

ZENTRALER KREDITAUSSCHUSS

MITGLIEDER: BUNDESVERBAND DER DEUTSCHEN VOLKSBANKEN UND RAIFFEISENBANKEN E.V. BERLIN • BUNDESVERBAND DEUTSCHER BANKEN E. V. BERLIN • BUNDESVERBAND ÖFFENTLICHER BANKEN DEUTSCHLANDS E. V. BERLIN • DEUTSCHER SPARKASSEN- UND GIROVERBANDE. V. BERLIN-BONN • VERBAND DEUTSCHER HYPOTHEKENBANKEN E. V. BERLIN

Anlage 2

der Schnittstellenspezifikation für die Datenfernübertragung zwischen Kunde und Kreditinstitut gemäß DFÜ-Abkommen

„Spezifikation für die FTAM-Anbindung“

Version 2.0 vom 3. November 2005

Änderungsverfolgung (gegenüber Version 1.1 vom 31.3.2005)

Kapitel	Beschluss-Datum	Art *	Beschreibung	Inkrafttreten
1.5, 1.7.3	03.11.2005	E	Aufnahme des Antwortcodes 52 mit der Textmeldung „EU-Version wird nicht mehr unterstützt“ Aufnahme der Textmeldung inklusive neu vergebener Textnummer 38 in Kapitel 1.7.3	01.01.2006
1.7	03.11.2005	E	Neues Unterkapitel zum Einfügen individueller Texte im Kundenprotokoll (Kapitelnummer 1.7.2.4). Die Nummerierung des folgenden Unterkapitels der gleichen Ebene verschiebt sich entsprechend.	01.01.2006

* F = Fehler; Ä = Änderung; K = Klarstellung; E = Erweiterung; L = Löschung

Inhaltsverzeichnis

1 Standards für die Kommunikation.....	1
1.1 Anwendungsprotokoll Kunden-/Bankrechner	1
1.1.1 Remote-User-ID	1
1.1.2 Remote-Password.....	1
1.1.3 Remote-Filename.....	1
1.2 FTAM-Verfahrensspezifikation.....	3
1.2.1 Virtual Filestore	4
1.2.2 FTAM Units	7
1.2.3 Anwendungsschicht ACSE	11
1.2.4 Presentation Layer	12
1.2.5 Session Layer	13
1.2.6 Transport Layer.....	14
1.2.7 Network Layer	14
1.2.8 Literaturverzeichnis.....	16
1.3 Die Adressierung bei der DFÜ mit Kunden	18
1.3.1 Adressierung X.25.....	18
1.3.2 Verwendung der X.25 Call User Data	18
1.3.3 Transport-Selektor	18
1.3.4 Session-Selektor	18
1.3.5 Presentation-Selektor.....	19
1.3.6 Adressierung der Anwendungsebene (Application Entity Title)	19
1.3.7 Initiator-ID und Passwort.....	19
1.4 Auftragsartenkennungen.....	20
1.4.1 Kategorie 1: Standardisierte Auftragsarten	20
1.4.2 Kategorie 2: Systembedingte Auftragsarten	22
1.4.3 Kategorie 3: Reservierte Auftragsarten für den zwischenbetrieblichen Zahlungsverkehr/Dateiaustausch.....	23
1.4.4 Kategorie 4: Sonstige reservierte Auftragsarten unter Verwendung nicht standardisierter Formate und Verfahren	23
1.5 Fehlermeldungen/Fehlercodes	25
1.6 Betrieb über asynchrone Verbindungen und PAD	26
1.6.1 Anforderungen an das Verfahren.....	26
1.6.2 Kurzbeschreibung der Lösung	26
1.6.3 Spezifikation.....	26
1.6.3.1 Kontroll- und Dateneinheiten.....	26

1.6.3.2	Protokollabläufe	27
1.6.3.3	Mögliche Abläufe beim Austausch von Dateneinheiten	28
1.6.3.4	Konfigurationsparameter	30
1.6.3.5	Datenformate der Kontroll- und Dateieinheiten	31
1.6.3.6	Berechnung der Checksumme und Codierung der NSDU im 7- oder 8-bit Modus	34
1.6.3.7	Decodierung und Prüfung der NSDU	35
1.6.4	Abkürzungsverzeichnis	38
1.7	Kundenprotokoll - inhaltliche und formale Festlegungen	39
1.7.1	Inhaltliche Festlegungen	39
1.7.2	Formale Festlegungen	40
1.7.2.1	Protokollierung der Aktionen auf der Bankseite	40
1.7.2.2	Protokollierung der Fehler bei Unterschriftsprüfung	42
1.7.2.3	Dateianzeige	43
1.7.2.4	Einfügen individueller Texte	43
1.7.2.5	Unterstützung fremdsprachiger Kundenprotokolle	44
1.7.3	Liste der möglichen Meldungen inklusive Textnummer	47
1.7.4	Dateianzeige auf Kunden- und Bankseite	49
2	Standards für die Sicherheit	51
2.1	Festlegungen / Symmetrischer Algorithmus	51
2.2	Kryptographische Verfahren des deutschen Kreditgewerbes für die Elektronische Unterschrift im Rahmen der Kunde-Bank-Kommunikation	53
2.2.1	Allgemeine Anforderungen	53
2.2.2	Elektronische Unterschrift der Version A003	54
2.2.2.1	Festlegungen	54
2.2.2.2	Definitionen	56
2.2.2.3	Voraussetzungen	56
2.2.2.4	Sicherung der Nachrichten	56
2.2.2.5	Formate	58
2.2.2.6	Beschreibung der Abläufe	59
2.2.2.7	Testdaten Elektronische Unterschrift	65
2.2.3	Elektronische Unterschrift der Version A004	69
2.2.3.1	Einleitung	69
2.2.3.2	RSA-Schlüsselkomponenten	69
2.2.3.3	Signatur-Algorithmus	71
2.2.3.4	Signaturverfahren gemäß DIN-Spezifikation	73
2.2.3.5	Referenzen	76
2.2.3.6	Signaturformat A004	77
2.2.3.7	Testdaten Elektronische Unterschrift	80
2.3	Verschlüsselung	84
2.3.1	Allgemeine Anforderungen	84
2.3.2	Schaffung der Voraussetzungen für die verschlüsselte Kommunikation	84
2.3.3	Vorbereitung der Verschlüsselung / Public-Key-Austausch	84

2.3.4 Ver- und Entschlüsselung	87
2.3.4.1 Vorgänge beim Sender	87
2.3.4.2 Vorgänge beim Empfänger	89
2.3.5 Beispielhafte Beschreibung der Abläufe	91
2.3.6 Interne Datenformate der Funktion „Verschlüsselung“	96
2.3.6.1 PDEK und DES-Schlüssel (DEK).....	96
2.3.6.2 EDEK	97
2.3.6.3 Exponent	98
2.3.6.4 Modulo	99
2.3.6.5 Testdaten Verschlüsselung.....	100
2.3.7 Liste der reservierten Antwortcodes.....	106
2.3.8 Abkürzungsverzeichnis	107

1 Standards für die Kommunikation

1.1 Anwendungsprotokoll Kunden-/Bankrechner

Das Protokoll beschreibt die Belegung der Request-Parameter Remote-User-ID, Password und Filename sowie der Anwendungsfehlermeldungen. Nicht genutzte Stellen dieser Felder werden mit Blanks aufgefüllt.

1.1.1 Remote-User-ID

Bankspezifische User-ID (8 Bytes alphanummerisch, beginnend mit einem Alphazeichen)

1.1.2 Remote-Password

Byte 1 – 8: Bankenspezifisches Password (8 Bytes alphanummerisch)

Byte 9 – 16: Bankenspezifisches neues Password (8 Bytes alphanummerisch); nur bei Auftragsarten INI und PWA

1.1.3 Remote-Filename

Das Feld enthält die inhaltliche Beschreibung des Transferauftrages und gegebenenfalls auftragspezifische Daten. Nur Byte 1 bis 44 werden genutzt. Die einzelnen Parameter sind durch Punkte voneinander getrennt. Aus Kompatibilitätsgründen beim Empfänger-Betriebssystem ist das erste Byte eines Qualifiers immer ein Alphazeichen. Die Grundstellung für nicht belegte Stellen ist 'N'. Der Aufbau ist:

PV.Kunden-ID.Auftragsart.Auftragsattribut[.Auftragsnummer] [.Auftragsparameter]

Erläuterung:

- PV (2 Bytes alphanummerisch):
Versionsnummer des Anwendungsprotokolls; zur Zeit konstant '**A3**'
- Kunden-ID (8 Bytes alphanummerisch):
Bankspezifische Kunden-ID
- Auftragsart-Kennung (3 Bytes alphanummerisch, siehe **1.4 Auftragsartenkennungen**)
- Auftragsattribut (5 Bytes alphanummerisch):

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Byte	Inhalt	Auswahlmöglichkeiten
1	Dateiart	O = Originaldatei mit zugehöriger Unterschriftsdatei U = Unterschriftsdatei D = Originaldatei ohne zugehöriger Unterschriftsdatei B = Originaldatei und EU-Datei in einer physikalischen Datei P = Protokolldatei Y = Dummy-Datei
2	Komprimierungsart ¹	N = Keine Komprimierung F = FLAM-Komprimierung Z = PKZIP-Komprimierung X = x-Press-Komprimierung ... Die zum Einsatz kommenden Komprimierungsprodukte werden durch die ZKA-Buchstabenkennungen festgelegt.
3	Verschlüsselungsart	N = Keine Verschlüsselung H = Hybrid-Verfahren DES/RSA R = RSA
4	Reserve	
5	Reserve	

- **Auftragsnummer (4 Bytes alphanummerisch):**
Je Kunde eindeutige Auftragsnummer für die Übertragungsrichtung Kunde/Bank. Dient als Ordnungsbegriff bei der PC-Host-Kommunikation, insbesondere beim Protokollabruf und bei der Synchronisation von Original- und Unterschriftsdatei.

Byte	Inhalt
1	A
2	laufende Nummer des Kunden-DFÜ-PCs, alphanummerisch
3-4	2 Stellen alphanummerisch (werden aufsteigend vergeben)

- **Auftragsparameter:**
Auftragspezifische weitere Parameter, falls für die Auftragsart erforderlich bzw. zugelassen. Zur Zeit:

Auftragsart	Auftragsparameter
STA und/oder PTK	optional Datum-von Datum-bis (VJJMMTT.BJJMMTT)

¹ Sofern nichts anderes vereinbart, ist das FLAM-Verfahren das Standard-Komprimierungsverfahren.

Auftragsart	Auftragsparameter
VPK	Kennung für User-ID des Kunden, der VPK-Auftrag unterschrieben hat Byte 1: U (Kennung für User-ID) Byte 2 – 9: User-ID
alle verschlüsselt zu übertragenden Dateien	Verschlüsselungs-Public-Key-Hashwert (33 Bytes alphanummerisch): Byte 1: H (Kennung für Hashwert) Byte 2-33: Verschlüsselungs-Public-Key in Hex-Darstellung

1.2 FTAM-Verfahrensspezifikation

Als Grundlage für die DFÜ-Kommunikation mit Kunden dienen folgende nationale und internationale Standards:

- Profile A/111 [38]
- FTAM [18]
- ASN.1 [14], [15]
- ACSE [16], [17]
- Presentation Service [12] und Protocol [13]
- Session Service [10] und Protocol [11]
- Profile T/331 [37]
- Transport Service [8] und Protocol [9]
- Datex-P [2]
- OSI-Adressierung [4]

Die internationalen Standards der ISO sind in vielen Teilen allgemein gehalten, um möglichst viele verschiedene Anwendungsszenarien abzudecken und um zukünftige Weiterentwicklungen nicht zu behindern.

Daher werden zusätzlich bestimmte Richtlinien zur Benutzung dieser Standards für bestimmte Anwendergruppen definiert. Diese Richtlinien, Functional Standards oder Profile genannt, werden von verschiedenen Normungsgremien erarbeitet (z. B. SPAG, CEN/CENELEC, ISO).

Das Profile A/111 des Guide to Use of Standards (GUS Rev. 3.1), herausgegeben von der Standards Promotion and Applications Group (SPAG), definiert einen Subset der Funktionen in den Ebenen 5 bis 7 einer FTAM-Anwendung. Dieses Profil wurde von der europäischen Normungsorganisation CEN/CENELEC in Abstimmung mit CEPT als europäische Vornorm ENV 41 204 [38] verabschiedet. Dieser Subset von OSI-Session, OSI-Presentation, OSI-

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

ACSE und FTAM ist die Grundlage für den Filetransfer bei der DFÜ-Kommunikation mit Kunden.

Das ebenfalls von SPAG herausgegebene Profil T/31 X befasst sich mit der Netzwerkverbindung und dem darauf aufsetzenden Transportprotokoll. Dieses Profil wurde ebenfalls als Vornorm ENV 41 104 [37] von CEN/CENELEC in Abstimmung mit CEPT übernommen und dient als weitere Grundlage für die DFÜ-Kommunikation mit Kunden.

Abbildung 1.1 zeigt den Zusammenhang der verschiedenen Protokollschichten in einer Übersicht.

