EMV Kernel Interface

version 4.31

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	2019-4-24	Add DF11, DF12, DF13 in emv_terminal_param_set_tlv	Michael Li
4.16	2019-4-26	Add 4.22 emv_offline_pin_verified	Michael Li
4.17	2019-4-26	Add 4.23 emv_log_file_clear	Michael Li
4.18	2019-5-5	Add 4.24 emv_get_offlinepin_times	Michael Li
4.19	2019-5-8	Update 4.15 Set acceptance for Bypass PIN	Michael Li
4.20	2019-6-11	Add 1.6 set_contactless_detach_enable	Michael Li
4.21	2019-11-21	Add ‘Interac’, ‘MIR’, ‘WISE’ Support	Michael Li
4.22	2019-12-24	Add Error Code	Michael Li

4.23	2020-3-9	Add 'C5_JCB' Support	Michael Li
4.24	2020-8-24	Update 5.10 kernel additional attribute	Michael Li
4.25	2020-8-28	Add additional attribute – 'DISABLE LED'	Michael Li
4.26	2020-12-22	1. Add 'EF10' (Combination Options) in AID parameters for JCB; 2. Add Pure Support; 3. Add Dynamic Reader Limits Support	Michael Li
4.26	2020-12-31	Update Annex B1	Michael Li
4.27	2021-4-16	1. Removed EF03 for C4; 2. Update Annex B2.	Michael Li
4.28	2021-6-22	Added attribution for log output & transit terminal	Michael Li
4.29	2021-09-17	Added terminal parameter EF11~EF14 for TRM	Michael Li
4.30	2022-02-21	1. Added 6. The Functions for Offline PIN 2. Added kernel Attribute B2b1, B3b8	Michael Li
4.31	2023-05-17	1. Added kernel Attribute B3b7, B3b6	Michael Li

1. IC READER 7

1.1 OPEN READER AND WAIT CARD 7

1.2 CLOSE READER 7

1.3 GET CURRENT CARD TYPE 7

1.4 GET CARD ATR 7

1.5 APDU COMMAND 7

1.6 SET DETACH ATTRIBUTION OF CONTACTLESS CARD FOR READER CLOSING 8

2. STORE AND SET EMV DATA 8

2.1 CHECK THE EXISTENCE OF DATA FOR THE TAG 8

2.2 GET THE DATA FOR THE TAG 9

2.3 GET THE DATA FOR THE TAG LIST 9

2.4 SET THE DATA FOR THE TAG 9

3. EMV TRANSACTION PROCESSING 9

3.1 EMVKERNEL INITIALIZE 9

3.2 INITIALIZE EMV TRANSACTION DATA 12

3.3 EMV PROCESSING FUNCTION 12

4. OTHER FUNCTIONS 12

4.1 GET EMV KERNEL VERSION 12

4.2 SET TRANSACTION AMOUNT 12

4.3 SET OTHER AMOUNT 13

4.4 SET TRANSACTION TYPE 13

4.5 SET EMV KERNEL TYPE 13

4.6 IS NEEDED ADVICE THE TRANSACTION 13

4.7 IS NEEDED SIGN THE TRANSACTION 13

4.8 SET THE TRANSACTION FORCE ONLINE 14

4.9 READ TRANSACTION RECORD FROM THE CARD 14

4.10 GET CANDIDATE APPLICATION LIST 14

4.11 GET CANDIDATE APPLICATION LIST WITH TLV FORMAT 14

4.12 SET THE SELECTED INDEX FOR APPLICATION SELECTION 14

4.13 SET THE RESULT OF CARDHOLDER ID CHECK 15

4.14 SET THE RESULT OF ONLINE PIN 15

4.15 SET ACCEPTANCE FOR BYPASS PIN 15

4.16 SET THE RESULT OF ONLINE AUTHENTICATION 15

4.17 GET KERNEL CHECKSUM 15

4.18 GET CONFIGURATION CHECKSUM 16
4.19 SET THE TRANSACTION FORCE AAC FOR FIRST GENERATE AC 16
4.20 GET PSEUDO TRACK1 DATA FOR AMEX & DISCOVER CONTACTLESS IN MSD MODE 16
4.21 GET PSEUDO TRACK2 DATA FOR AMEX & DISCOVER CONTACTLESS IN MSD MODE 16
4.22 CLEAR EMV TRANSACTION LOG 16
4.23 GET CURRENT PROCESS TYPE 16
4.24 GET TAP FLAG 17
4.25 SET APPLICATION AUTHENTICATION TRANSACTION FOR PURE CARD 17
4.26 GET ATOL DATA FOR PURE CARD 17
4.27 GET ATDTOL DATA FOR PURE CARD 17

