ZEITCONTROL CARDSYSTEMS GMBH

ZCPurse V3.1

Product Specification

2017-05-30

For more information and contact info see:

www.cipurse.de

Detailed specification of ZCPurse, the ZeitControl OSPT CIPURSE implementation.

Date	Version	Author	Change Description
2013-10-11	0.1	MP	Initial version
2013-12-10	0.2	MP	Changes for new software version
2014-02-03	2.0	MP	Adaptation to current product version
2014-07-16	2.2	MP	Adaptation to current product version
2014-08-05		MP	Added Performance Optimization Hints
2015-09-02	3.1	MP	Adaptation to current product version ZCPurse V3.1
2015-12-03		MP	Review
2016-08-10		MP	Review
2017-03-13		MP	Added specification of “Maximum size of security attribute in proprietary format for each ADF”
2017-05-13		MP	Added section “Optional Preinstalled Products and Features”
2017-05-26		MP	Technical data added
2017-05-30		MP	Review

Content

1 Introduction

ZCPurse is the ZeitControl implementation of the CIPURSE Open Standard specification. This document covers the details of the ZCPurse implementation including proprietary extensions and clarifications as well as technical data.

For more information about CIPURSE Open Standard and to get access to standard documents visit:

http://www.osptalliance.org

ZCPurse is based on the following CIPURSE specifications:

- CIPURSE(TM) V2, Operation and Interface Specification, Revision 2.0 [1]
- CIPURSE(TM) V2, Cryptographic Protocol, Revision 1.0 [2]
- CIPURSE(TM) V2, CIPURSE(TM) L Profile Specification, Revision 2.0 [3]
- CIPURSE(TM) V2, CIPURSE(TM) S Profile Specification, Revision 2.0 [4]
- CIPURSE(TM) V2, CIPURSE(TM) T Profile Specification, Revision 2.0 [5]

The listed products are further compliant to several international standards, including:

- ISO/IEC 7816-3:2006 "Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange" [6]
- ISO/IEC 14443-1:2008 "Identification cards – Contactless integrated circuit cards – Proximity cards – Physical characteristics" [7]
- ISO/IEC 14443-2:2010 "Identification cards – Contactless integrated circuit cards – Proximity cards – Radio frequency power and signal interface" [8]
- ISO/IEC 14443-3:2011 "Identification cards – Contactless integrated circuit cards – Proximity cards – Initialization and anticollision"[1] [9]
- ISO/IEC 14443-4:2008 "Identification cards – Contactless integrated circuit cards – Proximity cards – Transmission protocol" [10]

[1] Type A according to ISO14443-3 is used.

2 Products

ZCPurse V3.1 is the current CIPURSE product of ZeitControl cardsystems GmbH.

The following variants are available:

Product	Form Factor	Profiles[2]	Memory
ZCPurse L	Plastic Card	L	512 Bytes
ZCPurse S 2k	Plastic Card	L, S	2kB
ZCPurse S 4k	Plastic Card	L, S	4kB
ZCPurse T 8k	Plastic Card	L, S, T	8kB
ZCPurse T 12k	Plastic Card	L, S, T	12kB
ZCPurse SAM [3]	Plastic Card, SIM	L, S, T (SAM)	12kB

Additionally the following optional preinstalled products exist:

Feature ID	Base Product	Description
Preload K0	Any ZCPurse product	Preloaded Secure Root Key K0
CPM SAM	ZCPurse SAM	CPM SAM

See section 9, “Preinstalled Products and Features“ for further details.

3 Basic Definitions and Concepts

3.1 Glossary

Term	Description
OSPT	Open Standard for Public Transportation Alliance e.V.
CIPURSE	CIPURSE specification
ChipID	ChipID is a unique identifier of each ZCPurse respective CIPURSE card. It includes the IC manufacturer code and manufacturer defined card identification data. This content can be read from EF.ID_Info at offset 8 with length 16.

