card

\* ***ERROR\_MISSING\_DATA*** = 19； //missing mandatory data

\* ***ERROR\_APP\_BLOCKED*** = 20； // application is blocked

\* ***ERROR\_POWER\_ON\_AGAIN*** = 21； // Please power on card again

\* ***ERROR\_CONTACTLESS\_INTERRUPT*** = 22； // contact card inserted when reading contactless card record

\* ***ERROR\_MSD\_NOT\_SUPPORTED*** = 30； // Magstripe Mode not suported

\* ***ERROR\_AMOUNT\_NOT\_PRESENT*** = 31； // amount not present

\* ***ERROR\_CCC*** = 32； // CCC Error for mastercard contactless

\* ***ERROR\_EXCHANGE\_RR\_DATA*** = 33； // Exchange relay resistance data error for mastercard contactless

\* ***ERROR\_GET\_PDOL\_DATA*** = 34； // Get PDOL data error

\* ***ERROR\_RESTART*** = 35； // Please restart the transaction

\* ***ERROR\_SEE\_PHONE*** = 36； // Please see phone

\* ***ERROR\_NEXT\_AID*** = 37； // Please select next aid

\* ***ERROR\_ANOTHER\_INTERFACE*** = 38； // Please try another interface

\* ***ERROR\_APP\_UNSUPPORTED*** = 39； // The app in card is unsupported

\*

\* when status = STATUS\_CONTINUE，desc means：

\* EMV\_CANDIDATE\_LIST = 1; //notify Application show Application Candidate List

\* EMV\_APP\_SELECTED = 2; //Application Select Completed

\* EMV\_READ\_APP\_DATA = 3; //Read Application Data Completed

\* EMV\_DATA\_AUTH = 4; //Data Authentication Completed

\* EMV\_OFFLINE\_PIN = 5; // notify Application prompt Caldholder enter offline PIN,

\* EMV\_ONLINE\_ENC\_PIN = 6; //notify Application prompt Caldholder enter Online PIN

\* EMV\_PIN\_BYPASS\_CONFIRM = 7; //notify Application confirm to Accepted PIN Bypass or not

