JR

FINANCIAL INDUSTRY STANDARD

OF THE PEOPLE'S REPUBLIC OF CHINA

ICS 35.240.40

A 11

Registration number:

JR/T 0025.17-2013

China financial integrated circuit card specifications – Part 17: Enhanced debit/credit application security specification

中国金融集成电路（IC）卡规范 - 第 17 部分：借记/贷记应用安全增强规范

Issued on: February 05, 2013 **Implemented on: February 05, 2013**

Issued by: People's Bank of China

Table of Contents

Foreword

JR/T 0025 “China financial integrated circuit card specifications” is divided into the following parts:

- Part 1: Electronic purse/electronic deposit application card specification (abolished);
- Part 2: Electronic purse/electronic deposit application specification (abolished);
- Part 3: Specification on application independent ICC to terminal interface requirements;
- Part 4: Debit/credit application overview;
- Part 5: Debit/credit application card specification;
- Part 6: Debit /credit application terminal specification;
- Part 7: Debit/credit application security specifications;
- Part 8: Contactless specification independent of application;
- Part 9: Electronic purse comprehensive application guide (abolished);
- Part 10: Debit/credit card personalization guide;
- Part 11: Contactless integrated circuit card communication specification;
- Part 12: Contactless integrated circuit card payment specification
- Part 13: Low-value payment specifications based on debit/credit application;
- Part 14: Comprehensive application specification based on contactless low-value payment application;
- Part 15: Electronic cash dual-currency payment specification;
- Part 16: IC card internet terminal specification;
- Part 17: Enhanced debit/credit application security specification.

This part is part 17 of JR/T 0025.

This part was drafted in accordance with the rules given in GB/T 1.1-2009.

China financial integrated circuit card specifications - Part 17: Enhanced debit/credit application security specification

1 Scope

This part as an enhancement to JR/T 0025.7, mainly describes the debit/credit application security features requirements based on SM2, SM3, SM4 algorithm, and the security mechanism and approved encryption algorithm to achieve these security features, including IC card offline data authentication methods based on SM2 and SM3, communication security between SM4-based IC card and issuer, and security mechanisms and encryption algorithms involved in implementing these security features.

This part applies to the security-related equipment, card, terminal machine, and management, etc., of the financial debit/credit card application issued or accepted by the bank. The objective user is mainly the card, terminal and encryption device design, manufacture, management, issuance, application system research, development, integration and maintenance, and other departments (units) related to the application of the financial debit/credit IC card.

2 Normative references

The following documents are essential to the application of this document. For the dated documents, only the versions with the dates indicated are applicable to this document; for the undated documents, only the latest version (including all the amendments) are applicable to this Standard.

JR/T 0025.4 China financial integrated circuit card specifications - Part 4: Debit/credit application overview

JR/T 0025.5 China financial integrated circuit card specifications - Part 5: Debit/credit application card specification

JRIT 0025.7 China financial integrated circuit card specifications - Part 7: Debit/credit application security specification

GM/T 0002 SM4 block cipher algorithm

A card encapsulated with one or more integrated circuits for processing and storage functions.

3.7

Interface device

The part of the terminal where the IC card is inserted, including the mechanical and electrical parts in it.

3.8

Issuer action code

The action of the issuer in accordance with the content of TVR.

3.9

Magstripe

Strips containing magnetically encoded information.

3.10

Payment system environment

The set of logical conditions established in the IC card, when a payment system application compliant with JR/T 0025 is selected.

3.11

Response

The message returned to the terminal from the IC card after completing the processing of the received command message.

3.12

Script

The commands or command sequence sent from the issuer to the terminal, for the purpose of continuously inputting commands to the IC card.