5. EMV PARAMETERS 18

5.1 CLEAR AID INFO 18
5.2 ADD AID INFO 18
5.3 CLEAR CAPK INFO 20
5.4 ADD CAPK INFO 20
5.5 SET EMV TERMINAL PARAMETERS BY TLV 21
5.6 CLEAR EXCEPTION FILE 22
5.7 ADD EXCEPTION FILE 22
5.8 CLEAR REVOKED CERTICATES 22
5.9 ADD REVOKED CERTIFICATE 23
5.10 SET EMV KERNEL ADDITIONAL ATTRIBUTE 23
5.11 SET DYNAMIC READER LIMITS 23

6. THE FUNCTIONS FOR OFFLINE PIN 24

6.1 IS OFFLINE PIN VERIFIED 24
6.2 GET THE TIMES OF OFFLINE-PIN ENTRY 24
6.3 SET CUSTOM PINPAD GUI FOR OFFLINE PIN 24
6.4 SET CUSTOM PINPAD TITLE FOR OFFLINE PIN 25
6.5 SET CUSTOM PINPAD PROMPT FOR OFFLINE PIN 25

ANNEX A：TAG LIST DEFINED BY MASTERCARD 25

A.1 CONTACTLESS KERNEL ID (DF810C) 25
A.2 CVM CAPABILITY – CVM REQUIRED (DF8118) 26
A.3 CVM CAPABILITY – NO CVM REQUIRED (DF8119) 27
A.4 KERNEL CONFIGURATION (DF811B) 27
A.5 MAG-STRIPE CVM CAPABILITY – CVM REQUIRED (DF811E) 28

A.6 MAG-STRIPE CVM CAPABILITY – NO CVM REQUIRED (DF812C) 28

ANNEX B：TAG LIST DEFINED BY AMERICAN EXPRESSPAY 29

B.1 CONTACTLESS READER CAPABILITIES (9F6D) 29

B.2 ENHANCED CONTACTLESS READER CAPABILITIES (9F6E) 30

ANNEX C：TAG LIST DEFINED BY JCB 32

C.1 TERMINAL INTERCHANGE PROFILE (9F53) 32

ANNEX D：TAG LIST DEFINED BY INTERAC 33

D.1 MERCHANT TYPE INDICATOR 33

D.2 TERMINAL TRANSACTION INFORMATION (TTI) 33

D.3 TERMINAL OPTION STATUS (TOS) 34

ANNEX E：TAG LIST DEFINED BY MIR 35

E.1 TERMINAL TPM CAPABILITIES (DF55) 35

ANNEX F: SELF-DEFINED TAG LIST 36

F.1 COMBINATION OPTIONS (EF10) 37

F.2 CONTACTLESS APPLICATION CAPABILITY (EF20) 37

F.3 CONTACTLESS POS IMPLEMENTATION OPTIONS (EF21) 39

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)
```

1.6 Set detach attribution of contactless card for reader closing

```
    /**
     * @param enable
     *    0 - don't detach card when close contactless reader;
     *    1 - waiting detach card when close contactless reader
     *    2 - do not control
     **/
    void set_contactless_detach_enable(int enable)
```

1.7 Get Contactless card Type

