SmartOS-K2

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.

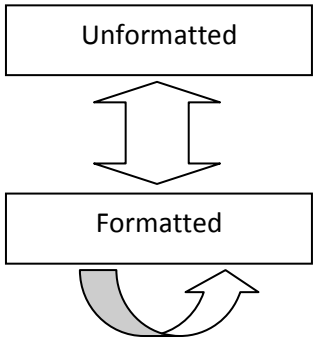
General Information

The smart card SmartOS K2 is a microprocessor, multiapplication, cryptographic smart card with 8K of EEPROM (about 4 taken by the operating system), compliant with ISO7816 and with any PC/SC or CCID smart card reader, that gives a set of commands compliant with ISO7816-4. SmartOS K2 is suitable for application requiring a secure storage, strong authentication, low cost and usability readiness such as PKI, digital signature, Single Sign-On, Biometric Recognition, Authentication, Encryption, etc.

The Format Key is an alphanumeric value of 10 numbers/characters. After 10 wrong trials the Format Key is blocked and cannot be unblocked.

Life Cycle

The life cycle of the SmartOS K2 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 8KB EEPROM (about 4 taken by OS)
- Protocol T = 1
- Compliant with ISO 7816 1,2,3
- Compliant with any PC/SC, CCID reader
- Cryptographic Algorithm: DES (3DES, AES, RSA, ECC, SHA1 only on K2e “enhanced”)
- Command set compliant with ISO 7816 4

Default values:

Format Key: 1234567890

Commands

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

Command	Unformatted	Formatted
APPEND RECORD		X
CHANGE REFERENCE DATA		X
CREATE FILE		X
DELETE FILE		X
EXTERNAL AUTHENTICATE		X
FORMAT	X	X
GET CHALLENGE		X
GET DATA	X	X
INTERNAL AUTHENTICATE		X
PUT DATA		X
READ BINARY		X
READ RECORD		X
RESET RETRY COUNTER		X
SELECT		X
UPDATE BINARY		X
UPDATE RECORD		X
VERIFY PIN		X

Table A1 - *Access Condition* for an *Elementary File*

Byte	Valore	Descrizione
0	00, FF, <*sdo_id*>*	Access Condition *Read*;:
1	00, FF, <*sdo_id*>*	Access Condition *Write*:
2	00, FF, <*sdo_id*>*	Access Condition *Delete*:
3	00, FF, <*sdo_id*>*	Access Condition *Activate*:

Table A2 - *Access Condition* for a *Dedicated File*

Byte	Valore	Descrizione
0	00, FF, <*sdo_id*>*	Access Condition *Create:* to creat other files in the DF
1	00, FF, <*sdo_id*>*	Access Condition *Write*: to write SDO in the DF
2	00, FF, <*sdo_id*>*	Access Condition *Delete:* to delete the DF
3	00, FF, <*sdo_id*>*	Access Condition *Activate*: to activare/deactive the DF

Table A3 - *Access Condition* for a *Security Data Object*

Byte	Valore	Descrizione
0	00, FF, <*sdo_id*>*	Access Condition *Write*: to write in the SDO
1	00, FF, <*sdo_id*>*	Access Condition *Change*: to modify the SDO
2	00, FF, <*sdo_id*>*	Access Condition *Unblock*: to unlock the SDO

* 00 stands for access always denied, FF stands for access always granted.

APPEND RECORD

CLA	INS	P1	P2	LC	DATA	LE
00	E2	00	00	len	*<data>*	00

Append a record in the currently selected EF.
LC contains the length of the data field and must be equal to the length of the record.