3.13

SM2 algorithm

ARQC: Authorization Request Cryptogram

ATC: Application Transaction Counter

ATM: Automated Teller Machine

AUC: Application Usage Control

CDA: Combined Dynamic Data Authentication/Application Cryptogram Generation (Combined DDA/AC Generation)

CDOL: Card Risk Management Data Object List

CID: Cryptogram Information Data

Cn: Compressed Numeric

CVM: Cardholder Verification Method

CVR: Card Verification Results

DDA: Dynamic Data Authentication

DDOL: Dynamic Data Authentication Data Object List

DES: Data Encryption Standard

DOL: Data Object List

ECB: Electronic Code Book

EF: Elementary File

EMV: Europay, MasterCard and VISA

FCI: File Control Information

GPO: Get Processing Options

IAC: Issuer Action Code

IC: Integrated Circuit

M: Mandatory

MAC: Message Authentication Code

MDK: Master Key

IC card uses SM2 algorithm to generate dynamic signature, combined dynamic signature, and application cryptogram generation in accordance with the following steps:

a) The terminal issues a GENERATE AC command in accordance with the definition in JR/T 0025.5 and the CDA request bit in the command is 1.

b) If the IC card will respond with TC or ARQC, then the IC card performs the following steps:

- IC card to generate TC or ARQC;

- IC card application connects the following data elements from left to right from SM3 for hash operation:

• In the first GENERATE AC command scenario:

• The value of the data element specified in the PDOL and sent to the IC card by the terminal in the GET PROCESSING OPTIONS command in the order in which it appears.

• The value of the data element specified in CDOL1 and sent to the IC card by the terminal in the first GENERATE AC command in the order in which it appears.

• The tag, length, and value of the data element returned by the IC card in response to this GENERATE AC command, based on the order in which they are returned, and excluding the signature dynamic application data.

• In the second GENERATE AC command scenario:

• The value of the data element specified by the PDOL and sent to the IC card by the terminal in the GET PROCESSING OPTIONS command in the order in which it appears.

• The value of the data element specified in the CDOL1 and sent to the IC card by the terminal in the first GENERATE AC command in the order in which it appears.

• The value of the data element specified in the CDOL2 and sent to the IC card by the terminal in the first GENERATE AC command in the order in which it appears.

• The tag, length, and value of the data element returned by the IC card in response to this GENERATE AC command, based on the order in

Using a single 16-byte IC card application cryptogram (AC) unique key MK_{AC} and the data source as described in clause 6.1.1 as the input, to calculate the 8-byte application cryptogram using the following two steps:

a) Generate a 16-byte application cryptogram process key, SK_{AC}, using the IC card application cryptogram (AC) unique key MK_{AC} and a two-byte IC card application transaction counter as input, in accordance with the algorithm described in 8.1.3.

b) Calculate the application cryptogram (TC, ARQC or AAC) using the 16-byte application cryptogram process key SK_{AC} which is generated from the previous step and “the selected data” as input, in accordance with the MAC algorithm specified in clause 8.1.2.

Detailed steps of generating cryptogram are as follows:

Step 1: The terminal sends the terminal data specified in the CDOL to the card by generating the application cryptogram command. If the CDOL has a transaction certificate (TC) hash result, the terminal will put this data in the command data domain.

Step 2: In accordance with the result of the card risk management, the card decides to return the cryptogram type as TC, AAC or ARQC, to generate the data block of cryptogram:

- Transaction certificate (TC) hash result (if present);

- Generate data to be sent to the card using the cryptogram command, excluding the TC hash result;

- Card internal data.

Step 3: Fill and segment the data as described in clause 8.1.2 “Padding and segment”.

Step 4: Generate the application cryptogram by using the symmetric key algorithm using the process key as shown in Figure 1 (the process key is generated by the IC card application cryptogram (AC) unique key MK_{AC} through dispersion, and the detailed generation method is as shown in clause 8.1.3).

Step 5: Take the left 8 bytes of the result of the previous step to get an 8-byte cryptogram.

Figure 2 -- Algorithm to generate ARPC

7 Security message

7.1 Message integrity and verification

7.1.1 MAC process key generation

The first step in the security message MAC generation consists of dispersing a unique 16-byte security message authentication code (MAC) unique key and the 2-byte ATC from the IC card to obtain a unique 16-byte security message authentication code (MAC) process key. AND the process key generation method is as shown in clause 8.1.3.