```
    /**
     * Get contactless card type
     * @return   CONTACTLESS_CARD_TYPE_A_CPU                0x0000
     *           CONTACTLESS_CARD_TYPE_B_CPU                0x0100
     *           CONTACTLESS_CARD_TYPE_A_CLASSIC_MINI       0x0001
     *           CONTACTLESS_CARD_TYPE_A_CLASSIC_1K         0x0002
     *           CONTACTLESS_CARD_TYPE_A_CLASSIC_4K         0x0003
     *           CONTACTLESS_CARD_TYPE_A_UL_64              0x0004
     *           CONTACTLESS_CARD_TYPE_A_UL_192             0x0005
     *           CONTACTLESS_CARD_TYPE_A_MP_2K_SL1          0x0006
     *           CONTACTLESS_CARD_TYPE_A_MP_4K_SL1          0x0007
     *           CONTACTLESS_CARD_TYPE_A_MP_2K_SL2          0x0008
     *           CONTACTLESS_CARD_TYPE_A_MP_4K_SL2          0x0009
     *           CONTACTLESS_CARD_UNKNOWN                   0x00FF
     */
    public native static int get_contactless_card_type()
```

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
* ***ERROR_PERFORM_TRANSACTION*** = 40; // Perform transaction bad sw
* ***ERROR_RECOVERY_NOT_SUPPORT*** = 41; // Tearing recovery not supported
* ***ERROR_RECOVERY_LIMIT_EXCEEDED*** = 42; // Recovery limit exceeded
* ***ERROR_SEE_ATTENDANT*** = 43
* ***ERROR_CANNOT_PROCESSED*** = 44
* ***ERROR_PAYMENT_NOT_ACCEPTED*** = 45
* ***ERROR_TOO_MANY_TAPS*** = 46
* **ERROR_CONDITIONS_NOT_SATISFIED** = 47
* // Used for MIR
* **ERROR_COMPLETE_RECOVERY_NOT_SUPPORT** = 48
* **ERROR_READ_RECORD_RECOVERY_NOT_SUPPORT** = 49
* **ERROR_PERFORM_RECOVERY_LIMIT_EXCEEDED** = 50
* **ERROR_COMPLETE_RECOVERY_LIMIT_EXCEEDED** = 51
* **ERROR_READ_RECORD_RECOVERY_LIMIT_EXCEEDED** = 52
* **ERROR_GEN_AC_BAD_CID** = 53 /** Generate AC get bad CID Value */
* // used for PURE
* **ERROR_REACTIVE_AFTER_ONLINE** = 54
* **ERROR_GET_PUT_DATA_ELEMENTS** = 55
* **ERROR_APP_AUTH** = 56
* **ERROR_UNKNOWN_PAYMENT_RESULT** = 57
* **ERROR_INIT_APP_AAC** = 58
* **ERROR_DE_NOT_SUPPORTED** = 59

```
*
*    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
*          EMV_2TAP= 16; // notify tap card again for Rupay contactless card
*/
```

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. Other 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] flag:  0: Disable bypass pin
                    1: Enable bypass pin
* return value      : < 0 : Fail
*                   : >= 0: Success
*/
int emv_set_bypass_pin(int flag)
```

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 Clear EMV transaction log

```
int emv_log_file_clear()
```

4.23 Get current process type

```
/*
* return value  : 1    EMV (Contact) Mode
```

```
*                : 2    EMV (contactless) Mode
*                : 3    Mag-stripe Mode
*/
int emv_get_process_type()
```

4.24 Get Tap Flag

```
/**
* get tap flag for after online authentication
* @return    0 - no additional tap
*                1 - long tap
*                2 - another tap
*/
int emv_get_tap_flag()
```

4.25 Set Application Authentication transaction for PURE Card

```
/**
* Pure set transaction to application authentication
* @param flag    1 - indicate this transaction is 'application authentication'
*                    0 - indicate this transaction is normal transaction
 */
void emv_pure_set_auth_trans(uint8_t flag)
```

4.26 get ATOL data for PURE Card

```
/**
 * Pure get atol data
 * @param pTagsValue buffer to store TLVs
 * @param tagsValueLength buffer size of pTagsValue
 * @return real length
 */
int emv_pure_get_atol_data(unsigned char *pTagsValue, int tagsValueLength)
```

4.27 Get ATDTOL data for PURE Card

```
/**
 * Pure get atdtol data
 * @param pTagsValue buffer to store TLVs
 * @param tagsValueLength buffer size of pTagsValue
 * @return real length
 */
int emv_pure_get_atdtol_data (unsigned char *pTagsValue, int tagsValueLength)
```

5. EMV parameters

5.1 Clear AID info

```
/**
* return value:  >=0: Success; < 0: Fail
*/
int emv_aidparam_clear(void)

int emv_contactless_aidparam_clear(void) /** It is especially for Contactless aid */
```

