

CEN

CWA 15748-6

WORKSHOP

July 2008

AGREEMENT

ICS 35.240.50

Supersedes CWA 15748-6:2008, February

English version

**Extensions for Financial Services (XFS) interface specification -
Release 3.10 - Part 6: PIN Keypad Device Class Interface -
Programmer's Reference**

This CEN Workshop Agreement has been drafted and approved by a Workshop of representatives of interested parties, the constitution of which is indicated in the foreword of this Workshop Agreement.

The formal process followed by the Workshop in the development of this Workshop Agreement has been endorsed by the National Members of CEN but neither the National Members of CEN nor the CEN Management Centre can be held accountable for the technical content of this CEN Workshop Agreement or possible conflicts with standards or legislation.

This CEN Workshop Agreement can in no way be held as being an official standard developed by CEN and its Members.

This CEN Workshop Agreement is publicly available as a reference document from the CEN Members National Standard Bodies.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

© 2008 CEN All rights of exploitation in any form and by any means reserved worldwide for CEN national Members.

Ref. No.: CWA 15748-6:2008 D/E/F

Table of Contents

Foreword	5
1. Introduction.....	8
1.1 Background to Release 3.10	8
1.2 XFS Service-Specific Programming	8
2. Pin Keypad	9
3. References	11
4. Info Commands	12
4.1 WFS_INF_PIN_STATUS.....	12
4.2 WFS_INF_PIN_CAPABILITIES	15
4.3 WFS_INF_PIN_KEY_DETAIL.....	23
4.4 WFS_INF_PIN_FUNCKEY_DETAIL.....	25
4.5 WFS_INF_PIN_HSM_TDATA.....	28
4.6 WFS_INF_PIN_KEY_DETAIL_EX.....	29
4.7 WFS_INF_PIN_SECUREKEY_DETAIL.....	31
4.8 WFS_INF_PIN_QUERY_LOGICAL_HSM_DETAIL	35
5. Execute Commands	36
5.1 Normal PIN Commands	36
5.1.1 WFS_CMD_PIN_CRYPT	36
5.1.2 WFS_CMD_PIN_IMPORT_KEY	39
5.1.3 WFS_CMD_PIN_DERIVE_KEY	42
5.1.4 WFS_CMD_PIN_GET_PIN.....	44
5.1.5 WFS_CMD_PIN_LOCAL_DES	47
5.1.6 WFS_CMD_PIN_CREATE_OFFSET	49
5.1.7 WFS_CMD_PIN_LOCAL_EUROCHEQUE.....	51
5.1.8 WFS_CMD_PIN_LOCAL_VISA.....	53
5.1.9 WFS_CMD_PIN_PRESENT_IDC	55
5.1.10 WFS_CMD_PIN_GET_PINBLOCK	57
5.1.11 WFS_CMD_PIN_GET_DATA	59
5.1.12 WFS_CMD_PIN_INITIALIZATION	62
5.1.13 WFS_CMD_PIN_LOCAL_BANKSYS	64
5.1.14 WFS_CMD_PIN_BANKSYS_IO	65
5.1.15 WFS_CMD_PIN_RESET	66
5.1.16 WFS_CMD_PIN_HSM_SET_TDATA.....	67
5.1.17 WFS_CMD_PIN_SECURE_MSG_SEND.....	69
5.1.18 WFS_CMD_PIN_SECURE_MSG_RECEIVE	71
5.1.19 WFS_CMD_PIN_GET_JOURNAL	73
5.1.20 WFS_CMD_PIN_IMPORT_KEY_EX.....	74
5.1.21 WFS_CMD_PIN_ENC_IO.....	77
5.1.22 WFS_CMD_PIN_HSM_INIT.....	79
5.1.23 WFS_CMD_PIN_SECUREKEY_ENTRY	80
5.1.24 WFS_CMD_PIN_GENERATE_KCV	83
5.1.25 WFS_CMD_PIN_SET_GUIDANCE_LIGHT	84
5.1.26 WFS_CMD_PIN_MAINTAIN_PIN.....	85
5.1.27 WFS_CMD_PIN_KEYPRESS_BEEP	86
5.1.28 WFS_CMD_PIN_SET_PINBLOCK_DATA	87
5.1.29 WFS_CMD_PIN_SET_LOGICAL_HSM.....	88
5.1.30 WFS_CMD_PIN_IMPORT_KEYBLOCK	90

