PBOCIC Chip Card Reading Process

https://blog.csdn.net/kxd_ysheng/article/details/21178101?_t=t

PBOCIC chip card reading process, refer to the above blog, and sort out the PBOCIC card reading process.

The content of the above blog is probably also based on other people, so several mistakes are the same as other places on the Internet, and I have corrected them and added more detailed explanations

The process of reading the PBOCIC card begins

(1) Application selection

One UnionPayThere may be multiple applications in the IC card, the so-called application is the application protocol and related data se between the card and the terminal (such as the card reader), the interaction between the card reader and the card is actually the interaction with the application, and the card transaction is actually to choose a certain application to do the transaction.

At present, the cards produced by card merchants are basically an application, even so, but according to UnionPay's specifications, the application selection step is also essential.

(1) The first is to select the PSE payment environment document 1PAY.SYS. DDF01

The command used is the select command, which is described in detail in Appendix B of the China Financial Integrated Circuit (IC) Card Debi Credit Card Specification (V0.8-20080325).

The following message is requested:

Command format: Select packet 00 A4 XX

Command parameters: 00 A4 04 00

0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31

00

Parameter length: 20

Card returns data:

Length: 40

Value: 6F 1E

84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31

A5 0C

5F 2D 02 7A 68

9F 11 01 01

88 01 01

90 00

The results returned here are strictly exactlyTLV format, the parsed data is as follows:

6F: FCI (File Control Information) template

84: DF name

A5: FCI-specific template

88: SFI of the basic file in the directory (the upper three digits are 0, followed by 100 to read the file data).

5F2D: Preferred language

9F11: Issuer code table index

BF0C: Issuer custom data -- (the card does not carry this data).

FCI (File Control Information) contains SFI (Short File Identification), and each app in the directory lists one app.

At this point, the PSE selection is complete.

If the card's response to the selected payment environment is not 90 00, you will need to try to select AID one by one.

(2) Read the records according to the SFI and select the application file associated with the PSE

The basic file associated with the PSE, which lists the supported payment apps.

hereSFI (corresponding to tag 88) is 01, and according to the read record specification, the upper 5 bits of p2 are SFI (01==>000**0 0001).**), the lower three digits are 100, so the value of p2 is 0x0C (**0000 1**100).

TABLE B.21 READ RECORD COMMAND REFERENCE CONTROL PARAMETER (P2)

b8	**B7**	**b6**	**b5**	**b4**	b3	b2	b1	significance
x	x	x	x	x				SFI
					1	0	0	Read the record specified in P1

The request packet is as follows:

Command format: Read Record packet 00 B2 P1 P2 00

Command parameters: 00 B2 01 0C 00

Parameter length: 5

Card returns data:

Length: 49

Value:

70 28

61 26

4F 08 A0 00 00 03 33 01 01 01

50 0A 50 42 4F 43 20 44 45 42 49 54

9F 12 0A 50 42 4F 43 20 44 45 42 49 54

87 01 01

90 00

The results returned here are also in accordance withTLV format, parsed as:

70: Basic data template

61: Apply a template. It exists only if PSE exists, including parameters related to the application directory entry

4F: Application Identifier (AID) = RID+PID (Application Provider and Private Application Identifier), Len=08, Value=A000000333010101

50: Application label, EMV is required data for application selection, data related to AID that is easy to remember, Len=0A, Value=50424F43204445424954

9F12: App preferred name, Len=0A, Value=50424F43204445424954

87: Application Priority Indicator 01

Once **you have an AID**, you should be selected for the **AID** list.

(3) After adding to the list, continue to read the next record of the file until there are no records to read

The request packet is as follows:

Command format: Read Record (Read Record) message 00 B2 P1 P2 00

Command parameters: 00 B2 02 0C 00

Parameter length: 5

The card returns data:

Length: 2

Value: 6A 83

(4) Select Apply

Join based on the app identifier of the read recordAID list, and then select the app based on the user's keystrokes or select the app based on priority.

The following message is used to request the command:

Command format: Select packet 00 A4 P1 P2 Lc data 00

Command parameters: 00 A4 04 00 07 A0 00 00 03 33 01 01 00

Parameter length: 13

Card Returns:

Length: 45

Value:

6F 45

84 08 A0 00 00 03 33 01 01 01 A5 39 50 0A 50 42 4F 43 20 44 45 42 49 54 87 01 01

9F 38 09 9F 7A 01 9F 02 06 5F 2A 02

5F 2D 02 7A 68

9F 11 01 01

9F 12 0A 50 42 4F 43 20 44 45 42 49 54

BF 0C 05 9F 4D 02 0B 0A

90 00

6F: FCI template

84: DF name

9F38: PDOL data (9F1A: terminal country code; 9F7A: Electronic Cash Indicator, 9F02: Authorized Amount, 5F2A: Transaction Currency Code).

5F2D: Preferred language

9F11: Issuer code table index

Make the final selection, ask the user to enter the amount, judge whether the POS can take the electronic cash transaction path, indicating the electronic cash indicator.