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)

int emv_contactless_aidparam_add( uint8_t *data, int dataLength) /** It is especially for Contactless aid */
```

name	Format	Length(byte)	tag
AID	b	5—16	9F06
Application selection Indicator（ASI）: Indicates whether the associated AID in the terminal must match the AID in the card exactly 0 – No; 1 - Yes	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

name	Format	Length(byte)	tag
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
Terminal Risk Management Data	b	Var.	9F1D
TTQ first byte	b	1	9F66
Account Type 00: Default – unspecified; 10: Savings; 20: Cheque/debit; 30: Credit	b	1	5F57
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	1	DF811B
C2: Mag-stripe CVM Capability – CVM Required (See A.5)	b	1	DF811E
[C2 & MIR]: Reader Contactless transaction limit (No On-device CVM)	n	6	DF8124
[C2 & MIR]:Reader Contactless transaction limit (On-device CVM)	n	6	DF8125
C2: Mag-stripe CVM Capability – No CVM Required (See A.6)	b	1	DF812C

name	Format	Length(byte)	tag
C2: Mag-stripe Application Version Number (Reader)	b	2	9F6D
C4: Contactless Reader Capabilities	b	1	9F6D
C4: Enhanced Contactless Reader Capabilities	b	4	9F6E
C5: Terminal Interchange Profile	b	3	9F53
C5: Combination Options	b	2	EF10
Is US Common Debit AID 0 – No; 1 - Yes	n	1	EF07
MIR: Terminal TPM Capabilities	b	2	DF55
MIR: Transaction Recovery Limit	b	1	DF56
MIR: Data Exchange Tag List	b	Var.	FF04
Is apply to NSICCS (Indonesia) 0 - No; 1 - Yes	n	1	EF08
Interac: Merchant Type Indicator(MTI)	n	1	9F58
Interac: Terminal transaction Information(TTI)	b	3	9F59
Interac: Contactless Receipt Required Limit	n	6	9F5D
Interac: Terminal Option Status(TOS)	b	2	9F5E
Interac: Interac Retry Limit	n	1	EF09
Pure: ATOL	b	Max 30	EF22
Pure: MTOL	b	Max 50	EF23
Pure: ATDTOL	b	Max 40	EF24

```
* C2 – Only for Mastercard MCL
* C4 – Only for American Expresspay
* MIR – Only for MIR
* Interac – Only for Interac
* Pure – Only for PURE
```

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
9F1B	Terminal Floor Limit
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
DF11	TAC—Default
DF12	TAC—Online
DF13	TAC—Denial
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
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
EF10	Combination Options(C5)
EF11	Floor limit checking support
EF12	Random Transaction Selection support
EF13	Velocity Checking support
EF14	Exception File support
EF20	Pure Contactless Application Capability
EF21	Pure Contactlesss POS Implementation Options

```
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 Transit Terminal
 bit 1 RFU

 Byte 2:
 bit 8 Enable contactless AID select.
 bit 7 Enable online CDA for Interac.
 bit 6 Disable Issuer Script for Visa Contactless
 bit 5 Disable Issuer Script for Discover Contactless
 bit 4 Disable LED
 bit 3 Disable log output
 bit 2 Disable sensitive log output
 bit 1 Force App selected callback For Contactless card

 Byte 3:
 bit 8 Exit Offline Pin instantly if card removed
 bit 7 Callback EMV_CANDIDATE_LIST if the contactless card have
multi-app
 bit 6 Don't set CVM Result if cvm method is CDCVM for Visa &
UnionPay contactless
 */
int emv_set_kernel_attr(byte[] data, int dataLength)
```

5.11 Set Dynamic Reader Limits

```
int emv_terminal_param_set_drl(byte[] data, int dataLength)
```

Data	Length
Enable/Disable Dynamic Reader Limits	1
DRL_PARAM drl1[Optional]	41
DRL_PARAM drl2[Optional]	41
DRL_PARAM drl3[Optional]	41
DRL_PARAM drl4[Optional]	41

The max number of DRL_PARAM(Dynamic Reader limits) is 8

The Structure of DRL_PARAM:

DRL_PARAM	Length
Enable/Disable Flag	1
Program ID	16
Program ID Length	1
Status Check Support(0:Disable, 1-Enable)	1
Zero Check(0:Disable, 1-Option 1,2-Option 2)	1
Enable/Disable Contactless transaction Limit	1
Contactless transaction Limit	6
Enable/Disable Contactless Floor Limit	1
Contactless Floor Limit	6
Enable/Disable Contactless CVM Limit	1
Contactless CVM Limit	6

6. The functions for Offline PIN

6.1 Is Offline PIN Verified