5.1.31	WFS_CMD_PIN_POWER_SAVE_CONTROL	91
5.2	Common commands for Remote Key Loading Schemes	92
5.2.1	WFS_CMD_PIN_START_KEY_EXCHANGE	92
5.3	Remote Key Loading Using Signatures	93
5.3.1	WFS_CMD_PIN_IMPORT_RSA_PUBLIC_KEY	93
5.3.2	WFS_CMD_PIN_EXPORT_RSA_ISSUER_SIGNED_ITEM	96
5.3.3	WFS_CMD_PIN_IMPORT_RSA_SIGNED_DES_KEY	98
5.3.4	WFS_CMD_PIN_GENERATE_RSA_KEY_PAIR	101
5.3.5	WFS_CMD_PIN_EXPORT_RSA_EPP_SIGNED_ITEM	103
5.4	Remote Key Loading with Certificates	105
5.4.1	WFS_CMD_PIN_LOAD_CERTIFICATE	105
5.4.2	WFS_CMD_PIN_GET_CERTIFICATE	106
5.4.3	WFS_CMD_PIN_REPLACE_CERTIFICATE	107
5.4.4	WFS_CMD_PIN_IMPORT_RSA_ENCIPHERED_PKCS7_KEY	108
5.5	EMV	110
5.5.1	WFS_CMD_PIN_EMV_IMPORT_PUBLIC_KEY	110
5.5.2	WFS_CMD_PIN_DIGEST	113
6.	Events	114
6.1	WFS_EXEE_PIN_KEY	114
6.2	WFS_SRVE_PIN_INITIALIZED	115
6.3	WFS_SRVE_PIN_ILLEGAL_KEY_ACCESS	116
6.4	WFS_SRVE_PIN_OPT_REQUIRED	117
6.5	WFS_SRVE_PIN_CERTIFICATE_CHANGE	118
6.6	WFS_SRVE_PIN_HSM_TDATA_CHANGED	119
6.7	WFS_SRVE_PIN_HSM_CHANGED	120
6.8	WFS_EXEE_PIN_ENTERDATA	121
6.9	WFS_SRVE_PIN_DEVICEPOSITION	122
6.10	WFS_SRVE_PIN_POWER_SAVE_CHANGE	123
7.	C - Header File	124
8.	Appendix-A	140
8.1	Remote Key Loading Using Signatures	141
8.1.1	RSA Data Authentication and Digital Signatures	141
8.1.2	RSA Secure Key Exchange using Digital Signatures	142
8.1.3	Initialization Phase – Signature Issuer and ATM PIN	144
8.1.4	Initialization Phase – Signature Issuer and Host	145
8.1.5	Key Exchange – Host and ATM PIN	146
8.1.6	Key Exchange (with random number) – Host and ATM PIN	147
8.1.7	Enhanced RKL, Key Exchange (with random number) – Host and ATM PIN	148
8.1.8	Default Keys and Security Item loaded during manufacture	149
8.2	Remote Key Loading Using Certificates	150
8.2.1	Certificate Exchange and Authentication	150
8.2.2	Remote Key Exchange	151
8.2.3	Replace Certificate	152
8.2.4	Primary and Secondary Certificates	153
8.3	German ZKA GeldKarte	154
8.3.1	How to use the SECURE_MSG commands	154
8.3.2	Protocol WFS_PIN_PROTISOAS	155
8.3.3	Protocol WFS_PIN_PROTISOLZ	156
8.3.4	Protocol WFS_PIN_PROTISOPS	157

