CEN

WORKSHOP

AGREEMENT

CWA 16926-65

February 2020

ICS 35.240.15; 35.240.40; 35.200

English version

Extensions for Financial Services (XFS) interface specification - Release 3.40 - Part 65: PIN Device Class Interface - Migration from version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

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

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Ref. No.:CWA 16926-65:2020 E

Table of Contents

European Foreword

This CEN Workshop Agreement has been developed in accordance with the CEN-CENELEC Guide 29 "CEN/CENELEC Workshop Agreements – The way to rapid consensus" and with the relevant provisions of CEN/CENELEC Internal Regulations – Part 2. It was approved by a Workshop of representatives of interested parties on 2019-10-08, the constitution of which was supported by CEN following several public calls for participation, the first of which was made on 1998-06-24. However, this CEN Workshop Agreement does not necessarily include all relevant stakeholders.

The final text of this CEN Workshop Agreement was provided to CEN for publication on 2019-12-12.

The following organizations and individuals developed and approved this CEN Workshop Agreement:

- ATM Japan LTD
- AURIGA SPA
- BANK OF AMERICA
- CASHWAY TECHNOLOGY
- CHINAL ECTRONIC FINANCIAL EQUIPMENT SYSTEM CO.
- CIMA SPA
- CLEAR2PAY SCOTLAND LIMITED
- DIEBOLD NIXDORF
- EASTERN COMMUNICATIONS CO. LTD – EASTCOM
- FINANZ INFORMATIK
- FUJITSU FRONTECH LIMITED
- FUJITSU TECHNOLOGY
- GLORY LTD
- GRG BANKING EQUIPMENT HK CO LTD
- HESS CASH SYSTEMS GMBH & CO. KG
- HITACHI OMRON TS CORP.
- HYOSUNG TNS INC
- JIANGSU GUOGUANG ELECTRONIC INFORMATION TECHNOLOGY
- KAL
- KEBA AG
- NCR FSG
- NEC CORPORATION
- OKI ELECTRIC INDUSTRY SHENZHEN

- OKI ELECTRONIC INDUSTRY CO
- PERTO S/A
- REINER GMBH & CO KG
- SALZBURGER BANKEN SOFTWARE
- SIGMA SPA
- TEB
- ZIJIN FULCRUM TECHNOLOGY CO

It is possible that some elements of this CEN/CWA may be subject to patent rights. The CEN-CENELEC policy on patent rights is set out in CEN-CENELEC Guide 8 “Guidelines for Implementation of the Common IPR Policy on Patents (and other statutory intellectual property rights based on inventions)”. CEN shall not be held responsible for identifying any or all such patent rights.

The Workshop participants have made every effort to ensure the reliability and accuracy of the technical and non-technical content of CWA 16926-65, but this does not guarantee, either explicitly or implicitly, its correctness. Users of CWA 16926-65 should be aware that neither the Workshop participants, nor CEN can be held liable for damages or losses of any kind whatsoever which may arise from its application. Users of CWA 16926-65 do so on their own responsibility and at their own risk.

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

Part 19: Biometrics Device Class Interface - Programmer's Reference

Parts 20 - 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

Part 48: XFS MIB Device Specific Definitions - Biometrics Device Class

Parts 49 - 60 are reserved for future use.

Part 61: Application Programming Interface (API) - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Service Provider Interface (SPI) - Programmer's Reference

Part 62: Printer and Scanning Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 63: Identification Card Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 64: Cash Dispenser Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 65: PIN Keypad Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 66: Check Reader/Scanner Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 67: Depository Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 68: Text Terminal Unit Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 69: Sensors and Indicators Unit Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 70: Vendor Dependent Mode Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 71: Camera Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 72: Alarm Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 73: Card Embossing Unit Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 74: Cash-In Module Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 75: Card Dispenser Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 76: Barcode Reader Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (this CWA) - Programmer's Reference

Part 77: Item Processing Module Device Class Interface - Migration from Version 3.30 (CWA 16926) to Version 3.40 (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: https://www.cen.eu/work/Sectors/Digital_society/Pages/WSXFS.aspx.