\* EMV\_PROCESS\_ONLINE = 8; //notify Application to Process Online

\* EMV\_ID\_CHECK = 9; //notify Application Check Cardholder's Identification

\*/

3.2 Initialize EMV transaction data

**void** **emv\_trans\_initialize**(**void**)

3.3 EMV processing function

**/\***

**\* return value： >=0 SUCCESS, <0 Fail**

**\*/**

**int** **emv\_process\_next**(**void**)

# 4. Others functions

4.1 Get EMV Kernel version

**/\*\***

**\* @param[out] buffer： the value of emv kernel version**

**\* @param[in] bufferLength：accepted max length of emv kernel version**

**\* return value： the length of emv kernel verion**

**\*/**

**int** **emv\_get\_version\_string**(unsigned char \*buffer, **int** bufferLength)

4.2 Set transaction amount

/\*\*

\* @param[in] amount： '\0' as ending mark

\* return value： >=0 Success; < 0 Fail

\* If strlen(amount) > 12, return -1

\*/

**int** **emv\_set\_trans\_amount**(unsigned char \*amount)

4.3 Set other amount

/\*\*

\* @param[in] amount： '\0' as ending mark

\* return value： >=0 Success; < 0 Fail

\* If strlen(amount) > 12, return -1

\*/

**int** **emv\_set\_other\_amount**(unsigned char \*amount)

4.4 Set transaction type

**int** **emv\_set\_trans\_type**(unsigned char transType)

#define TRANS\_GOODS\_SERVICE 0x00

#define TRANS\_CASH 0x01

#define TRANS\_INQUIRY 0x04

#define TRANS\_TRANSFER 0x05

#define TRANS\_PAYMENT 0x06

#define TRANS\_ADMIN 0x07

#define TRANS\_CASHBACK 0x09

#define TRANS\_CARD\_RECORD 0x0A

4.5 set emv kernel type

/\*\*

\* @param[in] kernelType： **1** *EMV Contact Kernel*

\*  **2 EMV Contactless Kernel**

\* 3 *UPCASH Kernel for China Union Pay*

\*/

**int** **emv\_set\_kernel\_type**(unsigned char kernelType)

4.6 Is needed advice the transaction

**/\*\***

**\*** return value： **1 need advice**

**\* 0 not need advice**

**\*/**

**int** **emv\_is\_need\_advice**(**void**)

4.7 Is needed sign the transaction

**/\*\***

**\*** return value： **1 need sign**

**\* 0 not need sign**

**\*/**

**int** **emv\_is\_need\_signature**(**void**)

4.8 Set the transaction force online

/\*\*

\* @param[in] flag： flag=1 Yes， flag = 0 No

\*/

**int** **emv\_set\_force\_online**(**int** flag)

4.9 Read transaction record from the card

**/\*\***

**\* @param[out] data : transaction record**

**\* @param[in] dataLength : accepted max length for the transaction record**

**\* return value : < 0 : Fail**

**\* : >= 0: record count**

**\*/**

**int** **emv\_get\_card\_record**(uint8\_t \*data, **int** dataLength)

4.10 Get candidate application list

**/\***

**\* @param[out] data : application list as "LV" format**

**\* @param[in] dataLength : accepted max length for application list**

**\* return value : < 0 : Fail**

**\* : >= 0: application count**

**\*/**

**int emv\_get\_candidate\_list**(uint8\_t \*data, **int** dataLength)

4.11 Get candidate application list with TLV Format

**/\***

**\* @param[out] data : application list with "TLV" format**

**\* Tag 4F: AID, It is the start of candidate record**

**\* Tag 9F11: Issuer Code Table Index**

**\* Tag 50: Application Label**

**\* Tag 9F12: Application Preferred Name**

**\* @param[in] dataLength : accepted max length for application list**

**\* return value : < 0 : Fail**

**\* : >= 0: the length of data**

**\*/**

**int emv\_get\_candidate\_list\_tlv**(uint8\_t \*data, **int** dataLength)

4.12 Set the selected index for application selection

**/\*\***

**\* @param[in] index : the selected index (started by 0)**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_set\_candidate\_list\_result**(**int** index)

4.13 Set the result of cardholder ID check

/\* ID Type（9F62）、ID Number(9F61)

**\* @param[in] result : 0: check Fail，1:check success**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_set\_id\_check\_result**(**int** result)

4.14 Set the result of Online PIN

**/\*\***

**\* @param[in] result : 0: Online PIN not input，1:Online PIN inputted**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_set\_online\_pin\_entered**(**int** result)

4.15 Set acceptance for Bypass PIN

**/\*\***

**\* @param[in] result : 0: refused bypass pin**

**1: accepted bypass pin**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_set\_pin\_bypass\_confirmed**(**int** result)

4.16 Set the result of online authentication

/\*\*

\* @param[in] result : -1:communication failed；0: host refused；1: host accepted

\* @param[in] respCode : 2 bytes response code from the host

\* @param[in] issuerRespData : the emv data from the host

\* @param[in] issuerRespDataLength : the length of the emv data from the host

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_set\_online\_result**(**int** result,