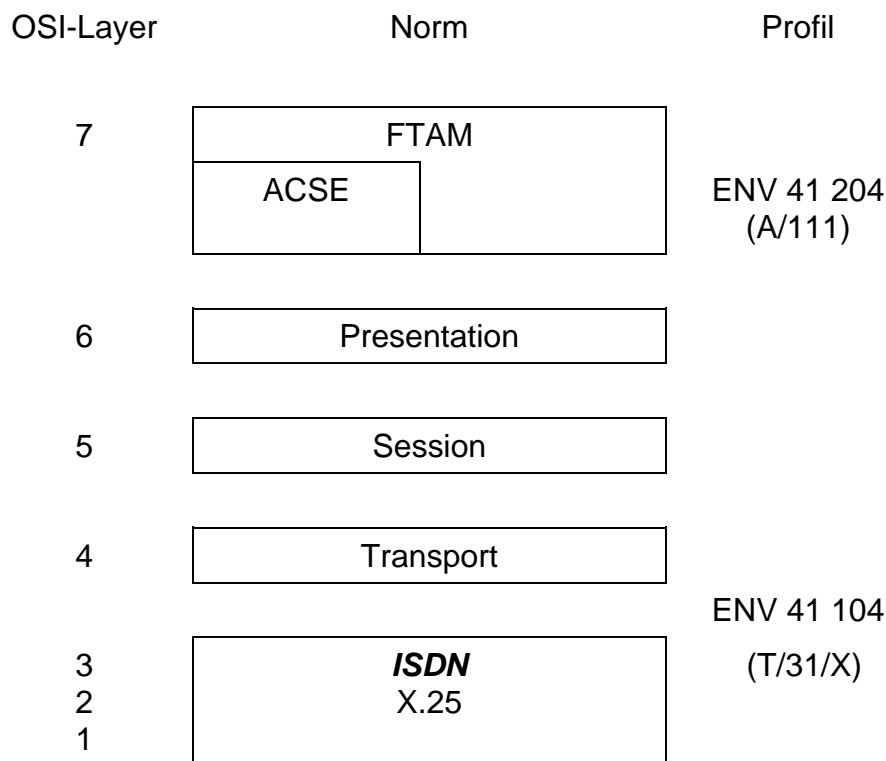


Abbildung 1.1 ISO OSI Schichtmodell

1.2.1 Virtual Filestore

Die Art der Dateihaltung und der Zugriff auf die Dateien ist bei den verschiedenen realen DV-Systemen sehr unterschiedlich. Um unterschiedliche reale Systeme miteinander zu verbinden, wird daher ein gemeinsames Modell zur Beschreibung von Dateien und ihrer Attribute benötigt. Dieses Modell wird Virtual Filestore genannt.

Rollenverteilung: Eine FTAM-Verbindung wird immer von einem FTAM-Initiator initiiert. Im Rahmen der DFÜ mit Kunden ist dies immer der Kunde. Der dazugehörige Partner im Remote-System ist ein FTAM-Responder. Der FTAM-Initiator kann sowohl Sender von Daten wie auch Empfänger sein. Gleiches gilt für den FTAM-Responder. Der FTAM-Initiator kann durch die Dienstleistung des FTAM-Responders auf den Virtual Filestore des Partnersystems zugreifen (siehe [Abbildung 1.2](#)).

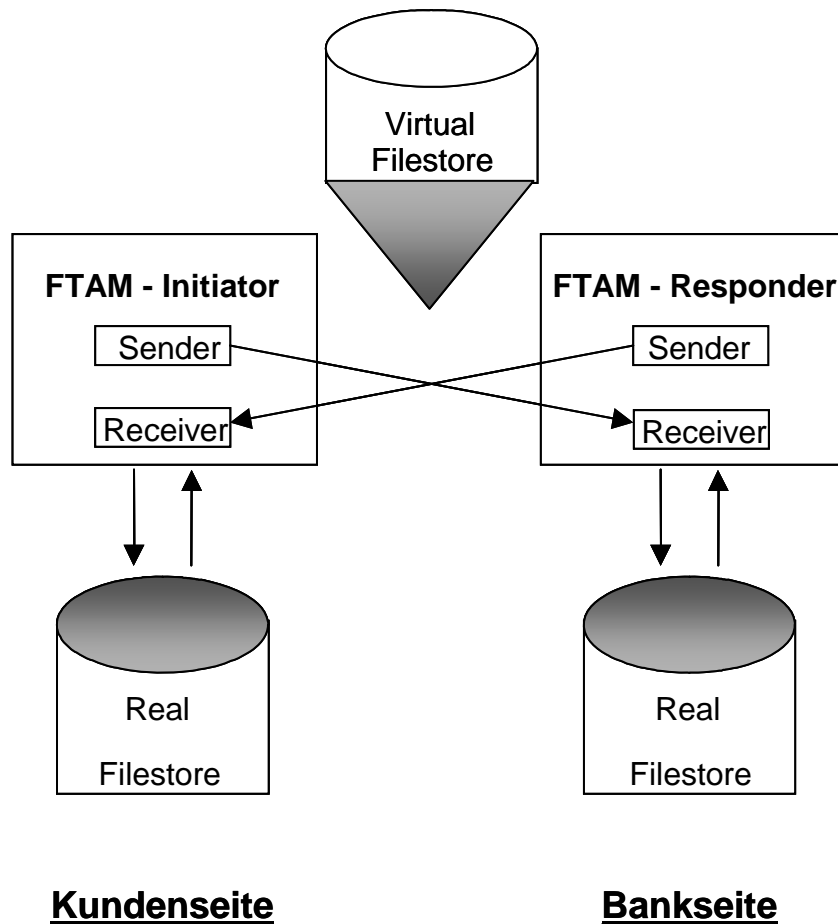


Abbildung 1.2 FTAM-Initiator/Responder

Jeder Partner in einem FTAM-Verbund bildet seine remote zugänglichen Daten bzw. Datenbereiche auf einem Virtual Filestore ab. Ein Virtual Filestore kann eine oder mehrere Dateien, ganze Directories (auch hierarchisch) oder sogar die gesamten Dateien auf dem Peripheriespeicher eines Systems beinhalten.

Dateistruktur:

Bei FTAM wird zur Zeit nur das File-Model hierarchical² verwendet. Bei diesem Modell besteht ein File aus verschiedenen Zugriffseinheiten, die File Access Data Units (FADU) genannt werden. Bei der DFÜ mit Kunden besteht eine Datei nur aus einer FADU, die nur die

² Object Identifier: {iso standard 8571 file-model (3) hierarchical (1)}.

Data Unit (DU) des Root-Knotens beinhaltet. Diese Data Unit wird in einzelnen Data Elements (DEs), stellenweise auch Strings genannt, übertragen.

Filenamen:

Nach ISO 8571 [20] besteht der Filename aus einem Vektor von GraphicStrings. Es werden Filenamen bis zu einer maximalen Länge von 44 Charakteren verwendet.

Der Filename muss nach ISO codiert [28] sein; EBCDI-Code ist nicht erlaubt.

FTAM Dateistruktur (Constraint Set):

Das Profile A/111 [38] unterstützt nur die Struktur unstructured³. Das bedeutet, dass nur auf die Datei als Ganzes zugegriffen werden kann und nicht auf Teile der Datei. Diese Zugriffsart entspricht der Anwendung bei der DFÜ mit Kunden.

Dateiinhalte (FTAM Document Type):

Das Profile A/111 [38] erlaubt nur die Document Types ISO FTAM unstructured text (FTAM-1) und ISO FTAM unstructured binary (FTAM-3).

FTAM-1 lässt die Übertragung beliebiger Textdateien zu. Für die Übertragung von Textdateien mit fester Satzlänge wird die Satzlänge der jeweiligen Datei an der Aufrufchnittstelle als Parameter übergeben.

FTAM-3 lässt die Übertragung von beliebigen Binärdateien zu. Diese können bei der DFÜ mit Kunden dann auch komprimiert (Komprimierung in der Anwendung, nicht in FTAM) sein.

Nach dem Profile A/111 ist für FTAM-3 die *string significance* nur mit dem Wert not significant zugelassen, wodurch keine Abbildung der Data Elements auf Sätze mehr möglich ist.

Die Satzstruktur wird daher aus den mitübertragenen Satzlängefeldern vor jedem Datensatz grundsätzlich nicht in FTAM rekonstruiert.

Die Rekonstruktion erfolgt vielmehr empfangersystemspezifisch auf Basis der Auftragsarten-Kennung des Kunden-/Bankrechner-Protokolls (siehe Kapitel Kontroll- und Dateneinheit). Dabei sind bei Binärdateien variabler Satzlänge folgende Alternativen möglich:

Alternative 1 (Userexits): Unmittelbar nach dem Empfang der Datei, noch während der FTAM-Verbindung, am Ende des FTAM-Select-Regimes, wird die Nachbehandlung (Rekonstruktion) der Dateien unter Nutzung eines entsprechenden Userexits durchgeführt. Voraussetzung ist hierfür, dass das jeweilig eingesetzte FTAM-Produkt die gewünschten Userexits anbietet.

Alternative 2 (Anwendung): Die Nachbehandlung (Rekonstruktion) erfolgt erst nach Beendigung der FTAM-Verbindung am Ende des FTAM-Regimes durch empfangersystemspezifische Anwendungen.

FTAM Zugriffsumgebung (Access Context):

Durch das Profile A/111 wird nur UA – Unstructured All Data Units Access Context – unterstützt. Das bedeutet, dass nur der Inhalt der Datei (keine Strukturinformation) – und dieser nur im ganzen (nicht in Teilen) – zugreifbar ist. Dieser Access Context korrespondiert mit der FTAM Dateistruktur (Constraint Sets) und ist für die Anwendungen bei der DFÜ mit Kunden ausreichend.

³ Object Identifier: {iso standard 8571 constraint-set (4) unstructured (1)}.

FTAM Dateiattribute (File Attributes):

Von der Kernel Group werden folgende Attribute benötigt:

- filename (siehe "Filename" in Kapitel 1.2.1 Virtual Filestore)
- permitted actions: replace, extended, read
- contents type (beschränkt auf Document Type Format)

Von der Storage Group, der Security Group und der Private Group werden keine Attribute benötigt.

1.2.2 FTAM Units

FTAM File Service:

Der External File Service ist in die Functional Units U1 bis U8 unterteilt, welche wiederum den fünf Dienstklassen (T, A, M, TM, U) zugeordnet sind. Der Internal File Service ist in die Functional Units U9 und U10 unterteilt.

Bei der DFÜ mit Kunden wird die Dienstklasse (Service Class) File Transfer Class (T) mit folgenden Functional Units benötigt:

Functional Unit	
U1	Kernel
U2	Read
U3	Write (nur write replace genügt)
U5	Limited File Management (optional in A/111)
U7	Grouping
U9	Recovery (Class 3) (optional in A/111)

Für die DFÜ mit Kunden wird U5 benötigt, da Files remote vom Sender angelegt werden. U9 wird im Zusammenhang mit PAD-Verbindungen benötigt.

Folgende dateibezogene Aktionen werden bei DFÜ mit Kunden verwendet:

- Select File
- Create File (in A/111 nur optional)
- Open File
- Close File
- Deselect File

Von den unterschiedlichen Zugriffsarten werden folgende benötigt:

- Read

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- Extend
- Write replace

FTAM Benutzung:

Mit dem ordnungsgemäßen Verbindungsabbau der FTAM-Verbindung (F-TERMINATE positiv quittiert) geht die Verantwortung auf den Empfänger der Datei über.

FTAM Protocol:

Folgende FTAM Protocol Data Units werden beim Kunden verwendet:

- F-Initialize FPDU
- F-Terminate FPDU
- F-U-Abort FPDU
- F-Select FPDU
- F-P-Abort FPDU
- F-Create FPDU
- F-Deselect FPDU
- F-Open FPDU
- F-Close FPDU
- F-Begin-Group FPDU
- F-End-Group FPDU
- F-Read FPDU
- F-Write FPDU
- F-Data-End FPDU
- F-Transfer-End FPDU
- F-Recover FPDU
- F-Cancel FPDU

In den folgenden Kapiteln werden sie mit den verwendeten Parametern aufgelistet, wobei das Feld Status folgende Bedeutung hat:

Status	Bedeutung
m	Pflicht (mandatory); weitere Detaillierung unter m1, m2 bzw. m3
m1	Pflicht (mandatory) im Standard (FTAM, ACSE, Presentation, Session)
m2	Pflicht im Profile A/111 (ENV 41 204) [38]
m3	Pflicht bei der DFÜ mit Kunden

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Status	Bedeutung
d	Für dieses Feld ist ein Defaultwert definiert. Falls bei Wert keine Einschränkung genannt ist, muss dieses Feld voll unterstützt werden.
s	Unterstützt (supported)
°	Optional (optional). Die Syntax muss unterstützt werden, die Semantik ist optional.
–	Nicht verwendet
x/y	x = Status für Initiator, y = Status für Responder

Protocol Data Unit	Status	Wert
Parameter		
F-Initialize	m1	
protocol-Version	d	version-1
presentation-context-management	d	false
service-class	d	transfer-class
functional-units	m1	
attribute-groups	d	
ftam-quality-of-service	m1	
contents-type-list	m2	
initiator-Identity	m2	GraphicString (maximal 8 Zeichen)
account	°	
filestore-password	m2/°	GraphicString (maximal 8 Zeichen)
checkpoint-window	d	bei Passwortänderung 9 bis 16 Zeichen
action-result	d	
diagnostic	m2	
diagnostic-type	m1	
error-identifier	m1	
error-observer	m1	
error-source	m1	
further-details	m2	
F-Begin-Group	m1	
Threshold	m1	
F-End-Group	m1	
(no parameters)		
F-Create	m3	
override	d	
initial-attributes	m1	filename only
create-password	m2/°	
request-access	m1	
access-passwords	m2/°	
concurrency-control	°	
account	°	

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Protocol Data Unit	Status	Wert
Parameter		
state-result	d	
action-result	d	
Diagnostic	m2	
F-Open	m1	
processing-mode	d	replace
contents-type	m1	FTAM-1 und FTAM-3
concurrency-control	°	
enable-fadu-locking	d	false
activity-identifizier	°	
recovery-mode	d	
state-result	d	
action-result	d	
diagnostic	m2	
state	°	
F-Write	m2	
file-access-data-unit-operation	m1	replace
file-access-data-unit-identity	m1	first
F-Data-End	m1	
action-result	d	
diagnostic	m1	
F-Transfer-End	m1	
action-result	m1	
diagnostic	m2	
F-Cancel	m1	
action-result	d	
diagnostic	m2	
F-Close	m1	
action-result	d	
diagnostic	m2	
F-Deselect	m1	
action-result	d	
charging	°	
diagnostic	m2	
F-Terminate	m1	
charging	°	
F-U-Abort	m1	
action-result	d	
diagnostic	m2	
F-P-Abort	m1	
action-result	d	
diagnostic	m2	

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Zusätzlich

Protocol Data Units	Status	Wert
Parameter		
F-Select	m1	
attributes	m1	
requested-access	m1	
access-passwords	o	
concurrency-control	o	
account	o	
state-result	d	
action-result	d	
diagnostic	m2	
F-Recover	m3	bei PAD-Verbindungen
activity-identifier	m1	
bulk-transfer-numer	m1	
requested-access	m1	
access-passwords	o	
recovery-point	d	
contents-type	m1	
state-result	d	
action-result	d	
dagnostic	s	
presentation-action	d	
F-Read		
file-access-data-unit-identify	m1	
access-context	m1	

1.2.3 Anwendungsschicht ACSE

Folgende Association Control Service Elements werden verwendet:

- A-Associate-Request (AARQ) APDU⁴
- A-Associate-Response (AARE) APDU
- A-Associate-Release-Request (RLRQ) APDU
- A-Associate-Release-Response (RLRE) APDU
- A-Abort (ABRT) APDU

⁴ APDU = Application Package Data Unit

Im folgenden werden sie mit den verwendeten Parametern aufgelistet.