```
/**
 * is offline verified
 * @return   -1 - Failed(Wrong PIN)
 *               0 – Not Request
 *               1 - YES
 */
int emv_offlinepin_verified()
```

6.2 Get the times of offline-pin entry

```
int emv_get_offlinepin_times()
```

6.3 Set Custom PINPAD GUI for Offline PIN

```
/**
 * set custom pinpad gui for offline pin
 * @param keyType
 *          bit 1 - bigger font
```

```
 *          bit 2 - ordered pinpad
 *          bit 4-3 - align mode : 00 - left alignment; 01 - middle alignment; 02 - right alignment
 *          bit 5 - Q3K, physical key pad with fix style. if this bit is set, bit1 will be ignored
 *          bit 6 - Q3, Landscape
 */
int emv_set_virtual_keypad_type(int keyType)
```

6.4 Set Custom PINPAD Title for Offline PIN

```
/**
 * set custom pinpad Title for Offline PIN
 * @param data              buffer of pinpad title
 *          dataLength      The length of pinpad title
 * @return
 */
int emv_set_pinpad_title(uint8_t *data, int dataLength)
```

6.5 Set Custom PINPAD Prompt for Offline PIN

```
/**
 * set Line1 & line2 prompt for offline pin
 * @param line1Data             buffer of line1 prompt
 *          line1DataLength     The length of line1 prompt
 *          line2Data           buffer of line2 prompt
 *          line2DataLength     The length of line2 prompt
 * @return
 */
int emv_set_pinpad_prompt(uint8_t *line1Data, int line1DataLength,
                          uint8_t *line2Data, int line2DataLength)
```

Annex A：Tag List defined by MasterCard

A.1 Contactless Kernel ID (DF810C)

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 = PURE contactless Reader/Kernel

9 = RUPAY
10 = Interac
11 = MIR
12 = IDEMIA WISE Contactless Reader/Kernel

A.2 CVM Capability – CVM Required (DF8118)

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*.

CVM Capability – CVM Required		
Byte 1	b8	Plaintext PIN for ICC verification
	b7	Enciphered PIN for online verification
	b6	Signature (paper)
	b5	Enciphered PIN for offline verification
	b4	No CVM required
	b3-1	Each bit RFU

A.3 CVM Capability – No CVM Required (DF8119)

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*.

CVM Capability – No CVM Required		
Byte 1	b8	Plaintext PIN for ICC verification
	b7	Enciphered PIN for online verification
	b6	Signature (paper)
	b5	Enciphered PIN for offline verification
	b4	No CVM required
	b3-1	Each bit RFU

A.4 Kernel Configuration (DF811B)

Tag: 'DF811B'
Length: 1
Format: b
Description: Indicates the Kernel configuration options.

Kernel Configuration		
Byte 1	b8	Mag-stripe mode contactless transactions not supported
	b7	EMV mode contactless transactions not supported
	b6	On device cardholder verification supported
	b5	Relay resistance protocol supported
	b4-1	Each bit RFU

A.5 Mag-stripe CVM Capability – CVM Required (DF811E)

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*.

Mag-stripe CVM Capability – CVM Required			
Byte 1	b8-5	CVM	
			0000: NO CVM
			0001: OBTAIN SIGNATURE
			0010: ONLINE PIN
			1111: N/A
			Other values: RFU
	b4-1	Each bit RFU	

A.6 Mag-stripe CVM Capability – No CVM Required (DF812C)

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*.

Mag-stripe CVM Capability – No CVM Required			
Byte 1	b8-5	CVM	
			0000: NO CVM
			0001: OBTAIN SIGNATURE
			0010: ONLINE PIN
			1111: N/A
			Other values: RFU
	b4-1	Each bit RFU	

Annex B：Tag List defined by American Expresspay

B.1 Contactless Reader Capabilities (9F6D)

Name	Description	Source	Format	Tag	Length	Values	Location/Usage
Contactless Reader Capabilities	A proprietary data element with bits 8, 7, and 4 only used to indicate a terminal's capability to support Kernel 4 mag-stripe or EMV contactless. This data element is OR'd with *Terminal Type*, Tag'9F35', resulting in a modified Tag '9F35', which is passed to the card when requested.	Terminal	n 2	'9F6D'	1	00 = Kernel 4 Contactless (Version 1.0 mag-stripe only) 40 = Kernel 4 (Contactless Version ≥ 2.0 mag-stripe only) 80 = Kernel 4 (Contactless Version ≥2.0 EMV mode and mag-stripe mode)	Configured in a reader compliant with Kernel 4 and passed to the card via a modified *Terminal Type*, Tag '9F35' when Tag '9F35' is present in the PDOL of the card