unsigned char \*respCode,

unsigned char \*issuerRespData,

**int** issuerRespDataLength)

4.17 Get Kernel checksum

**/\*\***

**\* @param[out] buffer： the value of emv kernel checksum**

**\* @param[in] bufferLength：accepted max length**

**\* return value： the length of kernel checksum**

**\*/**

**int** **emv\_get\_kernel\_checksum**(unsigned char \*buffer, **int** bufferLength)

4.18 Get Configuration checksum

**/\*\***

**\* @param[out] buffer： the value of configuration checksum**

**\* @param[in] bufferLength：accepted max length**

**\* return value： the length of configuration checksum**

**\*/**

**int** **emv\_get\_config\_checksum**(unsigned char \*buffer, **int** bufferLength)

4.19 Set the transaction Force AAC for first generate AC

/\*\*

\* @param[in] flag： flag=1 Yes， flag = 0 No

\*/

**int** **emv\_set\_force\_aac**(**int** flag)

4.20 Get Pseudo Track1 Data for Amex & Discover Contactless in MSD Mode

**/\*\***

**\* @param[out] data： the value of track1 data**

**\* @param[in] dataLength：accepted max length**

**\* return value： the length of track1 data**

**\*/**

**int** **emv\_generate\_pseudo\_track1**(**byte[]** data, int dataLength)

4.21 Get Pseudo Track2 Data for Amex & Discover Contactless in MSD Mode

**/\*\***

**\* @param[out] data： the value of track2 data**

**\* @param[in] dataLength：accepted max length**

**\* return value： the length of track2 data**

**\*/**

**int** **emv\_generate\_pseudo\_track2**(**byte[]** data, int dataLength)

4.22 Get Contactless Kernel Id

**/\*\***

**\* get contactless kernel id**

**\* @return 0-contact 2-MasterCard 3-VISA 4-AMEX**

**\* 5-JCB 6-DISCOVER 7-CUP 8-GEMALTO PURE**

**\*/**

**int emv\_get\_kernel\_id()**

4.23 Get EMV Process Type

**/\*\***

**\* get process type of contactless card**

**\* @return 2-EMV mode**

**\* 3-MSD mode**

**\*/**

**int emv\_get\_process\_type()**

# 5. EMV parameters

5.1 Clear AID info

/\*\*

\* return value： >=0: Success; < 0: Fail

\*/

**int** **emv\_aidparam\_clear**(**void**)

5.2 Add AID info

**/\***

**\* @param[in] data : see form below，format is TLV**

**\* @param[in] dataLength : the length of the data**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_aidparam\_add**( uint8\_t \*data, **int** dataLength)

| name | Format | length（byte） | tag |
| --- | --- | --- | --- |
| AID | b | 5－16 | 9F06 |
| Application selection Indicator（ASI） | b | 1 | DF01 |
| Application version number | b | 2 | 9F08 |
| TAC－Default | b | 5 | DF11 |
| TAC－Online | b | 5 | DF12 |
| TAC－Denial | b | 5 | DF13 |
| Terminal floor limit | b | 4 | 9F1B |
| Threshold value for Biased Random Selection | b | 4 | DF15 |
| Maximum Target Percentage to be used for Biased Random Selection | cn | 1 | DF16 |
| Target Percentage to be used for Random Selection | cn | 1 | DF17 |
| Default DDOL | b | Var. | DF14 |
| Ability for Online PIN | b | 1 | DF18 |
| Application Label | an | 1-16 | 50 |
| Application Preferred Name | an | 1-16 | 9F12 |
| Application Priority Indicator | b | 1 | 87 |
| Merchant Identifier | an | 15 | 9F16 |
| Acquirer Identifier | n | 6-11 | 9F01 |
| MCC | n | 4 | 9F15 |
| Transaction Reference Currency Code | n | 3 | 9F3C |
| Transaction Reference Currency Exponent | n | 1 | 9F3D |
| Default TDOL | b | Var. | DF22 |
| Contactless Floor Limit | n | 6 | DF19 |
| Contactless Limit | n | 6 | DF20 |
| CVM Limit | n | 6 | DF21 |
| Contactless Kernel ID (See A.1) | n | 1 | DF810C |
| C2: CVM Capability – CVM Required (See A.2) | b | 1 | DF8118 |
| C2: CVM Capability – No CVM Required (See A.3) | b | 1 | DF8119 |
| C2: kernel configuration (See A.4) | b | 2 | DF811B |
| C2: Mag-stripe CVM Capability – CVM Required (See A.5) | b | 1 | DF811E |
| C2: Reader Contactless transaction limit (No On-device CVM) | n | 6 | DF8124 |
| C2: Reader Contactless transaction limit (On-device CVM) | n | 6 | DF8125 |
| C2: Mag-stripe CVM Capability – No CVM Required (See A.6) | b | 1 | DF812C |
| C4:Contactless Reader Capabilities | b | 1 | 9F6D |
| C4:Enhanced Contactless Reader Capabilities | b | 4 | 9F6E |
| Is US Common Debit AID  0 – No; 1 - Yes | n | 1 | EF07 |
| Is apply to NSICCS (Indonesia)  0 - No; 1 - Yes | n | 1 | EF08 |