7.1.2 MAC calculation

The MAC is calculated by using the MAC process key generated in accordance with the method described in 7.1.1 and applying the mechanism described in 8.1.2 to the message to be protected.

In this part, the length of the MAC is 4. After getting the 16-byte result through calculation using the method aforementioned, the leftmost 4 bytes are used as the MAC.

7.2 Message privacy

7.2.1 Encryption process key generation

The first step in security message encryption/decryption consists of dispersing of the unique 16-byte encryption security message encryption unique key and the 2-byte ATC from the IC card to obtain a unique 16-byte encryption process key. And the process key generation method is as shown in clause 8.1.3.

7.2.2 Encryption and decryption

The encryption/decryption of the plaintext/encrypted command data domain is carried out by using the encryption process key generated in accordance with the method described in clause 7.2.1 and applying the mechanism described in clause 8.1.1.

Encrypt the blocks X_1, X_2, ..., X_K into 16-byte blocks Y_1, Y_2, ..., Y_K using the encryption process key K_s in accordance with the grouping encryption algorithm in the CBC mode, so that when i = 1, 2, ..., K, respectively calculate: Y_i: = ALG (K_s) [$X_i \oplus Y_{i-1}$]. The initial value of Y_0 is:

Y0: = ("00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00")

Which is recorded as: Y: = (Y_1 II Y_2 II ... II Y_K) = ENC (K_s) [MSG].

Decryption process is as follows:

a) ECB mode

When i = 1, 2, ..., K, respectively calculate: X_i: = ALG_{-1} (K_s) [Y_i].

b) CBC mode

When i = 1, 2, ..., K, respectively calculate: X_i: = ALG_{-1} (K_s) [Y_1] $\oplus Y_{i-1}$.

The initial value of Y_0 is:

Y0: = ("00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00" II "00")

To get the original message MSG, connect the blocks X_1, X_2 ..., X_K, and if padding is used, delete the ("80" II "00" II "00" II ... II "00" II) from the last block X_k, written as: MSG = DEC (K_s) [Y].

8.1.2 Message authentication code

The 16-byte grouping encryption algorithm in CBC mode and the MAC process key K_s are used to calculate an S-byte MAC value (4 ≤ S ≤ 8) S for a message MSG of any length as follows.

a) Padding and block

In accordance with GB/T 16649.4 to pad the message MSG, mandatorily add one “80” byte to the right end of the MSG, and then add the minimum “00” byte to the right end, to make the so that the result message length MSG: = (MSG "80" II "00" II "00" II ... II "00" II) is an integer multiple of 16 bytes.

The MSG is then split into 16-byte blocks X_1, X_2, ... X_K.

The SM2 signature scheme uses the following three functions:

- A signature function Sign (S_K) [M] depending on the private key S_K, which outputs two numbers r and s of the same length;

- A verification function Verify (P_K) (M, Sign (S_K) [M]] depending on the public key P_K, which outputs True or False, indicating that the verification is correct or failure;

- A hashing algorithm SM3 [ ] that maps the message of any length to a 32-byte hash value.

8.2.2 Digital signature generation

The process of calculating the signature S by MSG for a message of any length is as follows:

a) Calculate Z_A = SM3 [$ENTL_A$ II ID_A II a II b II x_G II y_G II x_A II y_A], where ID_A is fixedly set to 16-byte fixed-length hexadecimal data 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38; and the value of $ENTL_A$ is at 2-byte data 0x00, 0x80;

b) Calculate the 32-byte HASH value of message MSG h: = SM3 [Z_A II MSG];

c) Calculate Sign (S_K) [h] to get two numbers r and s;

d) The digital signature S is defined as S: = r II s, i.e., the digital signature S is made up of the numbers r and s in series.