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

1. Migration Information

XFS 3.40 has been designed to minimize backwards compatibility issues. This document highlights the changes made to the PIN device class between version 3.30 and 3.40, by highlighting the additions and deletions to the text.

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 PIN pad 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

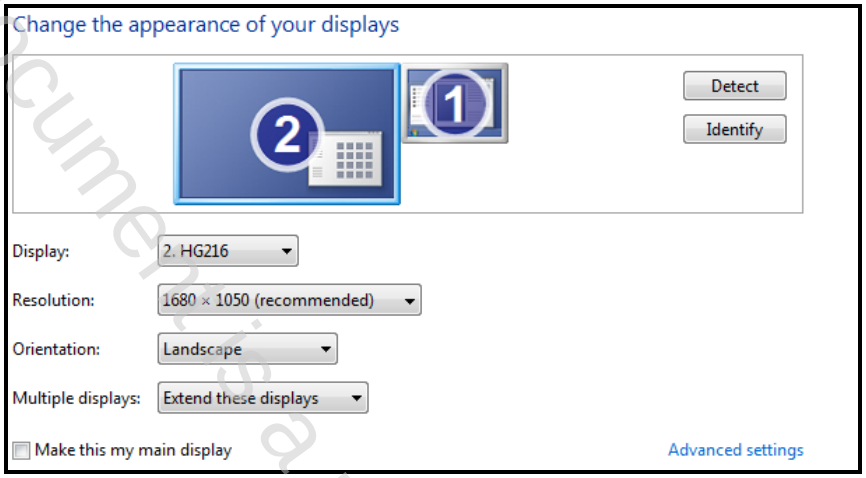
- WFS_CMD_PIN_SECURE_MSG_RECEIVE
- WFS_CMD_PIN_GET_JOURNAL
- WFS_SRVE_PIN_OPT_REQUIRED
- WFS_CMD_PIN_HSM_INIT
- WFS_SRVE_PIN_HSM_TDATA_CHANGED

Certain levels of the PCI EPP security standards specify that if a key encryption key is deleted or replaced, then all keys in the hierarchy under that key encryption key are also removed. Key encryption keys have the WFS_PIN_USEKEYENCKEY type of access. Applications can check impact of key deletion using WFS_INF_PIN_KEY_DETAIL or WFS_INF_PIN_KEY_DETAIL_EX.

2.1 Encrypting Touch Screen (ETS)

An encrypting touch screen device is a touch screen securely attached to a cryptographic device. It can be used as an alternative to an encrypting pin pad (EPP). It supports key management, encryption and decryption.

It is assumed that the ETS is a combined device. It overlays a display monitor which is used to display lead-through for a transaction. It is assumed that the display monitor is part of the Windows desktop, and can be the Windows primary monitor or any other monitor on the desktop. E.g. the following diagram shows 2 monitors extended across the desktop, with monitor 1 being the primary monitor and the ETS being overlaid on monitor 2 whose origin is (-1680.0).



The touch screen can optionally be used as a “mouse” for application purposes, while XFS PIN operations are not in progress or optionally when non-secure XFS PIN commands are in progress.

The CEN interface supports two types of ETS

- Those which activate touch areas defined by the application.
- Those which activate a random variation of touch areas defined by the application.

The Service Provider, when reporting its capabilities, reports the absolute position of the ETS in Windows desktop coordinates. This allows the application to locate the ETS device in a multi-monitor system and relate it to a monitor on the desktop.

At any point in time, a single touch area of the ETS can operate in one of 4 modes:-

- **Mouse mode** - a “touch” simulates a mouse click. This mode is optional. This may not be supported by some ETS devices. Configuration of the click is vendor specific. e.g. WM_LBUTTONDOWN. This is also the mode that, if supported, is active when none of the other modes are active.
- **XFS Data mode** - a “touch” maps to an XFS key and the value of the key is returned in an event (as in clear numeric entry using WFS_CMD_PIN_GET_DATA).
- **XFS PIN mode** - a “touch” maps to an XFS key and the value of the key is returned in an event only if the key pressed is not WFS_PIN_FK_0 through WFS_PIN_FK_9 (as in PIN entry using WFS_CMD_PIN_GET_PIN).
- **XFS Secure mode** - a “touch” maps to an XFS key and the value of the key is returned in an event only if the key pressed is not WFS_PIN_FK_0 through WFS_PIN_FK_9 and not WFS_PIN_FK_A through WFS_PIN_FK_F (as in key entry using WFS_CMD_PIN_SECUREKEY_ENTRY).