(2) Application initialization

(5) GET PROCESSING OPTIONS (GPO--GET PROCESSING OPTIONS) NOTIFIES THE CARD TRANSACTION TO BEGIN.

The combination of command packets requires the PDOL data (9F38) returned in step 4 and packs them according to the PDOL options

Request Command Packet:

TABLE B.12 GET PROCESSING OPTIONS COMMAND PACKETS

encode	value
CLA	'80'
INS	'A8'
P1 P2	'00'
Lc	'00'
Data domains	PDOL related data, if present, or 8300
Le	'00'

Command format: GPO packet 80 A8 00 00 XX (length) 83 (PDOL label) XX (PDOL length) PDOL 00

Command parameters: 80 A8 00 00 0B

83 09

9F 02 06 5F 2A 02 00

Parameter length: 14

Card Returns:

80 16

7C 00

08 01 01 00

10 01 01 01

10 03 06 00

18 01 01 00

20 01 01 00

90 00

GPO response format: 80 XX (length) XX (application interaction characteristics) XX (AFL).

So, the application interaction feature is 7C 00 (0111 1100) and the AFL is 08 01 01 00 10 01 01 01 10 03 06 00 18 01 01 00 20 01 01 00

The packets returned here are notTLV format.

7C00 is the application interaction feature, analysis:

bit8: Reserved (RFU);

bit7:1=SDA support;Data-at-rest authentication

bit6:1=Support DDA;Dynamic data authentication

bit5:1=Cardholder authentication is supported

bit4: Perform endpoint risk management

bit3: Issuer authentication is supported

bit2: Reserved

bit1: 1=CDA support

Byte 2: Reserved (RFU)

The 08010100100101011003060018010100200101 00 are **AFLs** (5 in total), AFLs (Application File Locators), and each AFLIncludes 4 bytes:

Byte **1**: **bit8-bit4**: **SFI (**short file identifier**).**

bit3-bit1:000

Byte **2**: The record number of **the first** record to be read in the file (cannot be 0).

Byte **3**: The record number of **the last record to be read** in the file (greater than or equal to byte 2).

Byte **4**: **the number of static data records starting from** the record number of byte **2** (starting from 0, not greater than (byte 3)- (byte 2) + 1).

Based on **the AFL** returned by the **GPO**, read the file.

The format of the read file number (P2) is: SFI (AFL first byte) is shifted by 3 bits to the left, and 100 is added to the right (*calculated as the READ RECORD command). P2*).

For example, the above 08 binary is **0000 1**000, because bit8-bit4 is SFI, so the real is: 0000 0001.

When reading a file, move 3 bits to the left and fill 0100 to the right (indicating that the specified record is read) to get 0000 1100, which is 0x0c. In the same way, P2 is as follows:

The first file: 08 01 01 00 byte 1 binary is **0000 1**000 after the left shift right 0000 1100 i.e. 0x0C

The second file: 10 01 01 01 byte 1 binary is **0001 0**000 after the left shift right 0001 0100 is 0x14

The third file: 10 03 06 00 bytes 1 binary is **0001 0**000 after the left shift right 0001 0100 is 0x14

The fourth file: 18 01 01 00 bytes 1 binary is **0001 1**000 after the left shift and the right complement 0001 1100 is 0x1C

The fifth file: 10 01 01 00 bytes 1 binary is **0001 0**000 after the left shift right 0001 0100 i.e. 0x14

Table **jl-1.0** analyzes the AFLs returned above

The nth file	**The first byte**	**The second byte (**the start record number**).**	**The third byte (**end record number**).**	**The third byte**	**remark**
1	0C	01	01	00	1 record from start to finish
2	14	01	01	01	1 record
3	14	03	06	00	4 records
4	**1C**	01	01	00	1 record
5	**14**	01	01	00	1 record

(3) Reading application data

Based on the **AFL** analysis result **in step 5**, you can assemble packets that read application data.

TABLE B.20 READ RECORD COMMAND MESSAGE

encode	value
CLA	‘00’
INS	‘B2’
P1	**Record number**
P2	**Refer to the control parameters** in Table B-21
Lc	does not exist
Data domains	does not exist
Le	‘00’

Table B.21 defines the reference control parameters for command messages.

TABLE B.21 THE READ RECORD COMMAND REFERENCES THE CONTROL PARAMETERS

b8	**B7**	**b6**	**b5**	**b4**	b3	b2	b1	significance
x	x	x	x	x				SFI
					1	0	0	Read the record specified in P1

Send

Command format: READ RECORD packet 00 B2 P1 P2 00

Command length: 5

Command parameters: 00 B2 **01 0C** 00 [according to the first file in the table **"jl-1.0″**].

accept

Length:

Value:

70 74

57 13 6214830100237436D22112201464200755165F

9F 0B 2D 4C 49 20 58 49 4E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 2 0 20 20

9F 61 28 31 34 32 34 32 39 31 39 38 32 30 35 31 34 34 34 31 36 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20

9F 62 01 00

90 00

70: Template

5F20: Name of cardholder

57: Two-track equivalence data (master account, separator, expiration date, service, PIN verification domain, custom data non-even supplement F).