Conditions:
- The current EF is LINEAR_FIXED o CYCLIC_FIXED
- The *Access Condition Write* on the current EF is verified

CHANGE REFERENCE DATA

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

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

Conditions:
- PIN or PUK 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

CREATE FILE

CLA	INS	P1	P2	LC	DATA	LE
00	E0	00	00	0A	*<File Descriptor>*	00

Creates an EF or a DF in the currently selected DF.
DATA field must be the following format:

Byte	Valore	Descrizione
0	00	Data isn't compliant to ISO7816
1-2	<*File ID*>	File ID (ex. 3100)
3	00 – DF 01 – Binary 02 – Linear Fixed 03 – Cyclic Fixed	File Type
4	00, FF, <*sdo_id*>*	If EF, Access Condition *Read* If DF, Access Condition *Create*
5	00, FF, <*sdo_id*>*	Access Condition *Write*;
6	00, FF, <*sdo_id*>*	Access Condition *Delete;*
7	00, FF, <*sdo_id*>*	Access Condition *Activate*
8-9	Hi-Low	If binary EF File Length as hi -low If EF linear fixed or cyclic fixed Hi is record length Low is number of records If DF must be 00 00

00 stands for access always denied, FF stands for access always granted.

Example:
to create a binary EF with ID 3100, length 128 byte, read and write linked to PIN 01
00 E0 00 00 0A 00 31 00 01 01 01 FF FF 00 80

Conditions:
- The *Access Condition Create* on the current DF must be verified

DELETE FILE

CLA	INS	P1	P2	LC	DATA	LE
00	E4	00	00	02	*<File ID>*	00

Deletes the EF or the DF having the ID specified in DATA
A DF is deletable if it is empty

Conditions:
- the *Access Condition Delete* must be granteed

EXTERNAL AUTHENTICATE

CLA	INS	P1	P2	LC	DATA	LE
00	82	00	*id*	*08*	*<Data>*	00

Executes an external autenticate using DATA and using the *Security Data Object* (the Key) specified in P2.
DATA fiels must be in *big-endian*
DATA

If authentication succeeds the specified *Security Data Object* is verified

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 CHALLENGE

CLA	INS	P1	P2	LC	DATA	LE
00	84	00	*00*	00	*<vuoto>*	08

Reads a 8 bytes random value used later for EXTERNAL AUTHENTICATE command

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

INTERNAL AUTHENTICATE

CLA	INS	P1	P2	LC	DATA	LE
00	88	00	*id*	*08*	*<Data>*	00

Executes a cryptographic operation on the data field using the *Security Data Object* specified in P2.
DATA must be in *big-endian,* LC is the length of DATA field

PUT DATA

CLA	INS	P1	P2	LC	DATA	LE
00	DA	00	*mode*	*len*	*<Data>*	00

If mode is equal to 10 or 20 writes the object as defined by DATA on the current DF (as described on the table below). If mode = 30 modify the Access Condition of the currently selected DF or EF.

Mode	Description
10	DATA describes a PIN-type Security Data Object
20	DATA describes a KEY-type Security Data Object
30	DATA contains the new Access Condition being associated to the currently selected EF.
40	DATA contains the new Access Condition being associated to the currently selected DF.

DATA field format for mode = 10 or 20

Byte	Value	Description
0	<SDO ID>	SDO ID (e.g. 20)
1	00	SDO type. Values other than 00 are reserved for future use
2	00 up to FF	Max number of wrong attempts **
3	00	RFU
4	00, FF, < SDO ID>*	Access Condition Change
5	00, FF, < SDO ID>*	Access Condition Write
6	00, FF, < SDO ID>*	Access Condition Unblock

7-8	Hi-Low	Object length in hi-low format
9-n	<data>	SDO contents

DATA field format for mode = 30 or 40

Byte	Value	Description
0	00	RFU
1	00	RFU
2	00	RFU
3	00	RFU
4	00, FF, < *SDO ID*>*	Se EF Access Condition *Read* *Se DF* Access Condition *Create*
5	00, FF, < *SDO ID*>*	Access Condition *Write*
6	00, FF, < *SDO ID*>*	Access Condition *Deletee*
7	00, FF, < *SDO ID*>*	Access Condition *Activate*
8	00	RFU
9	00	RFU

* a value of 00 indicates in no way the permission is granted; inversely, a value of FF indicates the permission is always granted.
** each SDO allows a maximum number of wrong verification attempts after which the SDO is blocked and can be unblocked via a RESET RETRY COUNTER APDU.