Table 1 Glossary

CIPURSE and OSPT are registered trademarks of the OSPT Alliance in Germany and other territories.

3.2 General Structure and Common Notation

For numeric notation we use the following:

- Single quotes, e.g. ‘A5’ or ‘A1 A5 75 08’: number or set of numbers in hexadecimal notation
- Suffix b, e.g. 0100 1101b: number in binary notation
- Suffix h, e.g. A5h: number in hexadecimal notation

[2] Supported CIPURSE profiles

[3] The ZCPurse SAM implements the CIPURSE SAM specification.

4 Supported Features

4.1 ZCPurse L

All required features according to L-Profile specification [3] are supported. Details may exceed requirements for L-Profile as described in Table 2.

4.1.1 Memory and File System Configuration

User Memory	512 Bytes
Number of keys in MF	2
Maximum number of ADFs	1
Maximum number of PxSEs	1
Maximum number of keys per ADF	2
Maximum number of EFs per ADF	2
Maximum size of security attribute in proprietary format for each ADF	64 bytes[4]
Supported EF types	linear record file linear value record file

Table 2 Memory Configuration Profile L

Use of PxSE is supported according to CIPURSE specification [1].

EF.ID_INFO is supported according to CIPURSE specification [1]. EF.ID_INFO can be selected independent of which DF is selected. Initial access conditions to EF.ID_INFO are as described in [3]. Attributes may be changed. A different set of access conditions is maintained for each ADF. Nevertheless write access to EF.ID_INFO content is not possible.

4.1.2 Command Set

Following commands are supported:

SELECT	MUTUAL_AUTHENTICATE	GET_CHALLENGE
READ_BINARY (only for EF.ID_Info)	READ_RECORD	UPDATE_RECORD
READ_VALUE	INCREASE_VALUE	DECREASE_VALUE
READ_FILE_ATTRIBUTES	UPDATE_FILE_ATTRIBUTES	UPDATE_KEY
UPDATE_KEY_ATTRIBUTES	CREATE_FILE	FORMAT (Proprietary)
Other Proprietary Commands[5]		

[4] According to [3] a maximum of 16 bytes needs to be supported by L profile cards. The value of 64 bytes as supported by ZCPurse L exceeds this limit. L profile cards of other manufactures may not accept security attributes in proprietary format with length greater than 16 bytes.

[5] See section 7 Proprietary Features

4.2 ZCPurse S 2k/4k

All required features according to S-Profile specification [4] are supported. Details may exceed requirements for S-Profile, as described in Table 3.

Note: ZCPurse S also supports creation of ADFs of Type CIPURSE™ L.

4.2.1 Memory and File System Configuration

User Memory	2 or 4kB
Number of keys in MF	8
Maximum number of ADFs	8
Maximum number of PxSEs	8
Maximum number of keys per ADF	8
Maximum number of EFs per ADF	8
Maximum size of security attribute in proprietary format for each ADF	64 bytes
Supported EF types	linear record file linear value record file cyclic record file binary file

Table 3 Memory Configuration Profile S

Use of PxSE is supported according to CIPURSE specification [1].

EF.ID_INFO is supported according to CIPURSE specification [1]. EF.ID_INFO can be selected independent of which DF is selected. Initial access conditions to EF.ID_INFO are as described in [3]. Attributes may be changed. A different set of access conditions is maintained for each ADF. Nevertheless write access to EF.ID_INFO content is not possible.

4.2.2 Command Set

Following commands are supported:

SELECT	MUTUAL_AUTHENTICATE	GET_CHALLENGE
READ_RECORD	APPEND_RECORD	UPDATE_RECORD
READ_VALUE	INCREASE_VALUE	DECREASE_VALUE
READ_BINARY	UPDATE_BINARY	READ_FILE_ATTRIBUTES
UPDATE_FILE_ATTRIBUTES	UPDATE_KEY_ATTRIBUTES	UPDATE_KEY
CREATE_FILE	DELETE_FILE	ACTIVATE_FILE
DEACTIVATE_FILE	FORMAT (Proprietary)	Other Proprietary Commands[6]

[6] See section 7 Proprietary Features

4.3 ZCPurse T 8k/12k

All required features according to T-Profile specification [5] are supported. Details may exceed requirements for T-Profile, as described in Table 4.