8.3.5	Protocol WFS_PIN_PROTCHIPZKA	158
8.3.6	Protocol WFS_PIN_PROTRAWDATA	159
8.3.7	Protocol WFS_PIN_PROTPBM	160
8.3.8	Protocol WFS_PIN_PROTHSMLDI	161
8.3.9	Protocol WFS_PIN_PROTGENAS	162
8.3.10	Protocol WFS_PIN_PROTCHIPPINCHG.....	165
8.3.11	Protocol WFS_PIN_PROTPINCMF.....	166
8.3.12	Protocol WFS_PIN_PROTISOPINCHG	167
8.3.13	Command Sequence	168
8.4	EMV Support.....	175
8.4.1	Keys loading.....	175
8.4.2	PIN block management	177
8.4.3	SHA-1 Digest	178
8.5	French Cartes Bancaires.....	179
8.5.1	Data Structure for WFS_CMD_PIN_ENC_IO	179
8.5.2	Command Sequence	181
8.6	Secure Key Entry	183
8.6.1	Keyboard Layout.....	183
8.6.2	Command Usage	187
9.	Appendix-B (Country Specific WFS_CMD_PIN_ENC_IO protocols)	188
9.1	Luxemburg Protocol.....	188
9.1.1	WFS_CMD_ENC_IO_LUX_LOAD_APPKEY.....	190
9.1.2	WFS_CMD_ENC_IO_LUX_GENERATE_MAC	192
9.1.3	WFS_CMD_ENC_IO_LUX_CHECK_MAC.....	193
9.1.4	WFS_CMD_ENC_IO_LUX_BUILD_PINBLOCK	194
9.1.5	WFS_CMD_ENC_IO_LUX_DECRYPT_TDES	195
9.1.6	WFS_CMD_ENC_IO_LUX_ENCRYPT_TDES	196
9.1.7	Luxemburg-specific Header File.....	197
10.	Appendix-C (Standardized <i>lpszExtra</i> fields).....	200
10.1	WFS_INF_PIN_STATUS.....	200
10.2	WFS_INF_PIN_CAPABILITIES	201

Foreword

This CWA is revision 3.10 of the XFS interface specification.

The CEN/ISSS XFS Workshop gathers suppliers as well as banks and other financial service companies. A list of companies participating in this Workshop and in support of this CWA is available from the CEN/ISSS Secretariat.

This CWA was formally approved by the XFS Workshop meeting on 2007-11-29. The specification is continuously reviewed and commented in the CEN/ISSS Workshop on XFS. It is therefore expected that an update of the specification will be published in due time as a CWA, superseding this revision 3.10.

The CWA is published as a multi-part document, consisting of:

Part 1: Application Programming Interface (API) - Service Provider Interface (SPI) - Programmer's Reference

Part 2: Service Classes Definition - Programmer's Reference

Part 3: Printer and Scanning Device Class Interface - Programmer's Reference

Part 4: Identification Card Device Class Interface - Programmer's Reference

Part 5: Cash Dispenser Device Class Interface - Programmer's Reference

Part 6: PIN Keypad Device Class Interface - Programmer's Reference

Part 7: Check Reader/Scanner Device Class Interface - Programmer's Reference

Part 8: Depository Device Class Interface - Programmer's Reference

Part 9: Text Terminal Unit Device Class Interface - Programmer's Reference

Part 10: Sensors and Indicators Unit Device Class Interface - Programmer's Reference

Part 11: Vendor Dependent Mode Device Class Interface - Programmer's Reference

Part 12: Camera Device Class Interface - Programmer's Reference

Part 13: Alarm Device Class Interface - Programmer's Reference

Part 14: Card Embossing Unit Device Class Interface - Programmer's Reference

Part 15: Cash-In Module Device Class Interface - Programmer's Reference

Part 16: Card Dispenser Device Class Interface - Programmer's Reference

Part 17: Barcode Reader Device Class Interface - Programmer's Reference

Part 18: Item Processing Module Device Class Interface - Programmer's Reference

Parts 19 - 24: Reserved for future use.