For mode = 30, the format of the DATA field is shown on Tables A1 and A2 depending on the currently selected file type
LC must specify the DATA field length.

READ BINARY

CLA	INS	P1	P2	LC	DATA	LE
00	B0	*mem*	*offset*	00	*< empty >*	*len*

Reads the content of the memory specified in P1 starting from the offset specified in P2.
P1 = 1 public memory.
P1 = 2 private memory.
LE number of bytes to read (up to 127 at a time)

Conditions:
- reading from protected memory can be done only after PIN verification
Example
Reads 10 bytes from public memory starting from 32:
00 B0 01 10 00 0A

READ RECORD

CLA	INS	P1	P2	LC	DATA	LE
00	B2	*rec*	*mode*	00	*<vuoto>*	*len*

Reads the content of a record in the currently selected EF.
The meaminig of P1 and P2 is described in the following table:

P1	P2	Descriprion
00	00	FIRST: first recotd in the EF
00	01	LAST: last record in the EF
00	02	NEXT: next record in the EF

00	03	PREV: prevoius record
xx	04	CURRENT/ABSOLUTE: if P1 = 0 reads the current record, else P1 is the index of the record.

LE contains the number of bytes to read.

Conditions:
- The *Access Condition Read* has been granted

RESET RETRY COUNTER

CLA	INS	P1	P2	LC	DATA	LE
00	2C	*01*	*id*	len	*<new value>*	*00*

Clears the error counter of the SDO as specified by P2 and modify the value of the object with the contents of the DATA field

Conditions:
– the Unblock Access Condition on the specified SDO has been verified

SELECT FILE

CLA	INS	P1	P2	LC	DATA	LE
00	A4	*mode*	*00*	len	*<file id>*	*len*

Selects the file as specified by the DATA field according to the mode specified by P2.

The meaning of P1 is as follows:

P1	Description
00 01 02	Select DF or EF, as specified by DATA field, on the current DF directly.
03	Select the parent DF of the current DF.
08	Select the DF or the EF using the absolute path specified by the DATA field.

LC must specify the length of the DATA field.
If LE > 0 the Response ADPU returns name, type, Access Conditions and size of the selected file

Conditions:
– the Read Access Condition on the current EF has been verified

UPDATE BINARY

CLA	INS	P1	P2	LC	DATA	LE
00	D6	*hi*	*low*	*len*	*<data>*	*00*

Writes the content in DATA fields in the memory specified by P1 starting from the offset specified in P2.
P1 = 1 public memory
P1 = 2 private memory.
LC must contains the length of the DATA field (up to 127 at a time).

Conditions:

- Writing in protected memory can be done only after PIN verification.

Example
Writes 10 bytes to public memory starting from 32:
00 D6 01 10 0A 31 32 33 34 35 36 37 38 39 30 00

UPDATE RECORD

CLA	INS	P1	P2	LC	DATA	LE
00	DC	*rec*	*mode*	*len*	*<data>*	*00*

Writes the contents of DATA on the currently selected EF on the record as specified by P1 and P2 according to the following table:

P2	Description
00	FIRST: overwrite the first record of the file
01	LAST: overwrite the last record of the file
02	NEXT: overwrite the next record
03	PREV: overwrite the previous record
04	CURRENT/ABSOLUTE: if P2 = 0 overwrite the current record else P2 contains the index of the record being overwritten.

LC must specify the length of the DATA field.

Conditions:
– the Write Access Condition on the current EF has been verified

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 PIN
P2 = 2 PUK.

If verification succeeds the *PIN*/*PUK* 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