Protocol Data Units	Status ⁵	Wert
Parameter		
A-Associate-Request	m	
protocol-Version	d	version-1 (0)
application context name	m1	iso standard 8571 application context (1) iso-ftam (1)
calling AP title	m2	1 3 9999 1 ftam-nil-ap-title (7)
calling AE Qualifier	m2	Integer
called AP title	m2	1 3 9999 1 ftam-nil-ap-title (7)
user information	m2	F-Initialize-Request FPDU
A-Associate-Response	m	
protocol version	d	version-1 (0)
result	m2	
responding AP title	m2	1 3 9999 1 ftam-nil-ap-title (7)
responding AE Qualifier	m2	Integer
application context name	m1	iso standard 8571 application context (1) iso-ftam (1)
user information	m2	F-Initialize-Response FPDU
A-Associate-Release Request	m	
user information	m2	F-Terminate-Request FPDU
A-Associate-Release- Response	m	
user information	m2	F-Terminate-Response FPDU
A-Abort	m	
abort source	m1	
user information	m2	F-P-Abort

1.2.4 Presentation Layer

Es kommt ein Presentation Layer nach dem in ISO 8822 [12] definierten Service und gemäß dem in ISO 8823 [13] definierten Protokoll zum Einsatz.

- Funktionsumfang:
Es werden nur die Kernel-Funktionen benötigt.
- Kontexte:
Es werden folgende, in Abstract Syntax Notation 1 (ASN.1) [14] definierte Datenstrukturen bearbeitet: ISO 8650-ACSE⁶, ISO 8571-FTAM PCI⁷, FTAM unstructured binary abstract syntax⁸

⁵ Erläuterungen siehe in Kapitel 1.2.2 „FTAM Units“

⁶ Object Identifier: {joint-iso-ccitt association-control (2) abstract-syntax (1) apdus (0) version 1 (1)}.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- Kodierungsregeln:
Die Datenstrukturen werden nach den Basic Encoding Rules for ASN.1 [15] codiert⁹.
- Wertebereiche:
Für das Encoding gelten folgende Einschränkungen:

Taglänge:	maximal 2 Oktetts
Längenfeld:	maximal 5 Oktetts
Inhalt:	maximal 8k Oktetts

1.2.5 Session Layer

- Namenskonventionen des Session-Selektors:
Der Session-Selektor kann maximal aus 16 Oktetts bestehen. Sein Wert ist im Kapitel 1.3: „Die Adressierung bei der DFÜ mit Kunden“ festgelegt.
- Genutzter Funktionsumfang:

Functional Unit	Support Level ⁵	Bemerkung
kernel	m1	
half-duplex	–	
duplex	m2	wird verwendet
negotiated release	–	
typed data	–	
capability data exchange	–	
minor synchronize	m3	bei FTAM Recovery (PAD)
major synchronize	–	
resynchronize	◦	bei Restart
exceptions	–	
activity management	–	

- Verwendete Tokens

⁷ Object Identifier: {iso standard 8571 abstract-syntax (2) ftam-pci (1)}.

⁸ Object Identifier: {iso standard abstract-syntax (2) unstructured-binary (4)}.

⁹ Object Identifier: {joint-iso-ccitt (2) asnl (1) basic-encoding (1)}.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Token	Support Level ⁵	Bemerkung
synchronize minor	o	bei FTAM Recovery oder Restart

- Protokoll Optionen

Protocol Options	Support Level ⁵	Bemerkung
basic concatenation	m1	
segmenting	o	

- Wertebereiche:
Die maximale Länge der SPDUs ist entsprechend 8 k Oktetts der TSDU definiert.
- User Data:
Die User Data enthalten die Connect-Request-PPDU der Presentation Layer.

1.2.6 Transport Layer

- Namenskonvention des Transport-Selektors:
Der Transport-Selektor darf maximal 32 Oktetts lang sein.
Sein Wert ist im Kapitel 1.3 „Die Adressierung bei der DFÜ mit Kunden“ festgelegt.
- Verwendete Protokollklasse(n):
Auf den beteiligten Systemen muss das Transportprotokoll Class 0 oder die beiden Klassen Class 0 und Class 2 implementiert sein.
- Verwendung der User Data:
User Data werden nicht verwendet.
- Recordlänge und Nachrichtenlänge:
Als Recordlängen sind 128, 256, 512, 1024, 2048, 4096 und 8192 Oktetts zugelassen.
Die maximale Nachrichtenlänge muss mindestens 8 k Oktetts betragen.

1.2.7 Network Layer

An der Kommunikationsstelle des Kunden ist ein Datex-P10-Hauptanschluss, alternativ ein PAD-Anschluss erforderlich.

- Datex-P10-Hauptanschluss:
Zwischen der Kommunikationsstelle der Kunden und der Bank werden gewählte virtuelle Verbindungen (SVC) aufgebaut. Die verfügbare Datenrate und die Anzahl der möglichen Verbindungen sind so auszulegen, dass einem Verbindungsaufbauwunsch der Bank in der Regel entsprochen wird.
Bei Verbindungsaufbau auszuhandelnde Leistungsmerkmale
Anrufe der Kunden an die Bank werden ohne das Dienstmerkmal „Gebührenübernahme“ durchgeführt.
Eine Teilnehmerbetriebsklasse (Closed User Group) wird zur Zeit nicht vorgesehen.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Verwendung der Call User Data (CUD)

Das X.25-Feld für CUD wird einheitlich mit 01 00 00 00 besetzt.

- PAD-Anschluss:
Für die Kommunikation über PAD-Anschluss wird ein gesondertes Sicherungsprotokoll verwendet. Diese Spezifikation ist den Standards für die Kommunikation (Kapitel 1.6 „Betrieb über asynchrone Verbindungen und PAD“) beigefügt. Das Verfahren ist als Teil der Schicht 3 oberhalb der X.25-Schnittstelle angesiedelt.

1.2.8 Literaturverzeichnis

- [1] DATEL-Handbuch, 2. Auflage April 1985. Herausgegeben vom Fernmeldetechnischen Zentralamt 6100 Darmstadt.
- [2] DATEX-Handbuch, 2. Auflage August 1988. Herausgegeben vom Fernmeldetechnischen Zentralamt 6100 Darmstadt.
- [3] ISO 7498, Information processing systems – Open Systems Interconnection – Basic reference model.
- [4] ISO/DIS 7498-3, Information processing systems – Open Systems Interconnection – Part 3: Naming and Processing.
- [5] ISO 8509, Information processing systems – Open Systems Interconnection – Service conventions.
- [6] ISO 8208, Information processing systems – Data Communications – X.25 packet level protocol for DTE. Version 1985_03_27.
- [7] ISO 8348, Information processing systems – Data Communications – Network Service definition. Version 1985_03_27.
- [8] ISO 8072, Information processing systems – Open Systems Interconnection – Transport Service Specification. Version 1986.
- [9] ISO 8073, Information processing systems – Open Systems Interconnection – Transport Protocol Specification. Version 1986.
- [10] ISO 8326, Information processing systems – Open Systems Interconnection – Basic connection session service definition.
- [11] ISO 8327, Information processing system – Open Systems Interconnection – Basic connection oriented session protocol specification.
- [12] ISO 8822, Information processing system – Open Systems Interconnection – Connection oriented presentation service definition.
- [13] ISO 8823, Information processing systems – Open Systems Interconnection – Connection oriented presentation protocol specification.
- [14] ISO 8824, Information processing systems – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1).
- [15] ISO 8825, Information processing systems – Open Systems Interconnection – Basic encoding rules for Abstract Syntax Notation One (ASN.1).
- [16] ISO 8649, Information processing systems – Open Systems Interconnection – Service definition for the Association Control Service Element.
- [17] ISO 8650, Information processing systems – Open Systems Interconnection – Protocol specification for the Association Control Service Element.
- [18] ISO 8571, Information processing systems, Open Systems Interconnection – File Transfer, Access and Management –.
- [19] ISO 8571 Part 1, Information processing systems, Open Systems Interconnection – File Transfer, Access and Management – Part 1: General introduction.
- [20] ISO 8571 Part 2, Information processing systems, Open Systems Interconnection – File Transfer, Access and Management – Part 2: Virtual filestore definition.
- [21] ISO 8571 Part 3, Information processing systems, Open Systems Interconnection – File Transfer, Access and Management – Part 3: File service definition.
- [22] ISO 8571 Part 4, Information processing systems, Open Systems Interconnection – File Transfer, Access and Management – Part 4: File protocol specification.
- [23] ISO 646, Information processing – ISO 7_bit coded character set for information interchange
- [24] ISO 2022, Information processing – ISO 7_bit and 8_bit coded character sets – Code extension techniques.

- [25] ISO 4873, Information processing – ISO 8_bit code for information interchange – Structure and rules for implementation.
- [26] ISO 6429, Information processing – ISO 7_bit and 8_bit coded character sets – Additional control functions for character-imaging devices.
- [27] ISO 6937 Part 2, Information processing – Coded character sets for text communication – Part 2: Latin alphabetic and nonalphabetic graphic characters.
- [28] ISO 8859_1, Information processing – 8_bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1.
- [29] ISO 2375, Data processing – Procedure for registration of escape sequences.
- [30] ISO DIS 8227 edition 1985, Information processing – Data encipherment – Specification of algorithm DEA 1.
- [31] ISO 8372, Information processing – Modes of operation for a 64_bit block cipher algorithm.
- [32] ISO 8730, Banking – Requirements for message authentication (wholesale).
- [33] ISO 8732, Banking – Key management (wholesale).
- [34] CCITT Recommendation X.224 –.
- [35] CCITT Recommendation X.410 – Message Handling Systems – Remote Operations and Reliable Transfer Server (1984).
- [36] CEN/CENELEC/CEPT, M_IT_02 – Directory of functional standards – (For interworking in an OSI Environment).
- [37] CEN/CENELEC/CEPT, ENV 41 104 – Information Systems Interconnection – Packet Switched Data Networks – Permanent access (T/331).
- [38] CEN/CENELEC/CEPT, ENV 41 204 – Simple file transfer (A/111), 23.02.88.
- [39] National Bureau of Standards (NBS) – Implementors Agreements – Draft Stable Document, Vers Okt. 87.
- [40] Standard Promotion and Application Group (S.P.A.G.) – Guide to the Use of Standards.
- [41] UK Government OSI Profile V3.0.
- [42] S.W.I.F.T Benutzerhandbuch
- [43] Ergebnisniederschrift der Sitzung des ZKA-Arbeitskreises „Belegloser Auslandszahlungsverkehr“ vom 20.10.1988.
- [44] Datenverschlüsselung – Festlegung des Algorithmus DEA 1 – ISO/DIS 8227 Ausgabe 1985.

1.3 Die Adressierung bei der DFÜ mit Kunden

Die Adressierung der Protokoll-Instanzen erfolgt nach den Prinzipien des Referenz-Modells, d. h. eine Instanz der Schicht (N) wird über die Adresse des service access point (SAP) der Schicht (N – 1) erreicht. Für die Anwendungen der DFÜ-Kommunikation mit Kunden wird eine feste Adressen-Struktur verwendet.

Ein Virtual Filestore wird über die Adresse des Presentation-Service-Access-Point (PSAP) gegebenenfalls in Verbindung mit der Initiator-ID selektiert.

Die PSAP-Adresse besteht aus der NSAP-Adresse, den Werten des T-Selektors, des S-Selektors und des P-Selektors. Die NSAP-Adresse wird hier durch die Datex-P-Rufnummer repräsentiert.

Im Rahmen der DFÜ mit Kunden sind die NSAP-Adresse (Datex-P-Rufnummer), der T-, S- und P-Selektor sowie auch der Application Entity Title von der Bank fest anzulegen. Die Auswahl eines kundenspezifischen Virtual Filestore geschieht hierbei über die Initiator-ID, die durch die Bank je Kommunikationstelle eindeutig vorgegeben wird.

Die folgenden Detailfestlegungen sind als Empfehlung zu verstehen. Jede Bank kann nach eigenen Kriterien die eigene Adresse und die Adressen der Kunden neu definieren. Es ist ratsam, die Selektoren der Schichten mit unterschiedlichen Namen zu belegen.

1.3.1 Adressierung X.25

Da die volle OSI-Adressierung erst mit der X.25-Version von 1984 eine weite Verbreitung gefunden hat, ist die NSAP-Adresse vorerst nur durch die X.25 Adresse (DTE-Nummer) zu bilden.

1.3.2 Verwendung der X.25 Call User Data

Zur Adressierung des OSI-Transportprotokolls ist der hexadezimale Wert 01 00 00 00 mit der Satzlänge 0 vorgesehen.

1.3.3 Transport-Selektor

Der Transport-Selektor besteht aus bis zu 32 Oktetts.

- Wert des T-Selectors der Bank: EB-TSAP
(X'45 42 2D 54 53 41 50')
- Wert des T-Selectors des Kunden: EB-CLIENT
(X'45 42 43 4C 49 45 4E 54')

1.3.4 Session-Selektor

Der Session-Selektor besteht aus bis zu 6 Oktetts.