\* C2 – Only for Mastercard MCL

\* C4 – Only for American Expresspay

5.3 Clear CAPK info

/\*\*

\* return value： >=0 Success; < 0 Fail

\*/

**int** **emv\_capkparam\_clear**(**void**)

5.4 Add CAPK info

**/\***

**\* @param[in] data : see form below，format is TLV**

**\* @param[in] dataLength : the length of the data**

**\* return value : < 0 : Fail**

**\* : >= 0: Success**

**\*/**

**int** **emv\_capkparam\_add**( uint8\_t \*data, **int** dataLength)

| Name | Format | length（byte） | tag |
| --- | --- | --- | --- |
| RID | b | 5 | 9F06 |
| Certification Authority Public Key Index | b | 1 | 9F22 |
| Certification Authority Public Key Expiration Date | n8 | 8 | DF05 |
| Certification Authority Public Key hash Algorithm Indicator | b | 1 | DF06 |
| Certification Authority Public Key Algorithm Indicator | b | 1 | DF07 |
| Certification Authority Public Key Modulus | b | Var. | DF02 |
| Certification Authority Public Key Exponent | b | 1 or 3 | DF04 |
| Certification Authority Public Key Checksum | b | Var. | DF03 |

5.5 Set EMV terminal parameters by TLV

|  |  |
| --- | --- |
| Supported Tag | Description |
| 5F2A | Transaction Currency Code |
| 5F36 | Transaction Currency Exponent |
| 9F16 | Merchant Identification |
| 9F1A | Terminal Country Code |
| 9F1C | Terminal Identification |
| 9F1E | IFD Serial Number |
| 9F33 | Terminal Capabilities |
| 9F35 | Terminal Type |
| 9F40 | Additional Terminal Capabilities |
| 9F4E | Merchant Name and Location |
| 9F66 | TTQ first byte |
| DF19 | Contactless floor limit |
| DF20 | Contactless transaction limit |
| DF21 | CVM limit |
| DF8104 | Balance Read Before Gen AC (C2) |
| DF8105 | Balance Read After Gen AC (C2) |
| DF811C | Max Lifetime of Torn Transaction Log Record (C2) |
| DF811D | Max Number of Torn Transaction Log Records (C2) |
| DF812D | Message Hold Time (C2) |
| DF8132 | Minimum Relay Resistance Grace Period (C2) |
| DF8133 | Maximum Relay Resistance Grace Period (C2) |
| DF8134 | Terminal Expected Transmission Time For Relay Resistance C-APDU (C2) |
| DF8135 | Terminal Expected Transmission Time For Relay Resistance R-APDU (C2) |
| DF8136 | Relay Resistance Accuracy Threshold (C2) |
| DF8137 | Relay Resistance Transmission Time Mismatch Threshold (C2) |
| EF01 | Status check support: 0 – No; 1 – Support |
| EF02 | Zero check support: 0 – No; 1 – Support |
| EF03 | Authorization Type For American Expresspay(C4): 0-Immediate; 1-Delayed |
| EF04 | CDCVM support: 0 – No; 1 – Support |
| EF05 | Extended Selection: 0 – No; 1 – Support |
| EF06 | Priority of US Common Debit AID：  0 – The priority of US Common Debit AID is lower than Global AID;  1 – The priority of US Common Debit AID is higher than Global AID |

**int** **emv\_terminal\_param\_set\_tlv**( uint8\_t \*data, int dataLength)

5.6 Clear Exception File

/\*\*

\* return value： >=0 Success; < 0 Fail

\*/

**int** **emv\_exception\_file\_clear**(**void**)

5.7 Add Exception File

Typedef struct{

unsigned char cardNo[19]; // PAN

unsigned char panSequence; // PAN Sequence Number

}ExceptionFile

**int** **emv\_exception\_file\_add**( unsigned char \*exceptFile)

5.8 Clear Revoked Certicates

/\*\*

\* return value： >=0 Success; < 0 Fail

\*/

**int** **emv\_revoked\_cert\_clear**(**void**)

5.9 Add revoked Certificate

Typedef struct{

unsigned char rid[5];

unsigned char capki;

}RevokedCert

**int** **emv\_revoked\_cert\_add**( uint8\_t \*revokedCert)

5.10 Set EMV Kernel additional attribute

/\* param data is less or equal 2 bytes,

\*

Byte 1:

bit 8 Enable auto perform UPCASH for contact card.

bit 7 Force select CUP application.

bit 6 Force check app version in FDDA for CUP contactless.

bit 5 Force online with Cash & CashBack for Visa contacltess.

bit 4 Subsequent Bypass PIN entry

bit 3 Disable PayWave AUC check.

bit 2 RFU

bit 1 RFU

Byte 2:

bit 8 Enable contactless AID select.