Part 25: Identification Card Device Class Interface - PC/SC Integration Guidelines

Parts 26 - 28: Reserved for future use.

Parts 29 through 47 constitute an optional addendum to this CWA. They define the integration between the SNMP standard and the set of status and statistical information exported by the Service Providers.

Part 29: XFS MIB Architecture and SNMP Extensions - Programmer's Reference

Part 30: XFS MIB Device Specific Definitions - Printer Device Class

Part 31: XFS MIB Device Specific Definitions - Identification Card Device Class

Part 32: XFS MIB Device Specific Definitions - Cash Dispenser Device Class

Part 33: XFS MIB Device Specific Definitions - PIN Keypad Device Class

Part 34: XFS MIB Device Specific Definitions - Check Reader/Scanner Device Class

Part 35: XFS MIB Device Specific Definitions - Depository Device Class

Part 36: XFS MIB Device Specific Definitions - Text Terminal Unit Device Class

Part 37: XFS MIB Device Specific Definitions - Sensors and Indicators Unit Device Class

Part 38: XFS MIB Device Specific Definitions - Camera Device Class

Part 39: XFS MIB Device Specific Definitions - Alarm Device Class

Part 40: XFS MIB Device Specific Definitions - Card Embossing Unit Class

Part 41: XFS MIB Device Specific Definitions – Cash-In Module Device Class

Part 42: Reserved for future use.

Part 43: XFS MIB Device Specific Definitions - Vendor Dependent Mode Device Class

Part 44: XFS MIB Application Management

Part 45: XFS MIB Device Specific Definitions - Card Dispenser Device Class

Part 46: XFS MIB Device Specific Definitions - Barcode Reader Device Class

Part 47: XFS MIB Device Specific Definitions - Item Processing Module Device Class

Parts 48 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Service Provider Interface (SPI) - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 62: Printer Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.02 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.03 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.01 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.0 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.02 (CWA 14050) to Version 3.10 (this CWA) - Programmer's Reference

In addition to these Programmer's Reference specifications, the reader of this CWA is also referred to a complementary document, called Release Notes. The Release Notes contain clarifications and explanations on the CWA specifications, which are not requiring functional changes. The current version of the Release Notes is available online from <http://www.cen.eu/iss/Workshop/XFS>.

The information in this document represents the Workshop's current views on the issues discussed as of the date of publication. It is furnished for informational purposes only and is subject to change without notice. CEN/ISSS makes no warranty, express or implied, with respect to this document.

This CEN Workshop Agreement is publicly available as a reference document from the National Members of CEN : AENOR, AFNOR, ASRO, BDS, BSI, CSNI, CYS, DIN, DS, ELOT, EVS, IBN, IPQ, IST, LVS, LST, MSA, MSZT, NEN, NSAI, ON, PKN, SEE, SIS, SIST, SFS, SN, SNV, SUTN and UNI.

Comments or suggestions from the users of the CEN Workshop Agreement are welcome and should be addressed to the CEN Management Centre.

Revision History:

1.0	May 24, 1993	Initial release of API and SPI specification.
1.11	February 3, 1995	Separation of specification into separate documents for API/SPI and service class definitions.
2.0	November 11, 1996	Update release encompassing the self-service environment.
3.0	October 18, 2000	Update release encompassing: New commands to support the German ZKA chip card standard. Support of Banksys Security Control Module. Added clarification note for Pin format 3624. Added WFS_CMD_PIN_ENC_IO, which is currently used for the Swiss proprietary protocol only. Double and triple zero clarification in WFS_CMD_PIN_GET_DATA. Key deletion in WFS_CMD_PIN_IMPORT_KEY inserted. For a detailed description see CWA 14050-20:2000 PIN Migration from Version 2.0 to Version 3.0.
3.02	May 21, 2003	Update release encompassing: New commands to support EMV, GIE-CB, Remote Key Loading (Signature and Certificate), OPT, MAA MAC, and Multiple-Part Key Loading. Added clarification notes on WFS_PIN_CRYPTTRIDESMAC to the WFS_INF_CAPABILITES and WFS_CMD_PIN_CRYPT. For a detailed description see CWA 14050-27:2003 PIN Migration from Version 3.0 to Version 3.02.
3.03	September 24, 2004	Update release encompassing: New command to support secure manual encryption key entry. New command to support the generation of a Key Check Value for a previously loaded symmetric key. Added support for the ZKA PROTGENAS. Existing command descriptions were modified to describe the way in which Signatures can be used to authenticate public key deletion within the RKL Signature scheme. For a detailed description see CWA 14050-42:2005 PIN Migration from Version 3.02 to Version 3.03.
3.10	November 29, 2007	For a description of changes see CWA 15748-65:2007 PIN Migration from Version 3.03 (see CWA 14050) to Version 3.10.

1. Introduction

1.1 Background to Release 3.10

The CEN/ISSS XFS Workshop aims to promote a clear and unambiguous specification defining a multi-vendor software interface to financial peripheral devices. The XFS (eXtensions for Financial Services) specifications are developed within the CEN/ISSS (European Committee for Standardization/Information Society Standardization System) Workshop environment. CEN/ISSS Workshops aim to arrive at a European consensus on an issue that can be published as a CEN Workshop Agreement (CWA).

The CEN/ISSS XFS Workshop encourages the participation of both banks and vendors in the deliberations required to create an industry standard. The CEN/ISSS XFS Workshop achieves its goals by focused sub-groups working electronically and meeting quarterly.

Release 3.10 of the XFS specification is based on a C API and is delivered with the continued promise for the protection of technical investment for existing applications. This release of the XFS specification has been prompted by a series of factors.

There has been a technical imperative to extend the scope of the existing specification to include new devices, such as the Barcode Reader, Card Dispenser and Item Processing Module.

Similarly, there has also been pressure, through implementation experience and additional requirements, to extend the functionality and capabilities of the existing devices covered by the specification.

1.2 XFS Service-Specific Programming

The service classes are defined by their service-specific commands and the associated data structures, error codes, messages, etc. These commands are used to request functions that are specific to one or more classes of Service Providers, but not all of them, and therefore are not included in the common API for basic or administration functions.

When a service-specific command is common among two or more classes of Service Providers, the syntax of the command is as similar as possible across all services, since a major objective of XFS is to standardize function codes and structures for the broadest variety of services. For example, using the **WFSExecute** function, the commands to read data from various services are as similar as possible to each other in their syntax and data structures.

In general, the specific command set for a service class is defined as a superset of the specific capabilities likely to be provided by the developers of the services of that class; thus any particular device will normally support only a subset of the defined command set.

There are three cases in which a Service Provider may receive a service-specific command that it does not support:

The requested capability is defined for the class of Service Providers by the XFS specification, the particular vendor implementation of that service does not support it, and the unsupported capability is **not** considered to be fundamental to the service. In this case, the Service Provider returns a successful completion, but does no operation. An example would be a request from an application to turn on a control indicator on a passbook printer; the Service Provider recognizes the command, but since the passbook printer it is managing does not include that indicator, the Service Provider does no operation and returns a successful completion to the application.

The requested capability is defined for the class of Service Providers by the XFS specification, the particular vendor implementation of that service does not support it, and the unsupported capability **is** considered to be fundamental to the service. In this case, a **WFS_ERR_UNSUPP_COMMAND** error is returned to the calling application. An example would be a request from an application to a cash dispenser to dispense coins; the Service Provider recognizes the command but, since the cash dispenser it is managing dispenses only notes, returns this error.

The requested capability is **not** defined for the class of Service Providers by the XFS specification. In this case, a **WFS_ERR_INVALID_COMMAND** error is returned to the calling application.