- Wert des S-Selectors der Bank: *darf nicht belegt werden*
- Wert des S-Selectors des Kunden: *darf nicht belegt werden*

1.3.5 Presentation-Selektor

Der Presentation-Selektor besteht aus maximal 4 Oktetts (39).

- Wert des P-Selectors der Bank: *darf nicht belegt werden*
- Wert des P-Selectors des Kunden: *darf nicht belegt werden*

1.3.6 Adressierung der Anwendungsebene (Application Entity Title)

Der Application Entity Title (AET) wird zur Selektion eines Virtual Filestore verwendet. Der AET besteht aus dem Application Process Title mit dem konstanten Wert „Iso (1) Identified-organisation (3) 9999 1 ftam-nil-ap-title (7)“ und einem Application Entity Qualifier (AEQ), der als Integer definiert ist. Damit ergibt sich konkret folgende Adressierung:

- AET: 1 3 9999 1 7
- AEQ zur Adressierung der Bank: Integerwert '0'
- AEQ zur Adressierung des Kunden: Integerwert wahlfrei zwischen -2^{31} und $2^{31}-1$

1.3.7 Initiator-ID und Passwort

Jeder Kommunikationsstelle wird eine Initiator-ID zugeordnet, die aus einer eindeutigen achtstelligen Kenn-Nummer der Kommunikationsstelle gebildet wird und der Kunden-ID aus der Bankparameterdatei entspricht.

Das Passwort wird in der verhashten Form verwendet.

Die Initiator-ID und das Passwort werden bei jedem Remote-Filezugriff angegeben. Sie dienen dem Remote-System zur Identifikation des Anrufers.

1.4 Auftragsartenkennungen

1.4.1 Kategorie 1: Standardisierte Auftragsarten

Kennung	Text	Satzlänge ¹⁰	Bits	Format
AAE	Senden Importakkreditiv Änderung	-1	7	DTALC / Änderung
AEA	Senden Exportakkreditive	-1	7	MT700, MT707, MT710 MT720, MT799
AIA	Senden Importakkreditive Avisierung	-1	7	DTALC / Avisierung
AID ¹¹	Senden Importakkreditive Dokumentenaufnahme	-1	7	DTALCA
AKA	Abholen Importakkreditive	-1	7	DTALCR
AKD ¹¹	Abholen Importakkreditive Abrechnung	-1	7	DTALCD
AWV	AWV-Meldung senden	-1	7	EDIFACT
AZM	AZV im Magnetbandformat senden (Satzlänge variabel)	-1	8	DTAZV-Magnetband
AZV	AZV im Diskettenformat senden	256	7	DTAZV-Diskettenformat
AZ2	AZV im Magnetbandformat senden (Satzlängenfeld 2 Bytes)	64	8	DTAZV-Magnetband
AZ4	AZV im Magnetbandformat senden (Satzlängenfeld 4 Bytes)	64	8	DTAZV-Magnetband
DDG	Abholen Devisenhandelsbestätigung	-1	7	MT300
DHB	Senden Devisenhandelsbestätigung	-1	7	MT300
DTE	Eilauftrag (IZV im DTAUS0-Format) senden	128	7	DTAUS0
DTI	IZV-Datei abholen	128	7	DTAUS0
DTM	MCV-Datei abholen (Format analog MCV)	64	8	DTAUS-Magnetband
DTT ¹²	Telegrafische Zahlungen senden	128	7	DTAUS0
DTV	Zahlungsverkehrsdateien von Service-Rechenzentren senden	128	7	DTAUS0
DT2	MC2-Datei abholen (Format analog MC2)	64	8	DTAUS-Magnetband
DT4	MC4-Datei abholen (Format analog MC4)	-1	8	DTAUS-Magnetband

¹⁰ Die Satzlänge „-1“ bedeutet „variable Satzlänge“.

¹¹ Inkrafttreten dieser Änderung am 01.09.2005.

¹² Diese Auftragsart wird nur noch bis zum 31.12.2004 unterstützt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Kennung	Text	Satzlänge ¹⁰	Bits	Format
EAB	Exportakkreditive abholen	-1	7	DTAEA MT700, MT707, MT710 MT720, MT799
EAD ¹¹	Abholen Exportakkreditive Abrechnung	-1	7	DTAEAD
ECS	Senden electronic-cash Lastschriftdatei	128	7	DTAUS0
EDC	Senden Maestro-Lastschriftdatei	128	7	DTAUS0
EEA	EDIFACT abholen ASCII	-1	7	EDIFACT
EEZ	EDIFACT abholen EBCDIC	-1	8	EDIFACT
EIB	Ausführungsanzeige (Exportinkasso) Bank an Kunde abholen	-1	7	EDIFACT
EIK	Senden Exportinkassi	-1	7	EDIFACT
ESA	EDIFACT senden ASCII	-1	7	EDIFACT
ESM ¹³	EU-Standardüberweisung (Zahlungsart 13) im Magnetbandformat (Satzlängen- feld 4 Bytes)	64	8	DTAZV-Magnetband
ESR	Einreichung von EDIFACT-Lastschriften	-1	7	EDIFACT
ESZ	EDIFACT senden EBCDIC	-1	8	EDIFACT
ESU ¹⁴	EU-Standardüberweisung (Zahlungsart 13)	256	7	DTAZV
EUE	Taggleiche grenzüberschreitende Euro-Eilzahlung	256	7	DTAZV
GAB	Abholen Garantien	-1	7	MT760, MT767
GAK	Senden Garantien	-1	7	MT760
GKT	GeldKarte-Umsatz senden (Datenauf- bau gemäß GeldKarte-Spezifikation)	128	7	DTAUS
IDD	Internationale Lastschriften	-1	7	MT104
IIB	Abholen Importinkassi	-1	7	MT410 – 419
IIK	Senden Importinkassi	-1	7	MT410 – 419
INT	Internationaler Zahlungsverkehr	-1	7	MT101, MT104
IZG	Inlandszahlungsverkehrsauftrag senden (nur Gutschriften)	128	7	DTAUS0
IZL	Inlandszahlungsverkehrsauftrag senden (nur Lastschriften)	128	7	DTAUS0
IZV	Inlandszahlungsverkehrsauftrag sen- den	128	7	DTAUS0
MAO	Abholen Magnetband-Datei aus optischer Beleglesung	-1	7	MAOBE
MCV	Senden IZV-Magnetbandformat (Satzlängenfeld 4 Bytes)	64	8	DTAUS-Magnetband

¹³ Soweit die separate Einlieferung von EU-Standardüberweisungen im Magnetbandformat vereinbart wurde, ist diese Auftragsart zu verwenden.

¹⁴ Soweit die separate Einlieferung von EU-Standardüberweisungen vereinbart wurde, ist diese Auftragsart zu verwenden (außer Magnetbandformat, dann ist ESM zu verwenden).

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Kennung	Text	Satzlänge ¹⁰	Bits	Format
MC2	Senden IZV-Magnetbandformat (Satzlängenfeld 2 Bytes)	64	8	DTAUS-Magnetband
MC4	Senden IZV-Magnetbandformat (Satzlänge variabel)	-1	8	DTAUS-Magnetband
POZ	Senden POZ-Datei	128	7	DTAUS0
RDT	Rücklastschrift an Kunde	128	7	DTAUS0
RFT	Request for Transfer	-1	7	MT101
STA	Abholen Swift-Tagesauszüge	-1	7	MT940
VMK	Abholen kurzfristige Vormerkposten	-1	7	MT942
VML ¹⁵	Abholen langfristige Vormerkposten	-1	7	MT942
WPA	Abholen Wertpapierabrechnung	-1	7	MT510, MT515
WPB	Abholen Wertpapierausführungsanzeige	-1	7	MT519, MT513
WPC	Abholen Depotaufstellung	-1	7	MT571, MT535
WPD	Abholen sonstige WP-Umsätze	-1	7	MT572, MT536

1.4.2 Kategorie 2: Systembedingte Auftragsarten

Kennung	Text	Satzlänge ¹⁶	Bits	Format
BPD	Bankparameterdatei abholen (automatische Abholung durch das Kundenprodukt)	-1	8	
INI	Passwort-Initialisierung	512	8	Public Key des Kunden für die EU; siehe Kapitel 2 „Standards für die Sicherheit“
PTK	Abholen Kundenprotokoll	-1	7	siehe 1.7 „Kundenprotokoll - inhaltliche und formale Festlegungen“
PUB	Senden Public Key zur Unterschriftenverifizierung	512	8	Public Key des Kunden für die EU; siehe Kapitel 2 „Standards für die Sicherheit“
PWA	Passwort-Änderung senden	1	7	Übertragung einer Dummy-Datei, die nur ein Leerzeichen enthält
SPR	Sperren der Zugangsberechtigung	1	7	Übertragung einer Dummy-Datei, die nur ein Leerzeichen enthält

¹⁵ Diese Auftragsart wird nur noch bis zum 31.12.2004 unterstützt.

¹⁶ Die Satzlänge „-1“ bedeutet „variable Satzlänge“.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Kennung	Text	Satzlänge ¹⁶	Bits	Format
VPB	Abholen Public Key der Bank zur Verschlüsselung	512	8	Public Key der Bank zur Verschlüsselung (siehe Kapitel 2.3.3 „Vorbereitung der Verschlüsselung / Public-Key-Austausch“)
VPK	Senden Public Key des Kunden zur Verschlüsselung	512	8	Public Key (siehe Kapitel 2.3.3 „Vorbereitung der Verschlüsselung / Public-Key-Austausch“)

1.4.3 Kategorie 3: Reservierte Auftragsarten für den zwischenbetrieblichen Zahlungsverkehr/Dateiaustausch

Kennung	Text	Satzlänge ¹⁷	Bits	Format
FIN	EDIFACT-FINPAY senden	-1	7	EDIFACT
IZS	Informationen von Zentralstellen	80	7	
SSP	ec-Karten-Sperrdatei	80	8	

1.4.4 Kategorie 4: Sonstige reservierte Auftragsarten unter Verwendung nicht standardisierter Formate und Verfahren

Kennung	Text	Satzlänge ¹⁷	Bits	Format
ATA	Teilausnutzung Importakkreditiv (Kreditinstitut an Kunde)	-1	7	
BKA ¹⁸	Auftragsart für elektronische Kontoauszüge	-1	8	
BZK	Barzahlungskarte			
DKI	Devisenkursinformationen abholen (Euro)	-1	8	
DMI	Abholen Devisenmarktinformationen	-1	7	
DSW	Abholen Devisenswapinformationen	-1	7	
ESG	ESG-Datei für Elektronische Zweitunterschrift abholen	-1	8	

¹⁷ Die Satzlänge „-1“ bedeutet „variable Satzlänge“.

¹⁸ Die Anforderungen (Geschäftsprozesse, Formate) an einen steuerrechtlich anerkannten elektronischen Kontoauszug sind noch nicht definiert. Reservierung dieser Auftragsart im Vorgriff auf zukünftige diesbezügliche gesetzgeberische Entscheidungen.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Kennung	Text	Satzlänge ¹⁷	Bits	Format
ESP	ESP-Datei für Elektronische Zweitunterschrift senden	-1	8	
FTB	Abholen beliebige Datei	-1	8	General-stream ASCII
FTB	Senden beliebige Datei	-1	8	General-stream ASCII
FTD	Freie Textdatei senden	-1	7	ASCII
FTD	Freie Textdatei abholen	-1	7	ASCII
IBI	Abholen Antwort auf Informationsabfrage	-1	8	
IBK	Abholen Institutsbestätigungsdatei Komplettbestand	-1	8	
IBW	Abholen Institutsbestätigungsdatei Komplettbestand weitere Datei	-1	8	
IBU	Abholen Institutsbestätigungsdatei tägliches Update	-1	8	
IKI	Senden Informationsabfrage	-1	8	
IKK	Senden Institutskonten Komplettbestand begrenzt auf 170 MB	-1	8	
IKU	Senden Institutskonten tägliches Update	-1	8	
IKW	Senden Institutskonten Komplettbestand weitere Datei	-1	8	
KTH	KTOHIN Automatisiertes Verfahren für die Änderung von Kontonummern und Bankleitzahlen	100	8	EBCDIC
KTR	KTORUECK Automatisiertes Verfahren für die Änderung von Kontonummern und Bankleitzahlen	100	8	EBCDIC
KKZ	Kontenkonzentration und Saldenausgleich			
TST	Senden Testdatei ASCII	-1	7	ASCII
TST	Abholen Testdatei ASCII	-1	7	ASCII
UPD	Updates abholen ¹⁹	-1	8	

¹⁹ Bereitstellung von Updates durch das Kreditinstitut und Abholung durch das Kundensystem, signalisiert durch die Auftragsart UPD.

1.5 Fehlermeldungen/Fehlercodes

Beschreibung der Ursache für die Ablehnung des Transfer-Auftrages oder Quittung auf Aufträge, die keinen File-Transfer erfordern (z. B. INI, PWA) (maximal 120 Bytes alphanumerisch).

Aufbau:

- Byte 1 – 40: freier Fehlertext (zur Anzeige am PC)
- Byte 41 – 44: Fehlercode
- Byte 45 – 120: nicht normiert

Zur Zeit sind folgende Fehlercodes definiert:

Nummer	Bedeutung
1	Auftrag durchgeführt (Spezifizierung des Auftrags erfolgt im Fehlertext)
2	User-ID nicht registriert
3	Falsches Passwort
5	User-ID gesperrt
7	Unzulässige Auftragsart
8	User-ID nicht initialisiert
13	(noch) keine Daten vorhanden; später versuchen
15	Keine Berechtigung für diese Auftragsart
16	Formalfehler
17	Sperrung der User-ID nach 3 Fehlversuchen
24	Keine Daten vorhanden
25	User-ID noch nicht freigegeben
26	Nicht normierter Fehler, Wiederholung nicht sinnvoll
27	Nicht normierter Fehler, Wiederholung sinnvoll
29	Abbruch der gesamten DFÜ-Verbindung
52	EU-Version wird nicht mehr unterstützt ²⁰

²⁰ Bei der Einreichung von INI- und PUB-Aufträgen prüft das Banksystem anhand der Public-Key-Datei (Feld "Versionsnummer") die vom Kunden genutzte EU-Version. Wenn die Prüfung auf dem Banksystem ergibt, dass die EU-Version nicht mehr unterstützt wird, wird der Auftrag abgelehnt. Das Banksystem gibt zur Quittierung den Antwortcode 52 an das Kundensystem zurück und erzeugt einen entsprechenden Eintrag im Kundenprotokoll.

