SmartOS-KW

(electronic wallet)

Reference Manual
v.1.1.0

Disclaimer of Liability

The content of this manual has been checked for agreement with the hardware described. Since deviations cannot be precluded entirely, full agreement is not guaranteed. However, the data in this manual are reviewed regularly and any necessary corrections will be included in subsequent versions. Suggestions for improvement are welcomed.

General Information

The smart card SmartOS KW is a microprocessor smart card with SmartOS-KW operating system, that implements an electronic wallet with a set of commands for managing the balance inside the card.
The smart card SmartOS KW is suitable for loyalty applications, prepaid cards, electronic wallets, etc. that needs low cost and readiness.

The set of commands supplied by SmartOS-KW implemens an electronic wallet: credit is protected by a user PIN and by an administration PIN. User PIN allow reading the available credit while administration PIN allow full access to the credit (read, write, update).

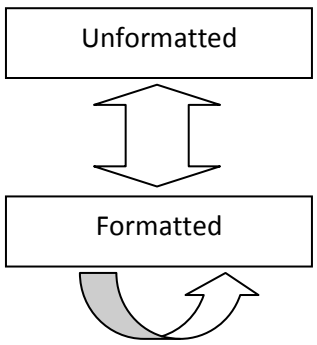
SmartOS KW supplies also a 256 byte of public memory for Onwer Data.

The User PIN is an alphanumeric value of max 8 numbers/characters. After 3 wrong trials the PIN is blocked and can be unblocked by the PUK.

The Administration PIN is an alphanumeric value of max 8 numbers/characters. After 3 wrong trials the PUK is blocked and cannot be unblocked.

Life Cycle

The life cycle of the SmartOS KW has two states: Unformatted and Formatted as shown in the following picture:



After production the smart card is in Unformatted state.
The command Format is used to move from Unformatted to Formatted state.
In Formatted state the smart card can be formatted again, infinite times, using the Format command.

Tecnical Specification

- Microchip with 2KB EEPROM (1,5 KB used by the operating system)
- SmartOS KW Operating System
- T = 1 Protocol
- Compliant with ISO 7816 part 1,2,3
- Compliant with any PC/SC, CCID reader
- Command Set (APDU) specific for electronic wallet, really easy to use
- user PIN max 8 digits, max 3 trials
- administrative PIN max 8 digits, max 3 trials

Default values:

User PIN: 12345678
Administrative PIN: 12345678
Format Key: 1234567890

Commands

The following table shows the set of commands respect to the state of the card:

Command	Unformatted	Formatted
CHANGE REFERENCE DATA		X
DECREASE BALANCE		X
FORMAT	X	X
GET BALANCE		
GET DATA	X	X
INCREASE BALANCE		X
READ OWNER DATA		X
SET BALANCE		X
WRITE OWNER DATA		X
VERIFY PIN		X

CHANGE REFERENCE DATA

CLA	INS	P1	P2	LC	DATA	LE
00	24	00	*id*	08	*<new value>*	00

Changed thte PIN or the PUK as specified in P2.
P2 = 1 User PIN
P2 = 2 Administative PUK.
DATA fields contains the new value for PIN/PUK.

Conditions:
- PIN has already been verified by the command Verify PIN

Example
PIN changed in 1234:
00 24 00 01 08 31 32 33 34 FF FF FF FF 00

DECREASE BALANCE

CLA	INS	P1	P2	LC	DATA	LE
80	D2	00	*00*	04	*<value> (as 32 bit integer)*	00

Decrease the balance with the specified value

Conditions:
- the Administrative PIN has been verified by Verify PIN

Example:
Decrease the balance of 100
80 D2 00 00 04 00 00 00 64 00

FORMAT

CLA	INS	P1	P2	LC	DATA	LE
C0	41	00	00	0A	*<format key>*	00

Formats the EEPROM and deletes the content moving the card is *Formatted* state.
DATA field must contain the right *format key*

Example
C0 41 00 00 0A 31 32 33 34 35 36 37 38 39 30 00

GET DATA

CLA	INS	P1	P2	LC	DATA	LE
00	CA	00	*mode*	00	*<empty>*	00

Reads system information specified in P2 as described in the following table:

mode	Description
80	Manufacturer
81	Microchip Identification Code
82	ID operating system (1- SmartOS K1, 2 – SmartOS K2, 3- SmartOS KW, 4- SmartOS CK)
83	Life Cycle: *unformatted* = 10, *formatted* = 20
85	Error counter format key
86	Error counter PIN
87	Error counter PUK

Example
Gets the PIN error counter:
00 CA 00 86 00 00

GET BALANCE

CLA	INS	P1	P2	LE	DATA	LE
80	B1	00	*00*	00	*<empty>*	04

Reads the balance in the card.
The balance is expressed as 32 bit integer

Conditions:
- il User PIN has been verified by Verify PIN

INCREASE BALANCE

CLA	INS	P1	P2	LC	DATA	LE
80	D1	00	*00*	04	*<value> (as 32 bit integer)*	00

Increments the balance with the specified value.