This design allows implementation of applications that can be used with a range of services that provide differing subsets of the functionalities that are defined for their service class. Applications may use the **WFSGetInfo** and **WFSAsyncGetInfo** commands to inquire about the capabilities of the service they are about to use, and modify their behavior accordingly, or they may use functions and then deal with **WFS_ERR_UNSUPP_COMMAND** error returns to make decisions as to how to use the service.

2. Pin Keypad

This section describes the application program interface for personal identification number keypads (PIN pads) and other encryption/decryption devices. This description includes definitions of the service-specific commands that can be issued, using the **WFSAsyncExecute**, **WFSExecute**, **WFSGetInfo** and **WFSAsyncGetInfo** functions.

This section describes the general interface for the following functions:

- Administration of encryption devices
- Loading of encryption keys
- Encryption / decryption
- Entering Personal Identification Numbers (PINs)
- PIN verification
- PIN block generation (encrypted PIN)
- Clear text data handling
- Function key handling
- PIN presentation to chipcard
- Read and write safety critical Terminal Data from/to HSM
- HSM and Chipcard Authentication
- EMV 4.0 PIN blocks, EMV 4.0 public key loading, static and dynamic data verification

If the PIN Pad device has local display capability, display handling should be handled using the Text Terminal Unit (TTU) interface.

The adoption of this specification does not imply the adoption of a specific security standard.

Important Notes:

- This revision of this specification does not define all key management procedures; some key management is still vendor-specific.
- Key space management is customer-specific, and is therefore handled by vendor-specific mechanisms.
- Only numeric PIN pads are handled in this specification.

This specification also supports the Hardware Security Module (HSM), which is necessary for the German ZKA Electronic Purse transactions. Furthermore the HSM stores terminal specific data.

This data will be compared against the message data fields (Sent and Received ISO8583 messages) prior to HSM-MAC generation/verification. HSM-MACs are generated/verified only if the message fields match the data stored.

Keys used for cryptographic HSM functions are stored separate from other keys. This must be considered when importing keys.

This version of PinPad complies to the current ZKA specification 3.0. It supports loading and unloading against card account for both card types (Type 0 and Type 1) of the ZKA electronic purse. It also covers the necessary functionality for 'Loading against other legal tender'.

Key values are passed to the API as binary hexadecimal values, for example:

0123456789ABCDEF = 0x01 0x23 0x45 0x67 0x89 0xAB 0xCD 0xEF

When hex values are passed to the API within strings, the hex digits 0xA to 0xF can be represented by characters in the ranges 'a' to 'f' or 'A' to 'F'.

The following commands and events were initially added to support the German ZKA standard, but may also be used for other national standards:

- WFS_INF_PIN_HSM_TDATA
- WFS_CMD_PIN_HSM_SET_TDATA
- WFS_CMD_PIN_SECURE_MSG_SEND

- WFS_CMD_PIN_SECURE_MSG_RECEIVE
- WFS_CMD_PIN_GET_JOURNAL
- WFS_SRVE_PIN_OPT_REQUIRED
- WFS_CMD_PIN_HSM_INIT
- WFS_SRVE_PIN_HSM_TDATA_CHANGED