1.6 Betrieb über asynchrone Verbindungen und PAD

Die Spezifikation behandelt das Verfahren zur Erkennung und Beseitigung von Datenverfälschungen bei Datenübertragungen über Telefonverbindungen.

Es werden die Protokollelemente und der Ablauf spezifiziert, weiterhin die Konfigurationsparameter und die Formate der Kontroll- und Dateneinheiten erläutert. Außerdem sind die benötigten Algorithmen zur Erkennung von Datenverfälschungen spezifiziert.

1.6.1 Anforderungen an das Verfahren

- Der Kommunikationsweg geht vom Modem über X.28/X.29-Protokoll zu einem PAD (in-house oder öffentlich) und von diesem PAD über eine normale X.25-Verbindung zum Partner.
- Es wird ein Verfahren spezifiziert, das unterhalb der OSI-Transportschicht arbeitet.
- Es besteht die Notwendigkeit, Datenübertragungen über Strecken, die nur einen 7-bit Zeichensatz unterstützen, zu realisieren. Außerdem sollten Zeichen, die eventuell eine Bedeutung als PAD-Steuerzeichen haben, nicht direkt übertragen werden.
- Da die höheren Protokolle (FTAM) 8-bit-Zeichen benötigen, wird ebenfalls unterhalb der Transportschicht ein Mechanismus zur Maskierung/Demaskierung der nicht übertragbaren Zeichen spezifiziert.

1.6.2 Kurzbeschreibung der Lösung

Prinzipiell kann bei unsicheren Verbindungen folgendermaßen gearbeitet werden:

- Zu übertragende Daten werden mit einer Checksumme gesichert. Auf der Empfangsseite muss die Checksumme gegengerechnet werden, und Quittungen benachrichtigen den Sender, ob die Übertragung erfolgreich war oder nicht.
- Da auch Quittungen verfälscht werden können, muss der Sender das Eintreffen von Quittungen durch Timer überwachen.
- Das Sicherungsverfahren selbst ist als eigendefinierte Protokollschicht zwischen der OSI-Transportschicht und der Netzwerkschicht angesiedelt. Das Verfahren versendet eigene Kontroll- und Dateneinheiten, in denen die Übertragungseinheiten der Transportschicht (TPDU) als Netzwerkdaten (NSDU) codiert sind.
- Umgekehrt empfängt das Verfahren Daten der Netzwerkschicht (NSDU), überprüft sie auf Korrektheit und behandelt sie als Elemente des eigenen Protokolls. Der Inhalt der Dateneinheiten wird decodiert und in dieser Form als TPDU an die Transportschicht hochgereicht.

1.6.3 Spezifikation

1.6.3.1 Kontroll- und Dateneinheiten

Folgende Einheiten (NSDU) werden spezifiziert:

- Austausch von Steuerparametern
 - CONNECT Mitteilung über gewünschte Steuerparameter durch den Initiator der Netzwerkverbindung
 - ACCEPT Bestätigung oder Modifizierung der Steuerparameter durch den passiven Partner.
- Datenaustausch
 - DATA Enthält in codierter Form sowohl Transport-Steuer-PDU als auch Transport-Daten-PDUs.
- Quittungen
 - ACK Positive Quittung für eine DATA NSDU.
 - NAK Negative Quittung für eine DATA NSDU

1.6.3.2 Protokollabläufe

Grundsätzlicher Ablauf:

- Nach Aufbau der Netzwerkverbindung sendet der Initiator eine CONNECT-NSDU, in der der Initiator dem Partner mitteilt, welche Protokollversion und welchen Übertragungsmodus (7-bit oder 8-bit) er benutzen will.
- Der Partner sendet nach Empfang der CONNECT-NSDU eine ACCEPT-NSDU, in der der Partner die vorgeschlagenen Steuerparameter bestätigt oder modifiziert (vergleiche 1.6.3.5 Datenformate der Kontroll- und Dateieinheiten, CONNECT und ACCEPT). CONNECT- und ACCEPT-NSDUs werden grundsätzlich nur einmal zu Beginn der Netzdatenphase ausgetauscht. Es ist jedoch möglich, dass die NSDUs verfälscht werden. Aus diesem Grund überwacht der Initiator den Empfang der ACCEPT-NSDU mit einem Timer, um die CONNECT-NSDU gegebenenfalls erneut zu versenden. Der Partner muss zu Beginn der Datenphase mehrere CONNECT-NSDUs empfangen können und er muss diese jeweils durch eine ACCEPT-NSDU bestätigen können.
- Erst nach Austausch der CONNECT- und ACCEPT-NSDU können Daten- und Quittungs-NSDUs gesendet werden.
- Das Quittungsverfahren für empfangene DATA-NSDUs gestaltet sich wie folgt:
 - Der Empfänger quittiert sofort jede empfangene DATA-NSDU
 - bei Korrektheit der Daten mit einer ACK-NSDU
 - bei verfälschten Daten mit einer NAK-NSDU.

Erhält der Sender eine NAK-NSDU, so wiederholt der Sender die zuletzt gesendete DATA-NSDU.

Da auch Quittungen verfälscht sein können, muss der Sender die zuletzt gesendete DATA-NSDU wiederholen, wenn nach einer bestimmten Zeitspanne keine Quittung eingetroffen ist. Umgekehrt muss der Empfänger doppelt empfangene Daten zwar positiv bestätigen, sie ansonsten aber ignorieren.

1.6.3.3 Mögliche Abläufe beim Austausch von Dateneinheiten

A und B bezeichnen die beiden beteiligten Kommunikationsinstanzen, die Zeit läuft von oben nach unten. Die Zahlen bedeuten eine Durchnummerierung der Einheiten.

Mit Fehler ist im Folgenden gemeint, dass eine der folgenden Bedingungen zutrifft:

- Eine NSDU verstößt gegen das spezifizierte NSDU-Format.
- Die Länge einer NSDU ist falsch.
- Der Typ einer NSDU kann nicht erkannt werden.
- Ein Sequenzfehler wurde erkannt.
- Bei der Gegenrechnung der Checksumme wird ein Fehler erkannt.

Beispiel 1 (Normalfall): Die Übertragung von A nach B und die Übertragung der Quittung ist erfolgreich. Wenn vorhanden, kann A mit der Übertragung der nächsten DATA-NSDU fortfahren.

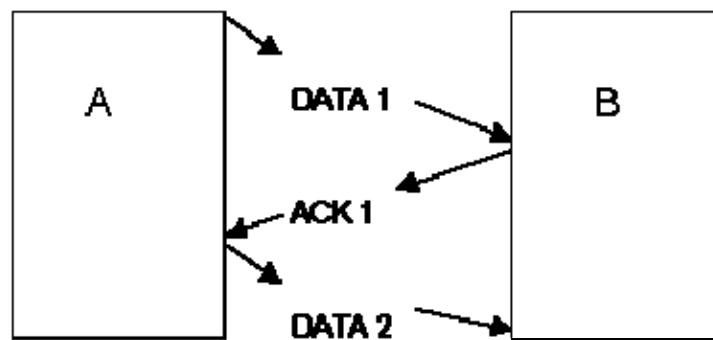


Abbildung 1.3 Mögliche Abläufe beim Austausch von Dateneinheiten – Beispiel 1

Beispiel 2 (B erkennt Fehler in den von A gesendeten Daten): B erkennt einen Fehler in den von A gesendeten Daten und schickt eine negative Quittung. A muss dieselbe NSDU noch einmal versenden.

Unter Umständen interpretiert der Empfänger eine verfälschte Kontroll-NSDU (CONNECT, ACCEPT, ACK oder NAK) als Daten-NSDU und sendet ein NAK mit einer Sequenznummer (BSN = Block Sequence Number), die noch gar nicht verschickt wurde. Ein solches NAK muss vom Sender ignoriert werden.

Randbedingungen, wie z. B. maximale Anzahl der Versuche, sind in Kapitel **1.6.3.4 Konfigurationsparameter** spezifiziert.

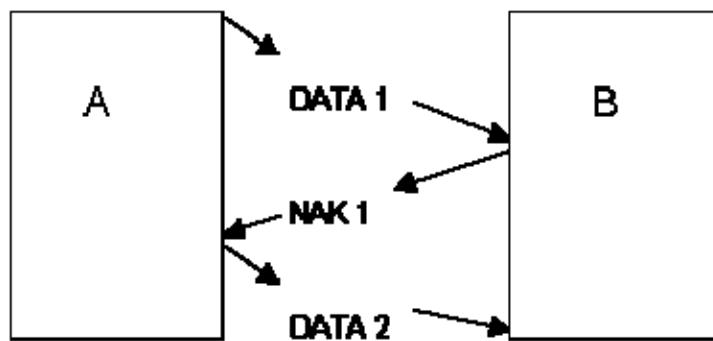


Abbildung 1.4 Mögliche Abläufe beim Austausch von Dateneinheiten – Beispiel 2

Beispiel 3 (Daten erreichen B nicht oder sind nicht als solche zu erkennen): A schickt Daten, bei B kommt überhaupt nichts an, bzw. die Daten sind so verfälscht, dass sie nicht als solche erkannt werden können. In diesem Fall läuft der Quittungsüberwachungstimer bei A ab, und A sendet die NSDU erneut.

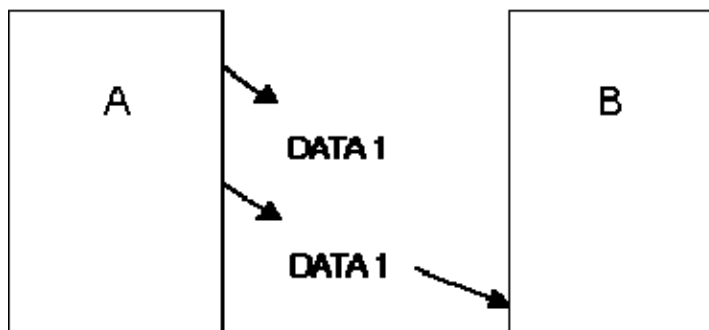


Abbildung 1.5 Mögliche Abläufe beim Austausch von Dateneinheiten – Beispiel 3

Beispiel 4 (Quittung von B erreicht A nicht): A schickt Daten, die von B (positiv oder negativ) quittiert werden. Beim Empfang der Quittung wird entweder ein Fehler erkannt, oder die Quittung wird überhaupt nicht empfangen. In diesem Fall läuft der Quittungsüberwachungstimer bei A ab, und dieselbe NSDU wird noch einmal gesendet.

Aus diesem Fall folgt, dass B erkennen muss, wenn zweimal hintereinander dieselbe NSDU empfangen wird. Die Kopie kann einfach vernichtet werden, muss aber quittiert werden.

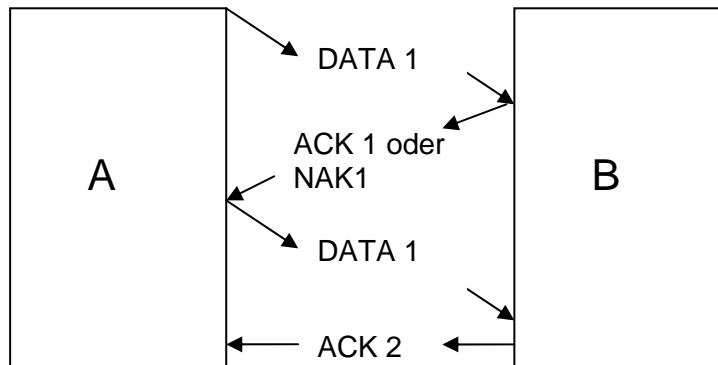


Abbildung 1.6 Mögliche Abläufe beim Austausch von Dateneinheiten – Beispiel 4

Protokollverstöße: Folgendes gilt als Protokollverstoß und führt zum Abbau der Verbindung:

- wenn NSDUs des Datenaustauschs (DATA, ACK, NAK) empfangen werden, bevor CONNECT- und ACCEPT-NSDUs ausgetauscht worden sind
- wenn CONNECT- oder ACCEPT-NSDUs empfangen werden, nachdem bereits Dateneinheiten empfangen wurden, so
- wenn Daten mit falscher Sequenznummer empfangen werden (Quittungen mit falscher Sequenznummer werden jedoch ignoriert).

1.6.3.4 Konfigurationsparameter

Die folgende Tabelle beschreibt die variabel konfigurierbaren Parameter für die lokale Instanz der beschriebenen Protokollschicht.

Parameter	Default	Bedeutung
BCS_ACK_TIMEOUT	5	Zeitspanne (in Sekunden), nach der eine Quittung empfangen worden sein muss. Nach Ablauf dieser Zeitspanne muss die unquittierte DATA-NSDU oder die unbestätigte CONNECT-NSDU wiederholt werden.
BCS_N_MAX_REP	5	Maximale Anzahl der Übertragungsversuche für eine NSDU. Ist die maximale Anzahl der Wiederholversuche überschritten, wird die Verbindung abgebaut. In diesem Fall sorgt die FTAM Recovery-Prozedur für den Wiederanlauf der Datenübertragung.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Parameter	Default	Bedeutung
BCS_G_MAX_REP	30	Maximale Anzahl der Wiederholungen für eine gesamte Verbindung. Ist die maximale Anzahl der Wiederholversuche überschritten, wird die Verbindung abgebaut. In diesem Fall sorgt die FTAM Recovery-Prozedur für den Wiederanlauf der Datenübertragung.
BCS_MODE	8	Bestimmt, ob die Übertragungstrecke im 7-bit- oder 8-bit-Modus arbeitet.