bit 7 RFU

bit 6 RFU

bit 5 RFU

bit 4 RFU

bit 3 RFU

bit 2 RFU

bit 1 RFU

\*/

**int** **emv\_set\_kernel\_attr(byte[] data, int dataLength)**

# Annex A：Tag List defined by MasterCard

A.1 Contactless Kernel ID

Tag: 'DF810C'

Length: 1

Format: b

Description: Indicates the kernel type of contactless application

2 = Kernel 2 for MasterCard AIDs

3 = Kernel 3 for Visa AIDs

4 = Kernel 4 for American Express AIDs

5 = Kernel 5 for JCB AIDs

6 = Kernel 6 for Discover AIDs

7 = Kernel 7 for UnionPay AIDs

8 = Kernel for PURE contactless Reader

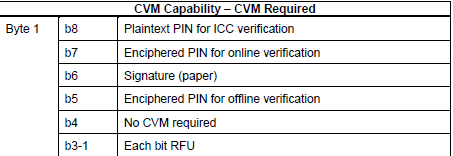
A.2 CVM Capability – CVM Required

Tag: 'DF8118'

Length: 1

Format: b

Description: Indicates the CVM capability of the Terminal and Reader when the transaction amount is greater than the *Reader CVM Required Limit*.



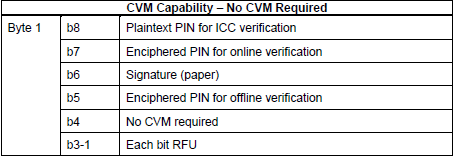
A.3 CVM Capability – No CVM Required

Tag: 'DF8119'

Length: 1

Format: b

Description: Indicates the CVM capability of the Terminal and Reader when the transaction amount is less than or equal to the *Reader CVM Required Limit*.



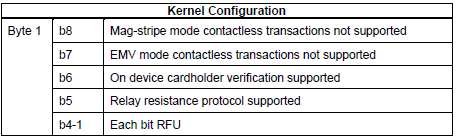
A.4 Kernel Configuration

Tag: 'DF811B'

Length: 1

Format: b

Description: Indicates the Kernel configuration options.



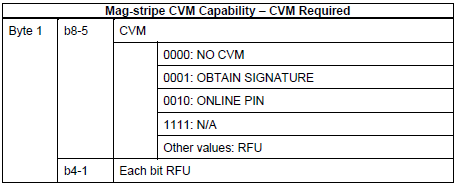
A.5 Mag-stripe CVM Capability – CVM Required

Tag: 'DF811E'

Length: 1

Format: b

Description: Indicates the CVM capability of the Terminal/Reader in the case of a mag-stripe mode transaction when the *Amount, Authorized (Numeric)* is greater than the *Reader CVM Required Limit*.



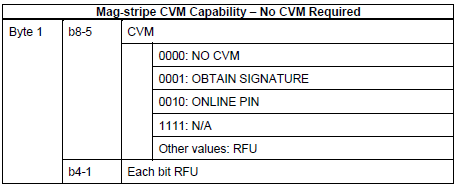
A.6 Mag-stripe CVM Capability – No CVM Required

Tag: 'DF812C

Length: 1

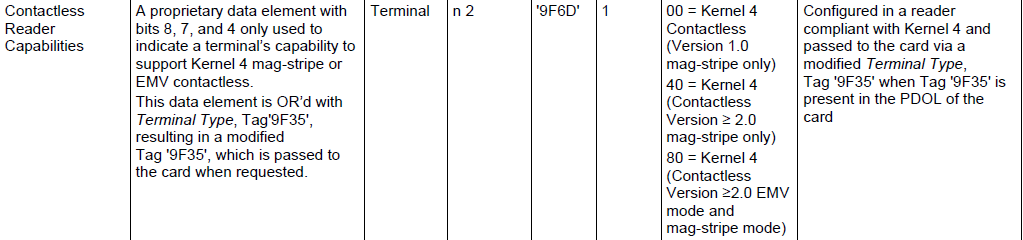
Format: b

Description: Indicates the CVM capability of the Terminal/Reader in the case of a mag-stripe mode transaction when the *Amount, Authorized (Numeric)* is less than or equal to the *Reader CVM Required Limit*.