The following concepts are introduced to define the relationship between the monitor and the ETS:-

- **Touch Key** – an area of the monitor which reacts to touch in XFS Data, PIN and Secure modes.
- **Touch Frame** – an area of the monitor onto which Touch Keys can be placed. There can be one or more Touch Frames. There may be just one Touch Frame which covers the whole monitor. Areas within a Touch Frame, not defined as a Touch Key, do not react to touch. Generally in XFS PIN and Secure modes, there would be only one Touch Frame covering the whole monitor. An empty Touch Frame disables that part of the monitor.

- **Mouse area** – an area outside of all Touch Frames in which touches behave like a mouse
- Thus XFS Data, PIN and Secure modes operate in a single Touch Frame or multiple Touch Frames. Mouse mode operates outside a Touch Frame, and is optional.

Note that there is a perceived risk in separating the drawing functionality from the touch functionality, but this type of risk is present in today's keyboard based systems. e.g. An application can draw on a monitor to prompt the user to enter a PIN and then enables the EPP for clear data entry. So the risk is no different than with an EPP – the application has to be trusted.

Depending upon the type of device, the application must then either inform the Service Provider as to the active key positions in the form of Touch Frames and Touch Keys using the WFS_CMD_PIN_DEFINE_LAYOUT command, or obtain them from the Service Provider using the WFS_INF_PIN_GET_LAYOUT command. This collection is now referred to as a "Touch Keyboard definition".

The application then uses the normal PIN commands to enable the touch keyboard definition on the ETS device:

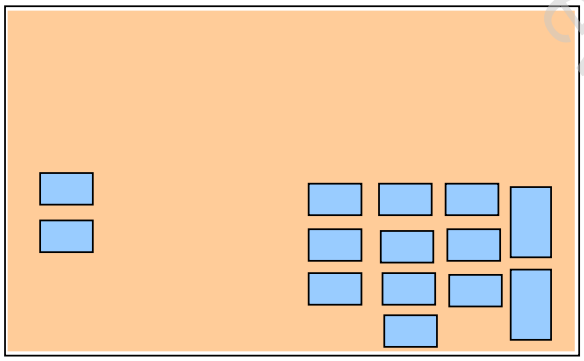
- PIN entry WFS_CMD_PIN_GET_PIN
- Clear data entry WFS_CMD_PIN_GET_DATA
- Secure key entry WFS_CMD_PIN_SECUREKEY_ENTRY

These commands are referred to as "keyboard entry commands" throughout the remainder of this document.

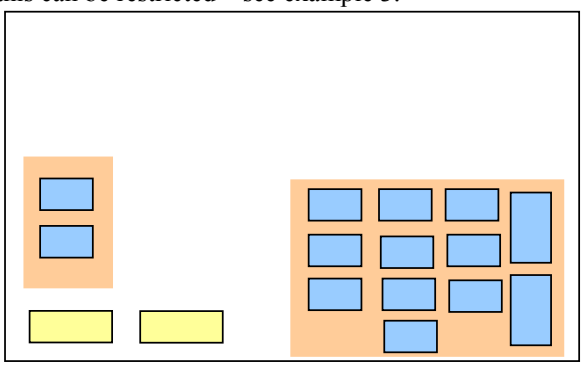
PCI compliance means that WFS_CMD_PIN_GET_PIN and WFS_CMD_PIN_SECUREKEY_ENTRY can only be used with a single Touch Frame that covers the entire monitor. i.e. Mouse mode cannot be mixed with either XFS PIN or Secure mode. If a Touch Key (or areas) is defined for an XFS key value and that key value is not subsequently specified as active in a WFS_CMD_PIN_GET_PIN, WFS_CMD_PIN_GET_DATA or WFS_CMD_PIN_SECUREKEY_ENTRY command, then the Touch Key is made inactive.