Die Werte können über Environmentvariablen (Umgebungsvariablen) eingestellt werden. Für eine nicht gesetzte Environmentvariable wird automatisch der Default-Wert angenommen.

1.6.3.5 Datenformate der Kontroll- und Dateieinheiten

Dieses Kapitel beschreibt die Formate der Einheiten (NSDU), die nach dem Protokoll (vgl. vorausgegangene Kapitel) ausgetauscht werden.

Der Aufbau einer NSDU richtet sich nach dem allgemeinen Schema:

Anzahl Bytes	1	1	1	n	2	1
	SOH	type	BSN	Daten	BCS	CR

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Bezeichnung	Bedeutung / Wert
SOH	Damit beginnt eine neue NSDU, SOH hat den Wert 0X01.
Type	Bestimmt den Typ einer NSDU
BSN	Sequenznummer der NSDU. Der Wert ist numerisch und wird als ASCII-Codierung der Zahl übertragen (bei den CONNECT- und ACCEPT-NSDUs wird kein BSN verwendet).
Daten	<i>Bei Data-NSDUs:</i> Zu übertragende Daten (codierte TPDUs). <i>Bei CONNECT- und ACCEPT-NSDUs</i> Steuerparameter (siehe Beschreibung von CONNECT und ACCEPT in diesem Kapitel) <i>Bei ACK- und NAK-NSDUs</i> Quittungs-NSDUs haben keinen Datenteil.
BCS	Checksumme (vgl. Beschreibung des Algorithmus in Kapitel 1.6.3.6 „Berechnung der Checksumme und Codierung der NSDU im 7- oder 8-bit Modus“)
CR	Beendet immer eine NSDU. CR hat den Wert 0x0d.

Die Codierung der Typenfelder erfolgt durch weit auseinanderliegende Bitmuster, um ein Verwechseln der Typen durch einzeln umkippende Bits unwahrscheinlich zu machen.

CONNECT:

Diese NSDU wird vom Initiator direkt nach Aufbau der Netzverbindung gesendet.

Anzahl Bytes	1	1	1	1	1	2	1
	SOH	0x20	version	window	bitmode	BCS	CR

Bezeichnung	Bedeutung	Wert
version	Gibt die Version des Sicherungsverfahrens des Initiators an	0X31
window	Sendefenster des Initiators	immer 0X31
bitmode	Gibt an, ob der Initiator den 7- oder 8-bit Übertragungsmodus vorschlägt	0X31 oder 0X38

ACCEPT:

Diese NSDU wird vom Responder nach Empfang der CONNECT-NSDU gesendet.

Anzahl Bytes	1	1	1	1	1	1	2	1
	SOH	0x23	result	version	window	bitmode	BCS	CR

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Bezeichnung	Bedeutung	Wert
result	Gibt an, ob der Responder die Verbindung annimmt	0X20: Aufbau OK
version	Gibt die Version des Sicherungsverfahrens des Responders an	0X31
window	Sendefenster des Responders	immer 0X31
bitmode	Gibt an, ob der Responder den 7- oder 8-bit Übertragungsmodus wählt	0X31 oder 0X38

Die Parameter version und bitmode werden ausgehandelt, d. h. der Responder kann kleinere Werte setzen, als der Initiator vorgeschlagen hat. Das bedeutet, version auf eine ältere Version und/oder bitmode von 8 auf 7 herunterzuhandeln.

Die in der ACCEPT-NSDU angegebenen Werte für version und bitmode bestimmen damit das Sicherungsverfahren für die Dauer der Verbindung.

DATA:

Anzahl Bytes	1	1	1	n	2	1
	SOH	0x40	BSN	Daten	BCS	CR

Bezeichnung	Bedeutung	Wert
BSN	beinhaltet die Sequenznummer dieser NSDU	0X30 bis 0X39 (siehe Beschreibung der BSN im allgemeinen NSDU-Schema, s.o.)

ACK:

Anzahl Bytes	1	1	1	2	1
	SOH	0x71	BSN	BCS	CR

Bezeichnung	Bedeutung	Wert
BSN	beinhaltet die Blocknummer der nächsten erwarteten DATA-NSDU	0X30 bis 0X39 (siehe Beschreibung der BSN im allgemeinen NSDU-Schema, s.o.)

NAK:

Anzahl Bytes	1	1	1	2	1
	SOH	0x7f	BSN	BCS	CR

Bezeichnung	Bedeutung	Wert
BSN	beinhaltet die Blocknummer der nächsten erwarteten DATA-NSDU	0X30 bis 0X39 (siehe Beschreibung der BSN im allgemeinen NSDU-Schema, s.o.)

1.6.3.6 Berechnung der Checksumme und Codierung der NSDU im 7- oder 8-bit Modus

Dieses Kapitel beschreibt die Codierung einer NSDU, also auch den Übergang von Daten einer TPDU zu Daten einer NSDU. Der Vorgang beinhaltet folgende Schritte:

- Erstelle SOH
- Berechne BCS
- Maskiere verbotene Zeichen (Shift-Out- bzw. Shift-In-Logik)
- Erstelle BCS und CR.

Algorithmus zur Berechnung der BCS:

Der Algorithmus zur Berechnung der BCS folgt dem ISO TRANSPORT LAYER Standard.

Verwendete Symbole	Bedeutung / Wert
CO, C1	Variablen, die im Algorithmus benötigt werden
X	Wert des ersten Bytes der BCS
Y	Wert des zweiten Bytes der BCS
N	Anzahl der zu versendenden Bytes ohne SOH und CR

Der Algorithmus beinhaltet folgende Schritte:

- Initialisieren von CO und C1 mit Null.
- Bearbeiten jedes Bytes von $i = 1$ bis n mit:
 - Addieren des Byte-Wertes zu CO und anschließend
 - Addieren des Wertes von CO zu C1.
- Berechnen von X und Y mit
$$X = CO - C1$$
$$Y = C1 - 2 * CO$$
- Speichern der Werte X und Y in den Bytes 1 und 2 der BCS.

Algorithmus zur Maskierung der Zeichen (Sender):

Verwendete Symbole	Wert
SO	0X0e
SI	0X0f
n	Anzahl der zu versendenden Bytes, einschließlich BCS, jedoch ohne SOH und CR

```
for (i = 1 to n) {
  x = buffer [i];
  if ((7 < bit_mode) && (0x80 <= x)) {
    x = x & 0x7f;
    if {x < 0x20} {
      output (SO);
      output (x + 0x40);
    } else {
      output (SI);
      output (x);
    }
  } else {
    if ((x & 0x7f) < 0x20) {
      output (SO);
      output (x + 0x20);
    }
    else {
      output (x);
    }
  }
}
```

Abbildung 1.7 Der Algorithmus zur Maskierung

Anmerkung: Unabhängig vom gewünschten Übertragungsmodus muss die CONNECT-NSDU immer im 7-bit-Modus codiert sein.

1.6.3.7 Decodierung und Prüfung der NSDU

Eine empfangene NSDU wird in folgenden Schritten bearbeitet:

- Prüfe das Format der NSDU
- Demaskiere verbotene Zeichen (Shift-Out- bzw. Shift-In-Logik)

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- Prüfe BCS
- Entferne SOH, BCS und CR.

Algorithmus zur Prüfung der BCS:

Der Algorithmus zur Prüfung der BCS folgt dem ISO TRANSPORT LAYER Standard.

Verwendete Symbole	Wert
CO, C1	Variablen, die im Algorithmus benötigt werden
n	Anzahl der zu demaskierenden Bytes einer empfangenden NSDU, einschließlich BCS, jedoch ohne SOH und CR

Der Algorithmus beinhaltet folgende Schritte:

- Initialisieren von CO und C1 mit Null.
- Bearbeiten jedes Bytes von $i = 1$ bis n mit
 - Addieren des Byte-Wertes zu CO und anschließend
 - Addieren des Wertes von CO zu C1.
- Wenn alle Bytes (einschließlich BCS) bearbeitet sind und einer oder beide Werte von CO und C1 nicht Null sind, liegt ein Checksummenfehler vor.

Es liegt in der Natur des Algorithmus, dass es nicht notwendig ist, explizit die gespeicherten Bytes der Checksumme zu vergleichen.

Algorithmus zur Demaskierung der Zeichen (Empfänger):

Verwendete Symbole	Wert
SO	0X0e
SI	0X0f
n	Anzahl der zu versendenden Bytes, einschließlich BCS, jedoch ohne SOH und CR

```
for (i = 1 to n) {
  x = buffer [i];
  if (x < 0x20) {
    switch (x) {
      case S0:
        i = i + 1;
        y = buffer [i];
        if ((y>=0x20) && (y<0x40) || ((y>=0xa0) && (y<0xc0)))
          output (y - 0x20);
        else {
          if ((y >= 0x40) && (y < 0x60))
            output ((y - 0x40) I 0x80);
          else
            error();
        }
        break;
      case S1:
        i = i + 1;
        y = buffer [i];
        if ((y >= 0x20) && (y < 0x80))
          output (y I 0x80);
        else
          error();
        break;
      default:
        error();
        break;
    }
  } else {
    if ((x >= 0x80) && (x < 0xa0))
      error();
    else
      output (x);
  }
}
```

Abbildung 1.8 Der Algorithmus zur Demaskierung

1.6.4 Abkürzungsverzeichnis

- ACK Acknowledge
- BCS Block CheckSum (Checksumme)
- BSN Block Sequence Number (Sequenznummer)
- CR Carriage Return
- FTAM File Transfer Access and Management
- NAK Negative Acknowledge
- NSDU Network Layer Service Data Unit
- OSI Open Systems Interconnection
- PAD Packet Assembler Disassembler
- PC Personal Computer
- PDU Protocol Data Unit
- TPDU Transport Protocol Data Unit
- SI Shift-In
- SO Shift-Out
- SOH Start of Header

1.7 Kundenprotokoll - inhaltliche und formale Festlegungen

Das Kundenprotokoll ist auf der Bankseite gemäß den folgenden Festlegungen zu erstellen. Es gelten hierbei folgende grundsätzliche Bestimmungen:

- Es dürfen in einer Zeile maximal 72 Zeichen ausgegeben werden.
- Es erfolgt kein Protokolleintrag über die Weiterverarbeitung. (Ausnahmen: EU-Prüfung, Fehler bei Dekomprimierung, Anzeige Dateinhalt)
- Der A- und E-Satz (bei Dateien im DTAUS-Format, z. B. IZV-Dateien), der Q-Satz und Z-Satz (bei Dateien im DTAZV-Format, z. B. AZV-Dateien) und der erste und letzte logische Satz (bei allen anderen Dateitypen) wird auch bei Dateien ohne EU ausgegeben²¹.

1.7.1 Inhaltliche Festlegungen

Dateiname des Kundenprotokolls:

Als Dateiname wird die jeweilige Kunden-ID mit der Extension „PTK“ verwendet:

<KUNDEN-ID>.PTK

Auflistung der einzelnen Datenfelder je Aktion auf der Bankseite:

Je Aktion auf der Bankseite sind folgende Daten im Kundenprotokoll zu dokumentieren:

Zu dokumentierende Daten	Beschreibung
Datum und Uhrzeit	Datum und Uhrzeit der Aktion auf dem Banksystem
Art der Aktion	Siehe Kap. 1.7.3 Liste der möglichen Meldungen inklusive Textnummer
Hostname	
Auftragsart	Klartext zu der vom Kunden benutzten Auftragsart, auf die sich die jeweilige Aktion der Bank bezieht. Beispiel: „Freie Textdatei im 7 Bit-Code senden“; siehe Kapitel 1.4 „Auftragsartenkennungen“

Folgende Felder sind bei EU-Prüfung gegebenenfalls mehrfach (d. h. je User) vorhanden:

- User-ID
- User-Name (nur soweit vorhanden)
- Auftragsnummer (Byte 23 bis 26 aus dem Remote-Filename; siehe Kapitel 1.1. „Anwendungsprotokoll Kunden-/Bankrechner“)

²¹ Entfällt bei unstrukturierten Dateien

- Ergebnis der Aktion (siehe Kapitel 1.7.3 „Liste der möglichen Meldungen inklusive Textnummer“)

Folgende Einträge sind (mit Ausnahme der Dateianzeige) nur bei EU-Prüfungen vorhanden:

Eintrag	Beschreibung
Dateiname auf Kundensystem	„Dateiname der Originaldatei“ aus EU-Datei; siehe Kapitel 2.2.2.5 „Formate“
Dateianzeige	Auftragsarten (Dateien im DTAUS-Format): Anzeige der wesentlichen Dateidaten ²² entsprechend dem Inhalt der Datenträgerbegleitzettel; siehe Kapitel 1.7.4 „Dateianzeige auf Kunden- und Bankseite“ Sonstige Auftragsarten ²³ : Bei <u>Dateien mit fester Satzlänge</u> wird der erste und letzte Satz gemäß der je Auftragsart spezifizierten Satzlänge angezeigt. Bei <u>Dateien mit variabler Satzlänge</u> wird der für das jeweilige Betriebssystem definierte erste und letzte logische Satz angezeigt (z.B. Satz vor dem ersten CR/LF, z.B. Satz vor dem letzten CR/LF).
Erläuternder Text im Fehlerfall	Dieses Feld wird nur angezeigt, wenn das Ergebnis der Aktion „EU-Prüfung“ einen Fehler anzeigt. Es ist damit als Subfeld hierzu zu verstehen, das den konkreten Fehlerfall (gegebenenfalls je User und je logischer Datei) erläutert; Beispiel: „Vereinbarter Höchstbetrag ueberschritten“, siehe Kapitel 1.7.3 „Liste der möglichen Meldungen inklusive Textnummer“.