3. References

1. XFS Application Programming Interface (API)/Service Provider Interface (SPI), Programmer's Reference Revision 3.10
2. RSA Laboratories, PKCS #7: <i>Cryptographic Message Syntax Standard</i> . Version 1.5, November 1993
3. SHA-1 Hash algorithm ANSI 9:30:2-1993: <i>Public Key Cryptography for Financial Services Industry Part 2</i>
4. EMVCo, EMV2000 Integrated Circuit Card Specification for Payment Systems, Book 2 – Security and Key Management, Version 4.0, December 2000
5. Europay International, EPI CA Module Technical – Interface specification Version 1.4
6. ZKA / Bank-Verlag, Köln, Schnittstellenspezifikation für die ec-Karte mit Chip, Online-Personalisierung von Terminal-HSMs, Version 3.0, 2. 4. 1998
7. ZKA / Bank-Verlag, Köln, Schnittstellenspezifikation für die ZKA-Chipkarte, Online-Vor-Initialisierung und Online-Anzeige einer Außerbetriebnahme von Terminal-HSMs, Version 1.0, 04.08.2000
8. 473x Programmers Reference Volume 1 - TP-820399-001A
9. 473x Programmers Reference Volume 2 - TP-820403-001A
10. 473x Programmers Reference Volume 3 - TP-820400-001A
11. 473x Programmers Reference Volume 4 - TP-820404-001A
12. 473x P-Model Programmers Reference - TP-820397-001A
13. 473x Log Reference Guide - TP-820398-001A
14. Diebold's Specification for support of Online Preinitialization and Personalization of Terminal HSMs (OPT) and support for the PAC/MAC standards for the 473x Protocol, Diebold USA, Revision 1.10, revised on May 2002
15. Groupement des Cartes Bancaires "CB", Description du format et du contenu des données cryptographiques échangées entre GAB et GDG, Version 1.3 / Octobre 2002
16. ITU-T Recommendation X.690 – ASN.1 encoding rules (also published as ISO/IEC International Standard 8825-1), 1997
17. German ZKA specification, published by: Bank-Verlag Koeln, Post Box 300191, 50771 Cologne, Germany; Tel: +49 221 5490-0; Fax: +49 221 5490-120
18. Banksys document "SCM DKH Manual Rel 2.x"
19. Diebold's and IBM's Specification for support of Online Preinitialization and Personalization of Terminal HSMs (OPT) and support for the PAC/MAC standards for the 473x Protocol, Diebold USA, Revision 1.8, revised on Jan-03-2001
20. ANSI X3.92, American National Standard for Data Encryption Algorithm (DEA), American National Standards Institute, 1983
21. ANSI X9.8-1995, Banking – Personal Identification Number Management and Security, Part 1 + 2, American National Standards Institute
22. ISO 9564-1, Banking – Personal Identification Number management and security, Part 1, First Edition 1991-12-15, International Organization for Standardization
23. ISO 9564-2, Banking – Personal Identification Number management and security, Part 2, First Edition 1991-12-15, International Organization for Standardization
24. IBM, Common Cryptographic Architecture: Cryptographic Application Programming Interface, SC40-1675-1, IBM Corp., Nov 1990
25. R.L: Rivest, A. Shamir, and L.M. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, v. 21, n.2, Feb 1978, pp. 120-126
26. Security for Computer Networks by Donald W. Davies & William L. Price, Second Edition, John Wiley & Sons, 1989
27. Regelwerk für das deutsche ec-Geldautomaten-System, Stand: 22. Nov. 1999
28. Bank-Verlag, Köln, Autorisierungszentrale GA/POS der privaten Banken, Spezifikation für GA-Betreiber, Version 3.12, 31. Mai 2000
29. dvg Hannover, Schnittstellenbeschreibung für Autorisierungsanfragen bei nationalen GA-Verfügungen unter Verwendung der Spur 3, Version 2.5, Stand: 15.03.2000
30. dvg Hannover, Schnittstellenbeschreibung für Autorisierungsanfragen bei internationalen Verfügungen unter Verwendung der Spur 2, Version 2.6, Stand: 30.03.2000
31. ZKA / Bank-Verlag, Köln, Schnittstellenspezifikation für die ec-Karte mit Chip, Geldkarte Ladeterminals, Version 3.0, 2. 4. 1998
32. ISO/IEC 9797-1: 1999
33. ISO 8731-2
34. ZKA / Bank-Verlag, Köln, Schnittstellenspezifikation für die ec-Karte mit Chip PIN-Änderungsfunktion, Version 3.0, 12.05.1999
35. ANS X9 TR-31 2005, Interoperable Secure Key Exchange Key Block Specification for Symmetric Algorithms