Note: ZCPurse T also supports creation of ADFs of Types CIPURSE™ L and CIPURSE™ S.

4.3.1 Memory and File System Configuration

User Memory	8 or 12kB
Number of keys in MF	8
Maximum number of ADFs	8
Maximum number of PxSEs	8
Maximum number of keys per ADF	8
Maximum number of EFs per ADF	32
Maximum size of security attribute in proprietary format for each ADF	64 bytes
Supported EF types	linear record file linear value record file cyclic record file binary file

Table 4 Memory Configuration Profile T

Use of PxSE is supported according to CIPURSE specification [1].

EF.ID_INFO is supported according to CIPURSE specification [1]. EF.ID_INFO can be selected independent of which DF is selected. Initial access conditions to EF.ID_INFO are as described in [3]. Attributes may be changed. A different set of access conditions is maintained for each ADF. Nevertheless write access to EF.ID_INFO content is not possible.

4.3.2 Command Set

Following commands are supported:

SELECT	MUTUAL_AUTHENTICATE	GET_CHALLENGE
READ_RECORD	APPEND_RECORD	UPDATE_RECORD
READ_VALUE	INCREASE_VALUE	DECREASE_VALUE
READ_BINARY	UPDATE_BINARY	READ_FILE_ATTRIBUTES
UPDATE_FILE_ATTRIBUTES	UPDATE_KEY_ATTRIBUTES	UPDATE_KEY
CREATE_FILE	DELETE_FILE	ACTIVATE_FILE
DEACTIVATE_FILE	PERFORM_TRANSACTION	CANCEL_TRANSACTION
FORMAT (Proprietary)	Other Proprietary Commands[7]	

[7] See section 7 Proprietary Features

4.4 ZCPurse SAM

All required features according to T-Profile specification [5] are supported. Details may exceed requirements for T-Profile, as described in Table 5.

Note: ZCPurse SAM also supports creation of ADFs of Types CIPURSE™ L and CIPURSE™ S.

ZCPurse SAM additionally implements SAM features according to [11].

4.4.1 Memory and File System Configuration

User Memory	8 or 12kB
Number of keys in MF	8
Maximum number of ADFs	8
Maximum number of PxSEs	8
Maximum number of keys per ADF	8
Maximum number of EFs per ADF	32
Maximum size of security attribute in proprietary format for each ADF	64 bytes
Supported EF types	linear record file linear value record file cyclic record file binary file

Table 5 Memory Configuration SAM

Use of PxSE is supported according to CIPURSE specification [1].

EF.ID_INFO is supported according to CIPURSE specification [1]. EF.ID_INFO can be selected independent of which DF is selected. Initial access conditions to EF.ID_INFO are as described in [3]. Attributes may be changed. A different set of access conditions is maintained for each ADF. Nevertheless write access to EF.ID_INFO content is not possible.

4.4.2 Command Set

Following commands are supported:

SELECT	MUTUAL_AUTHENTICATE	GET_CHALLENGE
READ_RECORD	APPEND_RECORD	UPDATE_RECORD
READ_VALUE	INCREASE_VALUE	DECREASE_VALUE
READ_BINARY	UPDATE_BINARY	READ_FILE_ATTRIBUTES
UPDATE_FILE_ATTRIBUTES	UPDATE_KEY_ATTRIBUTES	UPDATE_KEY
CREATE_FILE	DELETE_FILE	ACTIVATE_FILE
DEACTIVATE_FILE	PERFORM_TRANSACTION	CANCEL_TRANSACTION
AUTHENTICATE_SAM	AUTHENTICATE_CBP	GENERATE_SM_ELEMENTS
VERIFY_SM_ELEMENTS	LOAD_KEY	PERFORM_SYM_CRYPTO
READ_SESSION_KEY	GENERATE_KEY	VERIFY_SAM_PASSWORD
END_SESSION	GET_KEY_INFO	DIVERSIFY KEYSET
FORMAT (Proprietary)	Other Proprietary Commands[8]	

[8] See section 7 Proprietary Features

5 Initial Configuration

Initially, if not ordered otherwise, the card is empty (no ADF preloaded).