Table 4-2: Contactless Reader Capabilities – Tag '9F6D'

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0			0				Deprecated
0	0			1				Not Available for Use
0	1			0				Contactless: Mag-Stripe – CVM Not Required (C-4 Version ≥ 2.2)
0	1			1				Contactless: Mag-Stripe – CVM Required (C-4 Version ≥ 2.2)
1	0			0				Deprecated – Contactless: EMV and Mag-Stripe (C-4 Version 2.1)
1	0			1				Not Available for Use
1	1			0				Contactless: EMV and Mag-Stripe - CVM Not Required (C-4 Version ≥ 2.2)
1	1			1				Contactless: EMV and Mag-Stripe - CVM Required (C-4 Version ≥ 2.2)

Note: Bits 6 and 5 and Bits 3 to 1 are reserved and must be set to zero. In *Terminal Type – Modified*, these bits will correspond to the values defined in EMV *Terminal Type*, Tag '9F35'.

B.2 Enhanced Contactless Reader Capabilities (9F6E)

Name	Description	Source	Format	Tag	Length	Values	Location/Usage
Enhanced Contactless Reader Capabilities	Proprietary Data Element for managing Contactless transactions and includes Contactless terminal capabilities (static) and contactless Mobile transaction (dynamic data) around CVM	Terminal	b 32	'9F6E'	4		Returned to the Card in the GET PROCESSING OPTIONS in response to PDOL.

Terminal Capabilities Byte 1								
b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**Meaning**
x[1]								1 = Contact mode supported[1]
	x							1 = Contactless Mag-Stripe Mode supported
		0[2]						0 = Contactless EMV full online mode not supported (full online mode is a legacy feature and is no longer supported)
			1					1 = Contactless EMV partial online mode supported
				1				1 = Contactless Mobile Supported
					x			1 = Try Another Interface after a decline.
						0		RFU
							0	RFU
Terminal CVM Capabilities Byte 2								
b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**Meaning**
1								1 = Mobile CVM supported
	x							1 = Online PIN supported
		x						1 = Signature
			x					1 = Plaintext Offline PIN
				0				RFU
					0			RFU
						0		RFU
							0	RFU
Transaction Capabilities Byte 3								
b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**Meaning**
x								1 = Reader is offline only
	x							1 = CVM Required
		0						RFU

			0					RFU
				0				RFU
					0			RFU
						0		RFU
							0	RFU
Transaction Capabilities Byte 4								
b8	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**Meaning**
x								1 = Terminal exempt from No CVM checks
	x							1 = Delayed Authorisation Terminal
		x						1 = Transit Terminal
			0	0				RFU
					X	X	X	C-4 Kernel Version:
					0	0	1	2.2 - 2.3
					0	1	0	2.4 - 2.6
					0	1	1	2.7
					1	x	x	RFU – other values

Annex C：Tag List defined by JCB

C.1 Terminal Interchange Profile (9F53)

TIP Byte 1 (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								CVM required by reader / N/A[23]
	1							Signature supported
		1						Online PIN supported
			1					On-Device CVM supported
				0				*RFU*
					1			Reader is a Transit Reader
						1		EMV contact chip supported
							1	(Contact Chip) Offline PIN supported

TIP Byte 2

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1								Issuer Update supported[24]
	x	x	x	x	x	x	x	*Each bit* RFU

TIP Byte 3 (Rightmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x	x	x	x	*Each bit* RFU

Annex D：Tag List defined by Interac

D.1 **Merchant Type Indicator**

Tag: '9F58'
Format: 1 byte numeric
Interface: Contactless
Description: Readers shall support the definition of a Merchant Type Indicator. There shall be 5 distinct MTIs defined for contactless payment processing. The MTI is intended for use by the Card's card risk management processing

Value	Meaning
01	Default
02	Petroleum
03	Grocery
04	*Transit Open Payments