1.7.2 Formale Festlegungen

Die formale Gestaltung des Kundenprotokolls erfolgt gemäß folgenden Festlegungen:

1.7.2.1 Protokollierung der Aktionen auf der Bankseite

1. Zeile:

- Datum (tt.mm.jj)
- 1 Leerzeichen
- Uhrzeit (hh:mm:ss)

²² Bei „Sammel-EU“ (mehrere logische Dateien mit einer EU) erfolgt die Anzeige je logischer Datei

²³ Bei 8-Bit-Dateien wird der erste und letzte Satz als HEXDUMP ausgegeben.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- 5 Leerzeichen
- Art der Aktion (maximal 50 Zeichen)

Beispiel:

```
14.11.02 11:39:05      Datei zur Bank uebertragen
```

3. Zeile:

- 9 Leerzeichen Einrückung (aus Übersichtlichkeitsgründen)
- Text: „Auftrag“ (= Art des Protokolleintrags)
- 4 Leerzeichen
- Doppelpunkt (dieses Zeichen steht immer an der 21. Stelle)
- 1 Leerzeichen
- Text der Auftragsart (42 Zeichen, ggf. mit Leerzeichen aufgefüllt)
- Auftragsartenkennung (3stellig)
- 1 Leerzeichen
- Auftragsnummer (4stellig)

Beispiel:

```
      Auftrag      : Freie Textdatei im 7Bit-Code senden      FTD A000
```

Sonstige Zeilen:

- 9 Leerzeichen Einrückung
- Art des Protokolleintrags (maximal 11 Zeichen)
- Doppelpunkt (dieses Zeichen steht immer an der 21. Stelle)
- 1 Leerzeichen
- Text des jeweiligen Protokolleintrages (maximal 50 Zeichen)

Beispiele:

```
      Hostname     : BVFTAMU
```

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

```
Ergebnis      : Uebertragung in Ordnung
                Datenerübertragung unverschlüsselt
                Datenübertragung unkomprimiert
```

Durch Anfügen zweier zusätzlicher Textzeilen an die Ergebniszeile wird dokumentiert, dass die DFÜ-Aufträge ohne bzw. mit Verschlüsselung und Komprimierung abgewickelt wurden. Die erste zusätzliche Zeile dokumentiert die Verschlüsselung, die zweite Zeile die Komprimierung des DFÜ-Auftrages.

1. Zeile:

- 22 Leerzeichen Einrückung
- Text: „Datenerübertragung verschlüsselt“ oder „Datenerübertragung unverschlüsselt“

2. Zeile:

- 22 Leerzeichen Einrückung
- Text „Datenerübertragung komprimiert“ oder „Datenerübertragung unkomprimiert“

Beispiele für die Protokollierung gesamt:

```
14.11.02 11:40:05   Datei zur Bank uebertragen
                   Hostname      : HOSTFTAM
                   Auftrag       : Beliebige Datei senden           FTB AAI0
                   Teilnehmer    : USER Teilnehmer User
                   Ergebnis      : Uebertragung in Ordnung
                                 Datenerübertragung unverschlüsselt
                                 Datenübertragung unkomprimiert
```

```
14.11.02 11:44:15   Datei zur Bank uebertragen
                   Hostname      : HOSTFTAM
                   Auftrag       : Beliebige Datei senden           FTB AAJ0
                   Teilnehmer    : USER Teilnehmer User
                   Ergebnis      : Uebertragung in Ordnung
                                 Datenerübertragung verschlüsselt
                                 Datenübertragung komprimiert
```

1.7.2.2 Protokollierung der Fehler bei Unterschriftsprüfung

Teilnehmerbezogene Fehler bei Unterschriftsprüfung:

- 9 Leerzeichen Einrückung
- Text: „EU von“
- 1 Leerzeichen

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- USER-ID (maximal 8 Zeichen)
- Doppelpunkt (dieses Zeichen steht an der 26. Stelle)
- 1 Leerzeichen
- Fehlertext (maximal 45 Zeichen)

Beispiel:

```
EU von BLUMPC : Vereinbarter Hoechstbetrag ueberschritten
```

Allgemeine Fehlertexte bei Unterschriftsprüfung:

- 9 Leerzeichen Einrückung
- Fehlertext (maximal 63 Zeichen)

Beispiel:

```
Die erforderliche Anzahl von EUs ist nicht vorhanden
```

1.7.2.3 Dateianzeige

- 4 Leerzeichen Einrückung
- Dateianzeige

Beispiel siehe 1.7.4 Dateianzeige auf Kunden- und Bankseite

1.7.2.4 Einfügen individueller Texte

In die Kundenprotokolldatei können auch bankindividuelle Texte eingefügt werden. Diese Texte können z.B. Verarbeitungsinformationen vom Hostsystem der Bank oder spezifische Kundeninformationen enthalten. Damit die PTK-Dateien maschinell auswertbar bleiben, werden diese Informationen entsprechend gekennzeichnet:

Zur Kennzeichnung enthält die erste Zeile der individuellen Texte immer die statische Anfangskennung "ZUSATZINFORMATION" und wird inklusive Zeitstempel wie die 1. Zeile eines PTK-Eintrags als „Art der Aktion“ eingestellt (vergl. Kapitel 1.7.2.1 „**Protokollierung der Aktionen auf der Bankseite**“). Als Endmarkierung genügt wie bei allen PTK-Einträgen der Zeitstempel des folgenden PTK-Eintrags.

Beispiel:

```
26.10.05 11:15:00      ZUSATZINFORMATION
=====
WIR MOECHTEN SIE AUF DIESEM WEG DARUEBER INFORMIEREN, DASS
DIE NUTZUNG DER ELEKTRONISCHEN UNTERSCHRIFT MIT A003-
SCHLUESSELN NUR ...

WE WOULD LIKE TO INFORM ...
=====
```

1.7.2.5 Unterstützung fremdsprachiger Kundenprotokolle

Das Kundenprotokoll kann nicht nur in deutscher Sprache, sondern auch optional in anderen Sprachen generiert werden.

In diesem Zusammenhang ist zu beachten, dass im Protokoll enthaltene Informationen, die nach Abholung auf der Kundenseite maschinell ausgewertet werden (z. B. Ergebnisse der EU-Prüfungen), gesondert gekennzeichnet werden müssen. Auf diese Weise kann gewährleistet werden, dass die maschinelle Auswertung der in verschiedenen Sprachen erzeugten Protokolle in der Kundensoftware funktioniert.

Zu diesem Zweck werden alle Informationen, die für eine maschinelle Auswertung von Bedeutung sind, durch Anfügen einer eindeutigen, 2stelligen Nummer gekennzeichnet. Der eigentliche Text wird von der eindeutigen Nummer durch ein Leerzeichen getrennt. Die Nummer wird durch Klammern „[]“ begrenzt.

Nach Durchführung eines Protokollabrufes können die eindeutigen Nummern dann durch das Kundensystem im Rahmen der maschinellen Auswertung unabhängig von der jeweiligen Sprache entsprechend interpretiert werden.

Für die Texte im Kundenprotokoll, die für eine maschinelle Auswertung in Frage kommen, ergibt sich somit folgender Aufbau:

TTTX'20'[NN]

- TTT eigentlicher Text
- X'20 Leerzeichen als Trennung zwischen Text und Nummer
- [NN] 2stellige, in Klammern gesetzte Nummer, die eindeutig sein muss

Für die maschinelle Auswertung kommen generell die Texte in Frage, die das Ergebnis des DFÜ-Auftrages inklusive Unterschriftsprüfung ausweisen. In der folgenden Tabelle sind die Textnummern und die dazugehörigen Texte, die für eine maschinelle Auswertung in Frage kommen, aufgelistet. Die einzelnen Texte sind in die Bereiche „Datenfernübertragung“, „Elektronische Unterschrift“, „Dateibezogene Nachverarbeitung“ sowie „Bankfachliche Prüfungen“ der Übersichtlichkeit halber unterteilt.

Textnummer	Text
Datenfernübertragung (Bereich 1-20)	
01	Übertragung in Ordnung
02	Abbruch der Übertragung
03	Datenübertragung unverschlüsselt
04	Datenübertragung verschlüsselt
05	Datenübertragung komprimiert
06	Datenübertragung unkomprimiert
07	Keine Daten vorhanden
Elektronische Unterschrift (Bereich 21-50)	
21	Unterschriftsprüfung
22	Zur EU gehörende Originaldatei noch nicht übertragen

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Textnummer	Text
23	Unterschrift(en) noch nicht uebertragen
24	Unterschrift(en) in Ordnung
25	Unterschrift(en) fehlerhaft
26	Teilnehmer hat mehrfach unterschrieben
27	Keine Unterschriftsberechtigung
28	Unterschrift ist falsch
29	Identische Unterschrift gefunden
30	Falsche Public Key-Version
31	Kein Public Key vorhanden
32	Public Key noch nicht freigegeben
33	Die erforderliche Anzahl von EUs ist nicht vorhanden
34	Angaben zur Originaldatei nicht bei allen EUs identisch
35	Datei nicht pruefbar. Auftrag komplett wiederholen !
36	Aufbau bzw. Groesse der EU-Datei falsch
37	EU-Berechtigung(en) nicht ausreichend
Dateibezogene Nachverarbeitung (Bereich 51-70)	
51	Fehler bei Dekomprimierung
52	Datei nicht lesbar
53	Fehler bei Entschluesselung
54	Datei ist in ihrem Aufbau fehlerhaft
Bankfachliche Prüfungen (Bereich 71-90)	
71	Keine Berechtigung für Konto
72	Vereinbarter Hoechstbetrag ueberschritten

Beispiele:

```
14.11.02 11:50:15      Datei zur Bank uebertragen
  Hostname      : HOSTFTAM
  Auftrag       : Beliebige Datei senden                      FTB AAI0
  Teilnehmer    : USER Teilnehmer User
  Ergebnis      : Uebertragung in Ordnung [01]
                  Datuebertragung unverschluesselt [03]
                  Datuebertragung unkomprimiert [06]
```

```
14.11.02 11:50:15      Datei zur Bank uebertragen
  Hostname      : HOSTFTAM
  Auftrag       : Beliebige Datei senden                      FTB AAJO
  Teilnehmer    : USER Teilnehmer User
  Ergebnis      : Uebertragung in Ordnung [01]
                  Datuebertragung verschluesselt [04]
                  Datuebertragung komprimiert [05]
```

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

```
14.11.02 11:51:55    Unterschriftspruefung [21]
  Hostname      : HOSTFTAM
  Auftrag       : Inlandszahlungsverkehrsdatei           IZV AAM0
  Teilnehmer    : USER Teilnehmer User
  Ergebnis      : Unterschrift(en) in Ordnung [24]
  Dateiname     : C:\DAT\IZV1.DTA
```

Displaydatei des Auftrags

```
14.11.02 11:51:55    Unterschriftspruefung [21]
  Hostname      : HOSTFTAM
  Auftrag       : Inlandszahlungsverkehrsdatei           IZV AAN0
  Teilnehmer    : USER Teilnehmer User
  Ergebnis      : Unterschrift(en) fehlerhaft [25]
  Dateiname     : C:\DAT\IZV1.DTA
```

Displaydatei des Auftrags

Die erforderliche Anzahl von EUs ist nicht vorhanden [33]

1.7.3 Liste der möglichen Meldungen inklusive Textnummer

Art der Aktion	Meldungs- bzw. Fehlermeldungstexte (deutsch)	Meldungs- bzw. Fehlermeldungstexte (englisch)
Übertragung	Datei zur Bank uebertragen Datei von Bank abgeholt Unterschrift zur Bank uebertragen	File submitted to the bank File collected from the bank Electronic signature submitted to the bank
Weiterverarbeitung	Unterschriftspruefung [21] Fehler bei Dekomprimierung [51] Fehler bei Entschluesselung [53] Anzeige Dateiinhalt	Signature verification [21] Decompression error [51] Decryption error [53] Display of the file content

Ergebnis der Aktion	Meldungs- bzw. Fehlermeldungstexte (deutsch)	Meldungs- bzw. Fehlermeldungstexte (englisch)
Übertragung	Uebertragung in Ordnung [01] Abbruch der Uebertragung [02] Datenuebertragung unverschluesselt [03] Datenuebertragung verschluesselt [04] Datenuebertragung komprimiert [05] Datenuebertragung unkomprimiert [06] Keine Daten vorhanden [07]	Transmission successful [01] Transmisson aborted [02] Data transfer not encrypted [03] Data transfer encrypted [04] Data transfer compressed [05] Data transfer not compressed [06] No data available [07]
Weiterverarbeitung	Zur EU gehoerige Originaldatei noch nicht uebertragen [22] Unterschrift(en) noch nicht uebertragen [23] Unterschrift(en) in Ordnung [24] Unterschrift(en) fehlerhaft [25] Fehler bei Dekomprimierung [51] Datei nicht lesbar [52] (<i>nur bei Aktion „Anzeige Dateiinhalt“</i>) Fehler bei Entschluesselung [53] Datei ist in ihrem Aufbau fehlerhaft [54] OK (<i>nur bei Aktion „Anzeige Dateiinhalt“</i>)	Corresponding original file still not sent [22] Electronic Signature(s) still not sent [23] Electronic signature(s) correct [24] Electronic signature(s) incorrect [25] Decompression error [51] File cannot be read [52] (<i>nur bei Aktion „Display file content“</i>) Decryption error [53] Incorrect file structure [54] OK (<i>nur bei Aktion „Display file content“</i>)

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Erläuternder Text bei Fehler in der EU-Prüfung	Meldungs- bzw. Fehlermeldungstexte (deutsch)	Meldungs- bzw. Fehlermeldungstexte (englisch)
Auf Teilnehmer bezogene Texte	Teilnehmer hat mehrfach unterschrieben [26] Vereinbarer Hoechstbetrag ueberschritten [72] Keine Unterschriftsberechtigung [27] Teilnehmer hat sich noch nicht initialisiert Teilnehmer noch nicht freigeschaltet Teilnehmer gesperrt Teilnehmereintrag nicht vorhanden Unterschrift ist falsch [28] Identische Unterschrift gefunden [29] Falsche Public Key-Version [30] ²⁴ Kein Public Key vorhanden [31] Public Key noch nicht freigegeben [32] Keine Berechtigung fuer Konto [71]	User signed multiple times [26] limit exceeded [72] No authorisation rights [27] User not yet initialized User not yet activated User is locked User not existent Electronic signature incorrect [28] Identical signature found [29] Public key version incorrect [30] ²⁴ Public key not existent [31] Public key not yet activated [32] No account authorisation [71]
Allgemeine Texte	Die erforderliche Anzahl von EUs ist nicht vorhanden [33] Angaben zur Originaldatei nicht bei allen EUs identisch [34] Datei nicht pruefbar. Auftrag komplett wiederholen ! [35] ²⁵ Aufbau bzw. Groesse der EU-Datei falsch [36] EU-Berechtigung(en) nicht ausreichend [37] EU-Version wird nicht mehr unterstuetzt [38]	Insufficient numbers of signatures [33] Data relating to original file not identical for all signatures [34] File not testable. Repeat complete order [35] ²⁵ Wrong structure or size of the signatures [36] Electronic signature(s) rights insufficient [37] EU-version is no longer valid [38]

²⁴ Diese Meldung wird dann protokolliert, wenn ein Kunde nach dem Umstieg von einer älteren Programmversion (altes EU-Format) auf eine neue Programmversion (neues EU-Format) Unterschriftsdateien an das Kreditinstitut schickt, ohne sich vorher neu initialisiert beziehungsweise eine Public Key-Änderung durchgeführt zu haben.