Initial values are as described in [1] and the corresponding profile specification ([3], [4] or [5]).

Specifically, the MF contains two keys for profile L and eight keys for other profiles which are all set to the following value:

```
'73 73 73 73 73 73 73 73 73 73 73 73 73 73 73 73'
```

6 Memory Usage

A DF requires the following memory:

Element	Memory per object of element
Basic DF Structure	19 bytes
FCP Security Attributes	Length of attribute excluding tag 86h and length byte
Key / NbrK>0	21 x NbrK bytes
Child Files / NbrEf	-

Table 6 Memory Usage per DF

A PxSE requires the following memory:

Element	Memory per object of element
Basic DF Structure	16 bytes
FCP Security Attributes	Length of attribute excluding tag 86h and length byte

Table 7 Memory Usage per PxSE

An EF requires the following memory:

Element	Memory per object of element
Basic EF Structure	13 bytes
NbrK>0	NbrK + 4 bytes
EF Data / Size	1 byte for each content byte according to specified size

Table 8 Memory Usage per EF

Examples:

A unsecured ADF with no proprietary security attributes and NbrEf specified = 2:

```
Basic DF Structure: 19 bytes
FCP Security Attributes: 0 bytes
NbrK: 0 bytes
= 19 + 0 + 0 = 19 bytes
```

A secured ADF with 2 keys, 4 bytes proprietary security attributes and NbrEf specified = 4:

```
Basic DF Structure: 19 bytes
FCP Security Attributes: 4 = 4 bytes (excluding Tag and Length)
NbrK: 21 x 2 = 42 bytes
= 19 + 4 + 42 = 65 bytes
```

A unsecured EF of size 64 bytes:

```
Basic EF Structure: 13 bytes
NbrK: 0 bytes
Size: 64 bytes
= 13 + 0 + 64 = 77 bytes
```

A secured EF with 2 keys and a data size of 12 bytes:

```
Basic EF Structure: 13 bytes
NbrK: 2 + 4 = 6 bytes
Size: 12 bytes
= 13 + 6 + 12 = 31 bytes
```

7 Proprietary Features and Limitations

7.1 Detection of ZeitControl ZCPurse Version

Each ZeitControl ZCPurse card returns the following MANUF data (see [1]) in EF.ID_INFO:

'5A 43 50 75 72 73 65 20 56 VV VV XX XX XX XX XX'[9]

VV VV is replaced by version number, i.e. '33 31' =ASCII "31" for version 3.1

XX XX XX XX XX are proprietary information not covered by this document.

7.2 FORMAT Command

The FORMAT command resets the memory content to initial delivery state of the card. This includes all settings for the MF.

The (unencrypted) command APDU is:

CLA	'80'	
INS	'FC'	
P1	'00'	
P2	'00'	
LC	'07'	
Data	'43 6F 6E 66 69 72 4D'	This is ASCII "ConfirM"
LE	Absent	

Table 9 FORMAT Command

Conditions of use:

- The MF must be selected, otherwise SW1SW2=6D00 (INS not supported) is returned.
- The Data "ConfirM" must be provided (see above), otherwise SW1SW2=6D00 (INS not supported) is returned.
- P1 and P2 must be both 0
- Le must not be present
  The Access Conditions for "DELETE_FILE(EF or ADF)" (ACG_MF_5, SMG_MF_1) must be fulfilled.
  Note: The access condition ACG_MF_5 is not enabled by default initial configuration for ZCPurse L. Take care to specify suitable conditions for ACG_MF_5 and SMG_MF_1 otherwise the format command is denied.