9F0B: Cardholder's name

9F61: Cardholder's ID number

9F62: ID Type, 00 - ID Card

Send:

00 B2 **01 14** 00 [According to the second **file of the table "JL-1.0"**]

Accept:

70 4A

5F 24 03 22 11 30

5F 25 03 12 11 21

5A 08 62 14 83 01 00 23 74 36

5F 34 01 00

9F 07 02 FF 00

8E 0C 00 00 00 00 00 00 00 00 42 03 1E 03

9F 0D 05 D8 60 9C A8 00

9F 0E 05 00 10 00 00 00

9F0F05D8689CF800

5F28020156

9000

9F0E : Issuer Behavior Code (IAC) - Reject

9F0D : Issuer Behavior Code (IAC) - Default

5F28: the country code of the card issuer

5F34: the serial number of the primary account of the application

5F25: The effective date of the application

5A: Two tracks

Send:

00 B2 **03 14** 00 [According to table **"JL-1.0"**, the third file, the first record】

Accept:

7081B3

9081B0A33373F56850C06B1DE59D93F4F2A362CEA2A57C37D5DC78E3BF84914B608A43ACD22539B6C0491E88652244D2A199E6516D0C6F
3C632F65202561106420FA406F530AE8D39CAF6EA5730E4A5AB15763FB42EAFA0AEE69A9E69640642E29CA15D5449B7AB74411A50EF811A
DEF4D8F93004C423E77E55AC263DA89308FCC05E3E0AA599565590C83CCE7FF6B17602F1BD7DB22C48A5A039FB97C48A8FA16083CC56DD
6FEC3C6E7E81F37DCA25A949EC9000

90: Issuer public key certificate - used for offline data authentication

Send:

00 B2 0**4 14** 00 [According to table "**JL-1.0**", the third file, the second record】

Accept:

7048

8F0103

920433DFA015

9F320103

9F470103

9F482A737DBBA0A11741BD57FEDF1BE60C551A88A24A82B5EB3182642F217AFA986E04DB7AC2F38057FCE8AC01

9F49039F3704

9F4A0182

9000

8F: CA public key index (PKI).

9F32: issuer public key index

9F4A: static data certification label list

92: the remainder of the issuer's public key

9F47: IC card public key index

9F48: The remainder of the IC card's public key

9F49: Dynamic Data Authentication Data Object List (DDOL).

Send:

00 B2 **05 14** 00 [According to table "**JL-1.0**", the third file, the third record】

Accept:

708193

93819088E48EC5AC68FD509E73786839978F465F4BDB905EF38B9FBD17147FE810EAB68EB5E86C4A23BA2E897A54892D1DAD75EE339C216
3CB172C66CE312AA9B585D5F24A3827342D504AF9EC2E95407AA72CE527DC1C0BA4D051B8BA6282AEDEEE1D3725EC8CA91EC515903D7E
100576B837F96ABCB789C10434AFEC23E30D33B509FBBFB40ACDB15DF282372B8F3FBC99D9000

93: Signed Static Application Data - SDA

Send:

00 B2 **06 14** 00 [According to table "**JL-1.0**", the third file, the fourth record】

Accept:

708194

9F468190810A5485E3B940508DDFF1E6AC2BF5B8990742BC1E11146C8BFC46D8C3C425ABA2C264CE2F478C320D198FF03E090EA61ECA3CC
CF493AA886EDAC940C2F0CDC248768204DF4DEC26758B18F67E9CC7AB84C7DA55F00BDF0127CAF12B09E93C1830120E5F700BAB5D9124F
209037FC3BE5BC44D34153760CB5E79890B C5D00491681ECDD1C9844874D9B0DFFF81236C899000

9F46: IC card public key certificate

Send:

00 B2 **01 1C** 00 [Fourth File, First Record]

Accept:

7048

8C1B9F02069F03069F1A0295055F2A029A039C019F37049F21039F4E14

8D1A8A029F02069F03069F1A0295055F2A029A039C019F37049F2103

5F30020220

9F42020156

9F08020020

9000

5F30 : 0220 Service Code

8C: Card Risk Management Data Object List 1 (CDOL1).

8D: Card Risk Management Data Object List 2 (CDOL2).

9F08: the version number of the application

9F42: Apply the currency code

Send:

00 B2 **01 24** 00 [Fifth Document, First Record]

Accept:

7008

9F140100

9F230100

9000

9F14: Continuous offline trading floor

9F23: Continuous offline trading cap

(4) Generate application ciphertext

This command is generated based on the Card Risk Management Data Object List 1 (CDOL1) read above.

Request Command Packet:

80AE40002E000000000009000000000000156000080000001561004100 0B84FBA072019024C4E000000000000000000000000000000000000000000
00

Card Returns:

801E800012E70CB84C238601F207010103A020000100000000007B168F919000

Data parsing returned: ciphertext information data (L:1) + Apply Transaction Counter (L:2) + Apply Redaction (L:8) + Issuer App Data

The whole process of reading the PBOCIC card is over.