Layouts defined with the WFS_CMD_PIN_DEFINE_LAYOUT command are persistent.

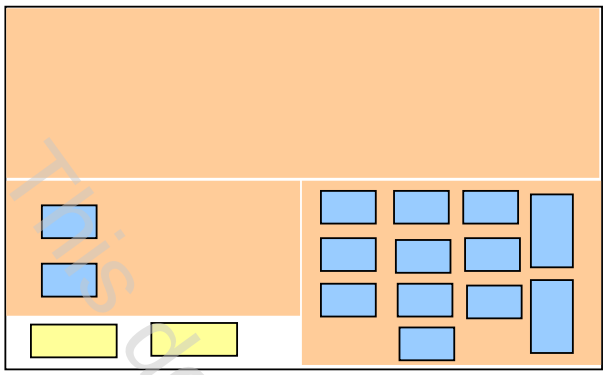
Example 1 – this screen only uses XFS Data mode – the entire screen is a Touch Frame. Mouse mode is not used.



Example 2 – this shows a monitor with two Touch Frames and 14 Touch Keys. The space within the Touch Frames not defined by a Touch Key are inactive (do not respond to touch). All areas outside a Touch Frame operate in Mouse mode. This example shows two Mouse mode "keys". e.g. Windows "Button", HTML "BUTTON" or a custom control. Other touches in Mouse mode are normally dealt with by the application event engine. However, this can be restricted – see example 3.



Example 3 – this screen uses Mouse and XFS Data modes – Mouse mode is used only in a restricted area. The touch keyboard definition has 3 frames. Frame 1 has no Touch Keys. Frame 2 has 2 Touch Keys; Frame 3 has 12 Touch Keys.



3. References

1. XFS Application Programming Interface (API)/Service Provider Interface (SPI), Programmer's Reference Revision 3.40
2. RSA Laboratories, PKCS #7: *Cryptographic Message Syntax Standard.* Version 1.5, November 1993
3. SHA-1 Hash algorithm ANSI X9.30-2:1993, *Public Key Cryptography for Financial Services Industry Part2*
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 cryprographiques 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 2018, Interoperable Secure Key Exchange Key Block Specification for Symmetric Algorithms

36. Oliself2 Specifiche Tecniche, PIN Block Detail for WFS_PIN_FORMAP
37. PCI Security Standards Council PCI PTS approval list https://www.pcisecuritystandards.org/approved_companies_providers/approved_pin_transaction_security.php
38. ISO 16609:2004 Financial Services – Requirements for message authentication using symmetric techniques
39. Australian Standard 2805.4 Electronic Funds Transfer – Requirements for Interface Part 4 – Message Authentication
40. ISO/IEC 10118-3:2004 Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions
41. FIPS 180-2 Secure Hash Signature Standard
42. ANS X9 TR-34 2012, Interoperable Method for Distribution of Symmetric Keys using Asymmetric Techniques: Part 1 – Using Factoring-Based Public Key Cryptography Unilateral Key Transport
43. Password industry standard of the People's Republic of China GM/T 0002-2012, GM/T 0003.1-2012, GM/T 0003.2-2012, GM/T 0003.3-2012, GM/T 0003.4-2012, GM/T 0003.5-2012, GM/T 0004-2012.
44. Financial industry standard of the People’s Republic of China PBOC3.0 JR/T 0025.17-2013.
45. ANS X9.24-1:2009, Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques
46. ISO/IEC 18033-3:2010 Information technology -- Security techniques -- Encryption algorithms -- Part 3: Block ciphers
47. FIPS PUB 197: Advanced Encryption Standard (AES) 48. ISO/IEC 9564-1:2017 Financial services – Personal Identification Number (PIN) management and security – Part 1: Basic principles and requirements for PINs in card-based systems
49. NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation
50. NIST Special Publication 800-38E: Recommendation for Block Cipher Modes of Operation: the XTS-AES Mode for Confidentiality on Storage Devices
51. Deutsche Kreditwirtschaft AES specification published by: The German Banking Industry Committee (GBIC) : Contact: info@die-dk.de