[9] 5A 43 50 75 72 73 65 20 56 = ASCII "ZCPurse V"

7.3 UPDATE FILE ATTRIBUTES Command

UPDATE FILE ATTRIBUTES command is supported according to [1]. Regarding size of **security attribute in proprietary format** the following applies. It **must not exceed size** of these attributes **used during creation of the ADF**.

If required the following procedure may be used to create an ADF with enough space for security attribute in proprietary format reserved. If applied the reserved space is available for security attribute in proprietary format at later time when changed using UPDATE FILE ATTRIBUTES command.

Use the following steps:

1. In CREATE ADF set security attribute in proprietary format to a byte string of maximum desired length. This string may for example consist of a number of 0 bytes. It must not exceed the maximum length supported by ZCPurse according to section 4.
2. Call UPDATE FILE ATTRIBUTES to set security attribute in proprietary format as desired for the ADF. For example:
    a. Call UPDATE FILE ATTRIBUTES with security attribute in proprietary format of length 0 to remove the security attribute in proprietary format from ADF. Still the space used during original creation of the ADF is reserved and available for later calls to UPDATE FILE ATTRIBUTES.
    b. Or call UPDATE FILE ATTRIBUTES with security attribute in proprietary format of length and content as desired to be assigned to the ADF. The length must be at most the length used for original creation of the ADF. The space reserved during original creation of the ADF is still reserved and not reduced even if a smaller size is used in here.

8 Performance Optimization Hints

To get optimal performance the following hints may help:

- If no PxSE use is required, select the ADF immediately using SELECT ADF with AID
- If PxSE use is required mark your ADF to be automatically selected on PxSE selection. This has to be done during ADF creation. Note that this automatic selection may not be possible if another ADF is already marked the same way: in this case your ADF must be selected manually with SELECT ADF.
- Order commands such that a minimum number of EF/ADF selections are required.
- For EFs use short file id (SFID). In a sequence of commands that access the same EF, only specify the SFID in first command. Use current EF (SFID=0) for subsequent commands in the sequence.
- Create EFs in order of use. The most used EF, or the EF that requires fastest access, should be created first.

9 Preinstalled Products and Features

Note: The following sub sections and products are preliminary and subject to change.

9.1 Preloaded Secure Root Key K0

Optionally on request, a **Secure Root Key K_0** can be installed on each ZCPurse card. Typically K_0 is installed as MF Key number 2. K_0 is a card individual key. K_0 is computed from secret **Master Key KM_0** using unique **ChipID** as diversification data. ChipID is taken from EF.ID_Info at offset 8 with length 16. This includes IC manufacturer code (IC_MAN, 1 byte) and manufacturer defined card identification data (CHIP_IDENT, 15 byte).

For using secure root key a CPM SAM is required.

Note: In general a single Master Key KM_0 is used for all cards and customers. On request of a customer, a customer specific Master Key KM_0 may be created, maintained and be used for that customer only.

9.2 CPM SAM

The Card Personalization Manager SAM (CPM SAM) allows secure transfer of keys and application DFs using Preloaded Secure Root Key K0. For that purpose it maintains 16 slots for application specific master keys (KM_{CPM0} to KM_{CPM16}). These keys can be used in diversified version (K_{CPM0} to K_{CPM16}) within each application specific end user card.

For ease of use the SAM initially includes two preloaded ADFs. Depending of use typically one of both is deleted before setting up the other one. The main difference in between both SAM ADF is:

- Master SAM ADF does not allow KM_{CPMx} to be used for operational use. This shall prevent these keys from being used for key export encryption. Only KM_0 respective K_0 shall be used for that purpose.
- Master SAM ADF allows export of KM_{CPMx} itself while Perso SAM ADF allows export of diversified versions (K_{CPMx}) of that key only. This means Perso SAM ADF can only be used to export card individual (or otherwise diversified) keys, while Master SAM ADF allows export of master keys.
- Perso SAM ADF allows use of diversified keys K_{CPMx} for operational use. By this the Perso SAM may authenticate to a target card using these keys.