8.2.3 Digital signature verification

The process of signing a message S composed of any length data by the MSG is as follows:

a) Calculate Z_A = SM3 [$ENTL_A$ II ID_A II a II b II x_G II y_G II x_A II y_A], where ID_A is fixedly set to 16-byte fixed-length hexadecimal data 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38; and $ENTL_A$ value is 2-byte data 0x00, 0x80;

b) Calculate the 32-byte HASH value of message MSG h: = SM3 [Z_A II MSG];

c) Verify (P_K) [h, S], if the function outputs True, it indicates that the verification is correct, if the output is False, the verification fails.

b) Dual algorithm card:

- In accordance with the value of the SM algorithm indicator (DF69) of the GPO command, it identifies the SM algorithm flow of this transaction and determines the algorithm environment of subsequent cards and related data elements:

- The directly returned GET PROCESSING OPTIONS response data is the AFL of the SM algorithm. AFL specifies the specific data related to the SM algorithm, including the public key parameters of the SM2 algorithm and the location of the certificate.

If the terminal does not support the SM algorithm, that is, the SM algorithm indication tag (DF69) cannot be identified by the terminal, the terminal will provide a data element of a specified length in the GET PROCESSING OPTIONS command and set its numerical part to 0 in hexadecimal. For a single SM algorithm card and dual algorithm card, the processing flow is as follows:

a) Single SM algorithm card:

- The directly returned GET PROCESSING OPTIONS response data is the AFL of SM algorithm. AFL specifies the specific data related to the SM algorithm, including the public key parameters of the SM2 algorithm and the location of the certificate. This will cause the offline data authentication process to fail because of a failure to find the corresponding public key index, and the transaction can try online or refuse the transaction.

b) Dual algorithm card:

- Default the RSA/SHA-1/3DES algorithm process in accordance with the value of SM algorithm indicator (DF69) of the GPO command, to determine the algorithm environment and related data elements of the follow-up card;

- The returned GET PROCESSING OPTIONS response data is the AFL of RSA/SHA-1/3DES algorithm. AFL specifies the specific data related to the RSA/SHA-1/3DES algorithm, including the RSA algorithm's public key parameters and the location of the certificate.

10.3.4 Follow-up process

See the definition in JR/T 0025.4.

10.4 Low-value payment process based on debit/credit application

K_{ENC}: = SM4 (KMC) [The rightmost 6 bytes of KEYDATA II “F0” II “01” II The rightmost 6 bytes of KEYDATA II “0F” II “01”].

K_{MAC}: = SM4 (KMC) [The rightmost 6 bytes of KEYDATA II “F0” II “02” II The rightmost 6 bytes of KEYDATA II “0F” II “02”].

K_{DEK}: = SM4 (KMC) [The rightmost 6 bytes of KEYDATA II “F0” II “03” II The rightmost 6 bytes of KEYDATA II “0F” II “03”].

11 PIN change/unlock command data calculation method

11.1 Change PIN value using current PIN

If the P2 parameter in the command is equal to “01”, the command data domain includes the PIN encrypted data and the MAC. The generation of the PIN encrypted data is performed in accordance with the following steps:

a) The issuer determines the security message encryption master key used to encrypt the data, and disperse it to generate the security message encryption unique key of the card: ENC UDK;

b) Generate process key K_S;

c) Generate 8-byte PIN data block D3;

1) Generate the first 8-byte data block D1:

Byte 1		Byte 2		Byte 3		Byte 4		Byte 5	Byte 6	Byte 7	Byte 8
0	0	0	0	0	0	0	0	Byte 5 ~ 8 of ENC UDK			

2) Generate the second 8-byte data block 02:

Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
0	N	P	P	P	P	P/F	P/F	P/F	P/F	P/F	P/F	P/F	P/F	F	F

N: The number of new PIN (hexadecimal)

P: New PIN, length 4-12 digits (2-6 bytes)

3) D1 and D2 perform exclusive OR to get D3

d) Generate 8-byte data block D4 using the current PIN:

Byte 1		Byte 2		Byte 3		Byte 4		Byte 5		Byte 6		Byte 7		Byte 8	
P	P	P	P	P/0	P/0	P/0	P/0	P/0	P/0	P/0	P/0	0	0	0	0

This is an excerpt of the PDF (Some pages are marked off intentionally)

------ The End ------