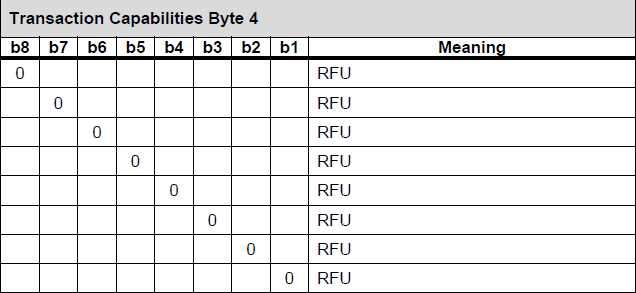
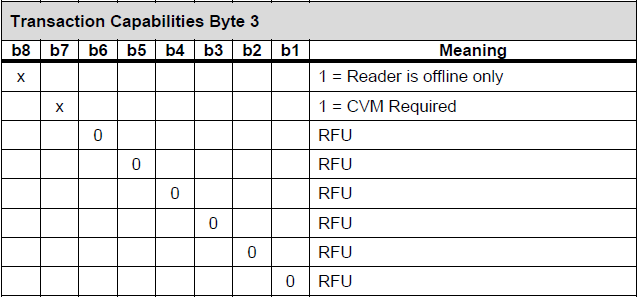
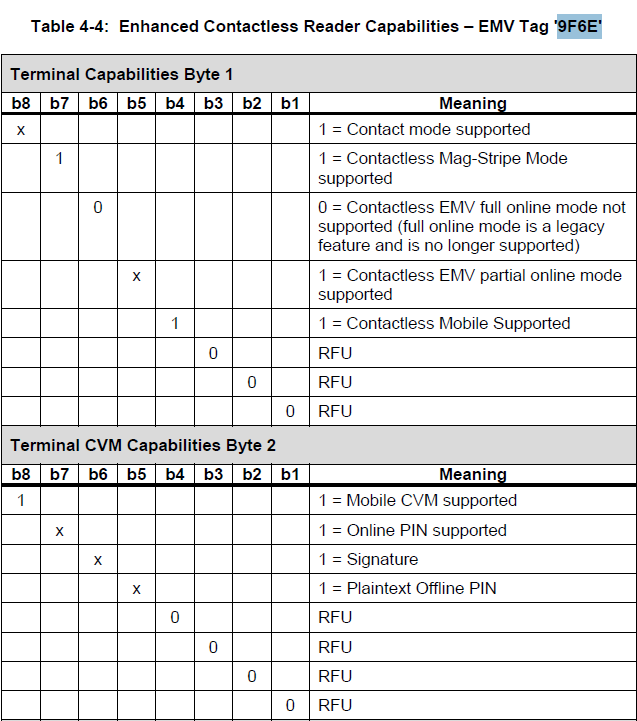
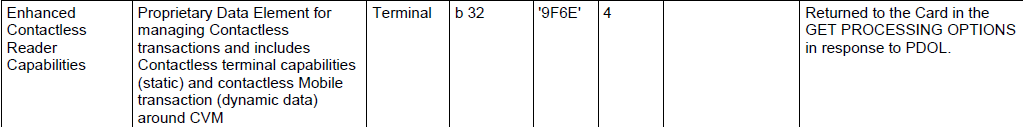


# Annex B：Tag List defined by American Expresspay

B.1 Contactless Reader Capabilities



B.2 Enhanced Contactless Reader Capabilities



# Annex C: Self-defined Tag List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag | Name | Format | Length | Description |
| EF01 | Status check support | n | 1 | [Terminal Parameter] 0 – No; 1 – Support |
| EF02 | Zero check support | n | 1 | [Terminal Parameter] 0 – No; 1 – Support |
| EF03 | Authorization Type For American Expresspay(C4) | n | 1 | [Terminal Parameter] 0-Immediate; 1-Delayed |
| EF04 | CDCVM support | n | 1 | [Terminal Parameter] 0 – No; 1 – Support |
| EF05 | Extended Selection | n | 1 | [Terminal Parameter] 0 – No; 1 – Support |
| EF06 | Priority of US Common Debit AID | n | 1 | [Terminal Parameter]  0 – The priority of US Common Debit AID is lower than Global AID;  1 – The priority of US Common Debit AID is higher than Global AID |
| EF07 | Is US Common Debit AID | n | 1 | [AID Parameter] 0 – No; 1 – Yes |
| EF08 | Is apply to NSICCS (Indonesia) | n | 1 | [AID Parameter] 0 - No; 1 - Yes, used for Bank Indonesia |