For each ADF access rights are restricted. Only following operations are allowed:

Right	Items	Restrictions
READ FILE ATTRIBUTES	all	always
READ BINARY	EF.SAMInfo	always
UPDATE BINARY	EF.SAMPwd	Key 2
	EF.SAMInfo	Key 2
UPDATE KEY, UPDATE KEY ATTRIBUTES	ADF	Key 2 only
DEACTIVATE FILE	ADF	Key 2
ACTIVATE FILE	ADF	Key 2
GENERATE KEY, LOAD KEY	All SAM keys	Key 1 (K_0), Key 2

Further K_0 is loaded to MF key number 2 of CPM SAM. Beside this content and access rights of MF are unchanged default values. SAM operator may want to change them as needed.

9.2.1 CPM Master SAM ADF

The target use for the CPM Master SAM ADF is securely maintaining the application specific master keys.

Target operations for the SAM ADF are:

- Generation of application specific master keys in SAM (GENERATE KEY)
- Transferring keys securely to one or more backup CPM Master SAMs
- Transfer application specific master keys securely to multiple CPM Perso SAMs.

The CPM Master SAM ADF includes the following items, identifier and keys:

AID:	‘D2 76 00 00 02 80 02 00 00 00 00 00 00 03 10’
Initial SAM Password[10]:	‘00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F’
SAM use:	LOAD SAM
ADF Key 1:	$\mathbf{K_0}$
ADF Key 2 (initial value)[11]:	‘73 73 73 73 73 73 73 73 73 73 73 73 73 73 73 73’
SAM Operational Keys (FN=60h):	Secure root master key $\mathbf{KM_0}$; Key ID: 0001h; Value preloaded;
SAM Personalization Keys (FN=70h):	16 key slots for application specific master keys (KM_{CPM0} to KM_{CPM16}); Key IDs: 0101h to 0110h; Can either be generated or loaded by SAM operator (CPM); Key use options: Can be exported; Cannot be exported in plaintext; Can be exported without diversification; Cannot be used as encryption key

[10] Value can be changed using UPDATE BINARY if authenticated with key 2.
[11] Value can be changed using UPDATE KEY. Restricted update self.

9.2.2 CPM Perso SAM ADF

The target use for the CPM Perso SAM ADF is personalization of end user cards.

The SAM may also be used as operational SAM, in this case the application specific personalization master keys shall not be loaded (only the operational keys shall be loaded). Further the SAM Use may be changed to STANDARD or BACK SAM as suitable, by modification of EF.SAMInfo.

The target operations are:

- Creating application DFs with assigned card individual keys securely into ZCPurse cards preloaded with secure root key K_0.
- Transferring application specific keys (KM_{CPM0} to KM_{CPM16}) securely to card as individual keys (or otherwise diversified) keys (K_{CPM0} to K_{CPM16})
- Perform authentication to ADF structures using diversified keys (K_{CPM0} to K_{CPM16}) based on application specific master keys (KM_{CPM0} to KM_{CPM16})

The CPM Master SAM ADF includes the following items, identifier and keys:

AID:	‘D2 76 00 00 02 80 02 00 00 00 00 00 00 03 11’
Initial SAM Password[12]:	‘00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F’
SAM use:	PERSO SAM
ADF Key 1:	**K_0**
ADF Key 2 (initial value)[13]:	‘73 73 73 73 73 73 73 73 73 73 73 73 73 73 73 73’
SAM Operational Keys (FN: 60h):	Secure root master key **KM_0**; Key ID: 0001h; Value preloaded; 16 key slots for application specific master keys (KM_{CPM0} to KM_{CPM16}); Key IDs: 0101h to 0110h; To be loaded from CPM Master SAM;
SAM Personalization Keys (FN:70h):	16 key slots for application specific master keys (KM_{CPM0} to KM_{CPM16}); Key IDs: 0101h to 0110h; To be loaded from CPM Master SAM; Key use options: Can be exported; Cannot be exported in plaintext; Must be diversified before export; Cannot be used as encryption key

[12] Value can be changed using UPDATE BINARY if authenticated with key 2.
[13] Value can be changed using UPDATE KEY. Restricted update self.