05	Undefined

D.2 Terminal Transaction Information (TTI)

Tag: '9F59'
Format: Binary
Interface: Contactless
Description: TTI shall contain data as defined in following table

TTI Byte 1 (leftmost)

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
1								Reader with display capability
	1							Interac Contact application available
		1						Interac Contact application at other Terminal
			1					CDA supported
				1				Offline Capable terminal (value '0' means online only terminal)
					1			Online PIN supported
						0		RFU
							0	RFU

TTI Byte 2

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
0	0							Contactless only Capable
0	1							Contactless & Mag stripe-read Capable
1	0							Contactless, Contact Chip & Mag stripe-read Capable
1	1							Contactless & Contact Chip Capable
		0						RFU
			0					RFU

				0				RFU
					1			Mobile NFC Device (FFI = '03') accepted
						1		Contactless Card (FFI = '00', '01' or '02') accepted
							1	Always '1'. Indicates acceptance data present in this version of TTI.

TTI Byte 3

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
0	0	0	0	0	0	0	0	All bits are RFU

D.3 Terminal Option Status (TOS)

Tag: '9F5E'

Format: Binary

Interface: Contactless

Description: TOS shall contain data as defined in following table.

TOS Byte 1 (leftmost)

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
1								Use other Interface if Different Currency
	1							Use other Interface if Different Country Code
		1						Use other Interface if Domestic transaction with different currency
			0					RFU
				0				RFU
					0			RFU
						0		RFU
							0	RFU

TOS Byte 2 (Rightmost)

B8	B7	B6	B5	B4	B3	B2	B1	Meaning
0	0	0	0	0	0	0	0	All bits are RFU

Annex E：Tag List defined by MIR

E.1 Terminal TPM Capabilities (DF55)

BYTE 1								
b8	b7	b6	b5	b4	b3	b2	b1	Value
x	-	-	-	-	-	-	-	Online PIN indicator (CVM)
1								Online PIN supported
0								Online PIN Not supported
-	x	-	-	-	-	-	-	Signature indicator (CVM)
	1							Signature supported
	0							Signature Not supported
-	-	x	-	-	-	-	-	CD-CVM indicator (CVM)
		1						CD-CVM allowed (for POS-terminals is always 1)
		0						CD-CVM Not supported (only for ATM)
-	-	-	x	-	-	-	-	RFU
-	-	-	-	x	-	-	-	EMV contact mode indicator
				1				EMV contact mode supported
				0				EMV contact mode Not supported
-	-	-	-	-	x	-	-	Offline-only terminal indicator
					1			Offline-only terminal
					0			Online-capable terminal
-	-	-	-	-	-	x	-	Delayed Authorization
						1		Terminal operates in Delayed Authorization mode
						0		Delayed Authorization mode disabled
-	-	-	-	-	-	-	x	ATM indicator
							1	Terminal is ATM
							0	Terminal is Not ATM

BYTE 2								
b8	b7	b6	b5	b4	b3	b2	b1	Value
x	-	-	-	-	-	-	-	Second tap indicator (to inform Reader when Terminal unable to go online)
1								Terminal unable to go online
0								Terminal able to go online
-	0	-	-	-	-	-	-	RFU
-	-	0	-	-	-	-	-	RFU
-	-	-	0	-	-	-	-	RFU
-	-	-	-	0	-	-	-	RFU
-	-	-	-	-	0	-	-	RFU
-	-	-	-	-	-	0	-	RFU
-	-	-	-	-	-	-	0	RFU

Annex F: Self-defined Tag List

Tag	Name	Format	Length	Description
EF01	Status check support	n	1	[Terminal & AID Parameter] 0 – No; 1 – Support
EF02	Zero check support	n	1	[Terminall & AID Parameter] 0 – No; 1 – Support
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
EF09	Interac Retry Limit	n	1	[AID Parameter]Retry times for Interac Retry Checks
EF10	Combination options(C5)	b	2	[Terminal Parameter]Combination options
EF11	Floor limit checking support	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF12	Random Transaction Selection support	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF13	Velocity Checking support	n	1	[Terminal Parameter] 0 – No; 1 – Support