Conditions:
- il Administrative PIN ha been verified by Verify PIN

Example:
Increase the balance of 100
80 D1 00 00 04 00 00 00 64 00

READ OWNER DATA

CLA	INS	P1	P2	LC	DATA	LE
80	B0	*00*	*Offset*	*len*	*<empty>*	*Len*

Reads data in the public memory (Owner Data) starting from the offset specified in P2.
LE is the number of bytes to read (up to 127 at a time)

Conditions:
- reading Onwer Data can be always performed

Example
Reads 10 bytes from public memory starting from 32:
80 B0 00 10 00 0A

SET BALANCE

CLA	INS	P1	P2	LC	DATA	LE
80	D3	*00*	*00*	*04*	*<value> (as 32 bit integer)*	*00*

Set the balance with the specified value

Conditions:
- The administrative PIN has been verified by Verify PIN

Example:
Set the balance with 100
80 D3 00 00 04 00 00 00 64 00

WRITE OWNER DATA

CLA	INS	P1	P2	LC	DATA	LE
80	D6	*00*	*offset*	*len*	*<data>*	*00*

Writes data in the public memory (Owner Data) starting from the offset specified in P2.
LC must contains the length of the DATA field (up to 127 at a time)

Example
Writes 10 bytes to public memory starting from 32:
80 D6 00 10 0A 31 32 33 34 35 36 37 38 39 30 00

VERIFY PIN

CLA	INS	P1	P2	LC	DATA	LE
00	20	00	*id*	*08*	*<pin>*	*00*

Verifies PIN or PUK as specified in P2
P2 = 1 User PIN
P2 = 2 Administrative PIN.

If verification succeeds the *PIN* is set to "verified" and the related error counter is cleared.

Example
Verifies the PINwith the value 1234:
00 20 00 01 08 31 32 33 34 FF FF FF FF 00

Error codes

SW1	SW2	Description
Normal Processing		
0x90	0x00	Successful Command
0x61	0xXX	Successful Command, SW2 contains the number of *APDU Response* bytes still available
Warning		
0x62	0x00	Generic Warning
0x62	0x81	Invalid or corrupted data
0x62	0x82	EOF reached
0x62	0x83	The selected file is invalid
0x62	0x84	Invalid File Control Information (FCI)
0x63	0x00	Generic Error
0x63	0x81	File is full
0x63	0xCX	Error meaning depends on the specific command
Processing Errors		
0x64	0xXX	Internal Processing Error
0x65	0x00	Generic Error
0x65	0x81	Storing Error
0x66	0xXX	Reserved for future extensions
Verify Errors		
0x67	0x00	Invalid Command Length (LC)
0x68	0x00	(CLA) Command not supported
0x68	0x81	Channel not supported
0x68	0x83	*Secure Messaging* Mode not supported
0x69	0x00	Command not permitted
0x69	0x81	The Command is incompatible with the file structure
0x69	0x82	Access denied (access permissions not granted)
0x69	0x83	*Security Object* is blocked
0x69	0x84	Invalid Command Data
0x69	0x85	Using conditions unsatisfied
0x69	0x86	Invalid Command, no file selected
0x69	0x87	*Secure Messaging* Object not found
0x68	0x88	Invalid *Secure Messaging* Object
0x6A	0x00	Wrong P1 or P2 field
0x6A	0x80	Wrong Parameters in DATA field
0x6A	0x81	Not Supported Function
0x6A	0x82	File Not Found
0x6A	0x83	Record Not Found
0x6A	0x84	Not enough free space on file or memory
0x6A	0x85	Inconsistent LC field respect to TLV structure
0x6A	0x86	Wrong P1 or P2 field
0x6A	0x87	Inconsistent LC field respect to P1 and P2 fields
0x6A	0x88	Object not found
0x6B	0x00	Wrong P1 and P2 fields
0x6C	0xXX	Wrong LE field. XX indicates the correct size of *APDU Response*'s DATA field
0x6D	0x00	Invalid or not supported INS field
0x6E	0x00	Invalid or not supported CLA field
0x6F	0x00	Internal Error