9.2.3 Technical Security Notes

Key 2 of each SAM ADF is initially loaded with a default value. The access conditions assigned to that key allows changing the key value. Since access rights for key value and key attributes are technically the same, the attributes of that key can be changed as well. The possible modification of attributes of that key may be used to enable that key for key encryption use. This results in a security risk for master keys (KM_{CPM0} to KM_{CPM16}), which then may be exported encrypted with this *insecure* key. This is typically not desired.

Also the password is setup with an initial value. This should for security reasons be changed.

So following operations are recommended to initialize a SAM (respective SAM ADF) for secure corporate use:

1. Replace default key number 2 in SAM ADF using UPDATE KEY with a secure final value. Note: Once the procedure described here is complete, the key cannot be changed anymore.
2. Set key attributes of key 2 to 00000100b=04h (key is valid, no use as key encryption key, key access rights are denied, neither key value nor attributes can be changed anymore) using UPDATE KEY ATTRIBUTES
3. Change the password of SAM using UPDATE BINARY on EF.SAMPwd (File ID = 1001h). Only change first 16 bytes in EF.SAMPwd, which are the passwords.

After this is done, the SAM can be fully trusted for corporate desired use. I.e. to generate secure master keys in KM_{CPM0} to KM_{CPM16} or to transfer existing master keys from another SAM to this SAM.

10 Security Conformance

The used chip hardware is certified according to Common Criteria level EAL6+ with certification id "ANSSI-CC-2014/84". Note: The software (ZCPurse OS) is not included in this certification.

11 Technical Data

11.1 Storage Resistance

Data retention:	25 Years
Erase write cycles:	500.000

11.2 Temperature Ratings

Chip operating temperature:	-25°C to +85°C
Chip storage temperature:	-65°C to +150°C
Standard PVC cards temperature resistance:	-30°C to 50°C[14]

[14] Enhanced materials and limits available on request

12 References

[1] OSPT Alliance, CIPURSE(TM) V2, Operation and Interface Specification, Revision 2.0, 2013-12-20.

[2] OSPT Alliance, CIPURSE(TM) V2, Cryptographic Protocol, Revision 1.0, 2012-09-28.

[3] OSPT Alliance, CIPURSE(TM) V2, CIPURSE(TM) L Profile Specification, Revision 2.0, 2013-12-20.

[4] OSPT Alliance, CIPURSE(TM) V2, CIPURSE(TM) S Profile Specification, Revision 2.0, 2013-12-20.

[5] OSPT Alliance, CIPURSE(TM) V2, CIPURSE(TM) T Profile Specification, Revision 2.0, 2013-12-20.

[6] ISO/IEC, *ISO/IEC 7816-4: Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange,* 2005.

[7] ISO/IEC, *ISO/IEC 14443: Identification cards — Contactless integrated circuit(s) cards — Proximity cards Part1: Physical characteristics,* Second edition 2008-06-15.

[8] ISO/IEC, *ISO/IEC 14443: Identification cards — Contactless integrated circuit(s) cards — Proximity cards Part 2: Radio frequency power and signal interface,* 2010.

[9] ISO/IEC, *ISO/IEC 14443: Identification cards — Contactless integrated circuit(s) cards — Proximity cards Part 3: Initialization and anticollision,* 2011.

[10] ISO/IEC, *ISO/IEC 14443: Identification cards — Contactless integrated circuit(s) cards — Proximity cards Part 4: Transmission protocol,* 2008.

[11] OSPT Alliance, CIPURSE(TM) V2, SAM Specification, Revision 1.0, 2013-10-14.