EF14	Exception File support	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF15	Issuer PK Revoke(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF16	Deferred Authorization(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF17	Data Storage(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF18	Extended Logging(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF19	Tearing Recovery(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF20	Contactless Application Capability (PURE)	b	5	[Terminal Parameter]
EF21	Contactless POS Implementation Options(PURE)	b	1	[Terminal Parameter]
EF22	ATOL(PURE)	b	Var.	[AID Parameter]Additional Tag Object List
EF23	MTOL(PURE)	b	Var.	[AID Parameter] Mandatory Tag Object List
EF24	ATDTOL(PURE)	b	Var.	[AID Parameter] Authentication Transaction Data Tag Object List
EF25	Support Other Interface(C6)	n	1	[Terminal Parameter] 0 – No; 1 – Support
EF26	Allow Expired Card(MIR)	n	1	[AID Parameter] 0 – No; 1 – Yes
EF27	Threshold Application Expiration Date (YYMMDD) (MIR)	b	3	[AID Parameter]

F.1 Combination options (EF10)

Combination Options Byte 1 (Leftmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								*RFU*
	1							Status Check supported
		1						Offline Data Authentication supported
			1					Exception File Check required[19]
				1				Random Transaction Selection supported
					1			*Magstripe Mode Supported*[20]
						1		EMV Mode Supported[21]
							1	Legacy Mode Supported[22]

Combination Options Byte 2 (Rightmost)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x	x	x	x	*Each bit RFU*

F.2 Contactless Application Capability (EF20)

Byte	Bit	TTPI bit meaning and category	Value

1	8-7	RFU (00)	Contactless Application Capability
	6	EMV contactless Mode support:	
	5	EMV contact transaction support	
	4	Terminal capability to process a transaction online	
	3	Online PIN support	
	2	Signature support	
	1	RFU (0)	0
2	8	Initial terminal request related to transaction completion	Value of “Reader Contactless Floor Limit Exceeded” Contactless Pre-Processing Indicator
	7	Terminal request related to cardholder verification	Value of “CVM Required Limit Exceeded ” Contactless Pre-Processing Indicator
	6	Single unit of currency (Status Check) Check result	Value of “Status Check Requested” Contactless Pre-Processing Indicator
	5	Zero Amount Check result	Value of “ Zero Amount ” Contactless Pre-Processing Indicator
	4	RFU (0)	0
	3	Terminal requirement related to offline CAM	Contactless Application Capability
	2-1	RFU (00)	00
3	8	Support of additional tap for Issuer Response communication	Contactless Application Capability
	7	Support of Consumer Device CVM checking as a possible CVM method	Contactless Application Capability
	6	Support of torn transaction recovery using ECHO command	Contactless Application Capability
	5-1	RFU (00000)	00000
4	8-1	RFU (00000000)	00000000
5		**Terminal Interchange Profile (TIP)**	
	8	Reliability of TIP byte value in TTPI	Contactless Application Capability
	7	SDA support	Contactless Application Capability
	6	DDA support 1= DDA is supported 0= DDA is not supported	If implementation 4 is supported,Contactless Application Capability In the other cases, 0

	5	Cardholder verification method supported by the terminal	Contactless Application Capability
	4	Support of complementary terminal checks and terminal action analysis	Contactless Application Capability
	3-2	RFU (00)	
	1	CDA support	Contactless Application Capability

F.3 Contactless POS Implementation Options (EF21)

Byte	Bit	Contactless POS Implementation Options
1	8	Implementation option 1: Retrieval of data element values stored in EEPROM using GET DATA command
	7	Implementation option 2: Update of data element values stored in EEPROM data slots using PUT DATA command (no secure messaging)
	6	Implementation option 3: terminal supporting only fixed Amount transaction
	5	Implementation option 4: Application Authentication Transaction support
	4	Implementation option 5: Capacity to restrict list of supported applications
	3	Implementation option 6: Long Tap support
	2	Implementation option 7: Online Additional Tap support
	1	Implementation option 8: ECHO command support