²⁵ Diese Meldung wird ausgegeben, wenn eine Betriebsstörung bei der Unterschriftsprüfung auftritt, z.B. zu wenig Speicherplatz

1.7.4 Dateianzeige auf Kunden- und Bankseite

Auftragsarten für Dateien im DTAUS-Format:

	Aus Feldnummer der DTAUS-Spezifikation
Überschrift	
Zahlungsart	A3
Bankleitzahl	A4
Kontonummer	A9
Auftraggeber	A6
Erstellungsdatum	A7
Anzahl der Zahlungssätze	E4
Summe der Beträge (EUR)	E8
Summe der Kontonummern	E6
Summe der Bankleitzahlen	E7
Ausführungstermin	A11b

Beispiel:

G U T S C H R I F T E N

Bankleitzahl : 30040000
Kontonummer : 0825112600
Auftraggeber : BANK-VERLAG
Erstellungsdatum : 10.05.00
Anzahl der Zahlungssätze : 1
Summe der Beträge (EUR) : 10.000,00
Summe der Kontonummern : 00000000001234567
Summe der Bankleitzahlen : 00000000007654321
Ausführungstermin : 10.05.2000

Abbildung 1.9 Auftragsarten für Dateien im DTAUS-Format

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Auftragsarten für Dateien im DTAZV-Format:

	Aus Feldnummer der DTAUS-Spezifikation
Überschrift	
Bankleitzahl	Q3
Kundennummer	Q4
Auftraggeberdaten	Q5
Erstellungsdatum	Q6
T-Satz-Informationen	
Auftragswährung	T13
Bankleitzahl	T3
Kontowährung	T4a
Kontonummer	T4b
Ausführungstermin	T5
Betrag	Summe der Felder T14a und T14b aller T-Sätze, bei denen die voranstehenden Felder T13, T3, T4a, T4b und T5 identisch belegt sind. Bei abweichender Belegung in derselben Datei werden diese T-Satz-Informationen entsprechend mehrfach angegeben.
Anzahl der Datensätze T	Kontrollsumme aus Feld Z4
Summe der Beträge	Kontrollsumme aus Feld Z3

Beispiel:

Bankleitzahl	: 30040000
Kundennummer	: 0000000001
Auftraggeberdaten	: KARL MUSTERMANN MUSTERSTR. 1 50825 KOELN
Erstellungsdatum	: 10.05.00

Auftragswährung	: ILS
Bankleitzahl	: 30040000
Kontowährung	: EUR
Kontonummer	: 1234567890
Ausführungstermin	: 10.05.00
Betrag	: 20.000,000

Anzahl der Datensätze T	: 000000000000001
Summe der Beträge	: 000000002000000

Abbildung 1.10 Auftragsarten für Dateien im DTAZV-Format

2 Standards für die Sicherheit

2.1 Festlegungen / Symmetrischer Algorithmus

Als Basisverschlüsselungsroutine wird der 2-Key-Triple-DES im CBC Modus gemäß I-SO 10116 (ANSI X3.106) verwendet (vgl. **Abbildung 2.1**).

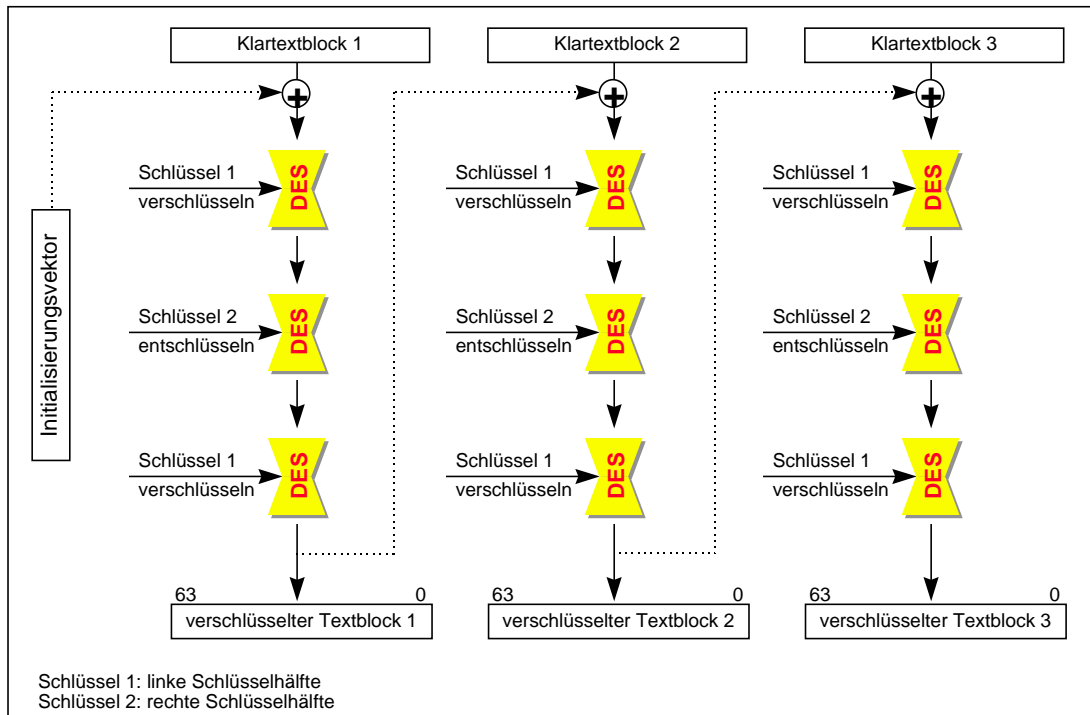


Abbildung 2.1: 2-Key-Triple-DES im CBC-Mode

Der Initial Chaining Value (ICV) wird auf X'00 00 00 00 00 00 00 00' festgelegt.

Die generierten Zufallszahlen, die als rechte und linke Schlüsselhälfte des 2-Key-Triple-DES verwendet werden, sind daraufhin zu überprüfen, dass es sich nicht um einen der in der nachfolgenden Tabelle aufgelisteten schwachen oder halbschwachen DES-Schlüssel handelt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Die schwachen Schlüssel des DES							
01	01	01	01	01	01	01	01
FE	FE	FE	FE	FE	FE	FE	FE
1F	1F	1F	1F	0E	0E	0E	0E
E0	E0	E0	E0	F1	F1	F1	F1

Die halbschwachen Schlüssel des DES							
01	FE	01	FE	01	FE	01	FE
FE	01	FE	01	FE	01	FE	01
1F	E0	1F	E0	0E	F1	0E	F1
E0	1F	E0	1F	F1	0E	F1	0E
01	E0	01	E0	01	F1	01	F1
E0	01	E0	01	F1	01	F1	01
1F	FE	1F	FE	0E	FE	0E	FE
FE	1F	FE	1F	FE	0E	FE	0E
01	1F	01	1F	01	0E	01	0E
1F	01	1F	01	0E	01	0E	01
E0	FE	E0	FE	F1	FE	F1	FE
FE	E0	FE	E0	FE	F1	FE	F1

2.2 Kryptographische Verfahren des deutschen Kreditgewerbes für die Elektronische Unterschrift im Rahmen der Kunde-Bank-Kommunikation

2.2.1 Allgemeine Anforderungen

Die zum Einsatz kommenden Sicherheitsverfahren müssen die Elektronische Unterschrift für die zu übertragenden Daten leisten. Hierbei ist folgendes Anforderungsprofil zu erfüllen:

- Die Unterschrift darf nur vom Unterzeichner geleistet werden können, so dass der Unterzeichner die Unterschrift nicht leugnen kann bzw. dass beweisbar ist, dass der Ursprung eines eventuellen Missbrauchs nur im Verantwortungsbereich des Unterzeichners liegen kann.
- Jeder mögliche Empfänger muss die Echtheit der Unterschrift prüfen können, wobei zusätzlich gewährleistet sein muss, dass diese Prüfung auch zu einem späteren Zeitpunkt (z. B. durch juristische Instanzen) möglich ist.
- Die Unterschrift muss in einem direkten Zusammenhang zu dem unterschriebenen Dateiinhalt stehen, so dass sie gleichzeitig den entsprechenden Dateiinhalt authentifiziert und so jeder mögliche Empfänger (insbesondere juristische Instanzen auch noch zu einem späteren Zeitpunkt) anhand der Unterschrift auch den Dateiinhalt verifizieren kann (Prüfung der Dateiintegrität).
- Die Unterschriftslösung muss auf jeden beliebigen Kontext anwendbar sein.
- Das Unterschriftenverfahren muss unter Performancegesichtspunkten auch auf weniger leistungsfähigen PC mit vertretbarer Rechenzeitintensität einsetzbar sein.
- Der Verwaltungsaufwand für die notwendige Aufbewahrung der zur Unterschriften-erzeugung und insbesondere -prüfung benötigten Daten (Kennungen) muss möglichst gering sein (einfaches Key Management).
- Die konkrete technische Lösung muss für die gängigen Betriebssysteme, die beim Unterzeichner bzw. Empfänger zum Einsatz kommen können, kompatibel einsetzbar sein.

Dieses Anforderungsprofil kann nur durch den Einsatz asymmetrischer kryptographischer Verfahren erfüllt werden.

Die Verwendung der Elektronischen Unterschrift wird für alle Datenübertragungen, die nicht der reinen Informationsbeschaffung dienen, dringend empfohlen, soweit in den besonderen Vereinbarungen für einzelne Verfahren nichts Abweichendes bestimmt ist.

Für jedes konkret zum Einsatz kommende Sicherheitsverfahren muss eine detaillierte Beschreibung der darin verwendeten mathematischen Verfahren sowie der verwendeten Datenstrukturen kostenlos offengelegt werden. Diese Beschreibung muss geeignet sein, ein funktional kompatibles Produkt durch beliebige Hersteller erstellen zu lassen. Außerdem muss ein positives Gutachten eines vom Kreditgewerbe bestimmten Gutachters zu dem jeweiligen Gesamtverfahren und speziell zu den darin zum Einsatz kommenden mathematischen Prozeduren vorgelegt werden.

2.2.2 Elektronische Unterschrift der Version A003

Für die Kreditinstitute besteht bis zum 31. Dezember 2004 die Verpflichtung zur Unterstützung der Version A003.

Unter Beachtung der oben genannten Festlegungen ist derzeit das nachfolgend beschriebene Verfahren für die Elektronische Unterschrift zugelassen.

2.2.2.1 Festlegungen

Asymmetrischer Algorithmus:

Derzeit wird das RSA-Verfahren eingesetzt. Dabei werden folgende Eigenschaften vorausgesetzt:

- Es wird ein konstanter öffentlicher Exponent e und ein für jeden Kunden individueller Modulus n für jedes eingesetzte RSA-Schlüsselsystem verwendet.
- Der Modulus n eines jeden RSA-Schlüsselsystems hat eine Länge von N Bit. Es sind keine führenden 0-Bits erlaubt, so dass auf jeden Fall gilt:

$$2^{N-1} \leq n < 2^N$$

- Der Zielwert für N ist 768, wobei eine aus der Suche nach starken Primzahlen resultierende Unterschreitung dieses Wertes um maximal 60 Bit zulässig ist.

n ist das Produkt zweier großer, zufällig ausgewählter Primzahlen p und q . Folgende Anforderungen werden an die Faktoren p und q gestellt:

p hat eine vorher festgelegte minimale Länge.
 $p - 1$ hat einen großen Primteiler r .
 $p + 1$ hat einen großen Primteiler s .
 $r - 1$ hat einen großen Primteiler.

- Die gleichen Forderungen werden an q gestellt.
- Die Längen von p und q sollen sich um höchstens 12 Bits unterscheiden.
- Der konstante öffentliche Exponent e wird für die Elektronische Unterschrift auf die 4. Fermatsche Primzahl festgelegt: $e = 2^{16} + 1$
- Bei der Wahl von p und q ist sicherzustellen, dass e kein Primfaktor von $p - 1$ oder $q - 1$ ist.

Hash-Algorithmus:

Derzeit wird der mit DFP bezeichnete Hash-Algorithmus benutzt. Basis für die Erzeugung dieses Hashwertes ist der Data Encryption Standard (DES, ANSI X3.92). M sei definiert als ein Datenblock beliebiger Länge. M wird durch Hinzufügen von Nullen auf der rechten Seite aufgefüllt, bis ein Vielfaches von 8 Bytes vorliegt (Paddingregel). Die 8 Bytes-Blöcke, deren Verkettung M ergibt, seien mit M_i ($1 \leq i \leq m$) bezeichnet. Die Funktion $DES(b,k)$ bezeichnet die DES-Verschlüsselung eines 8 Byteslangen Blockes b mit dem Schlüssel k . Mit \otimes wird nachfolgend eine Exklusiv-Oder-Funktion (XOR) bezeichnet. o_j sind Blöcke der Länge 8 Bytes mit

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

$$o_{-1} = 0;$$
$$o_0 = 0.$$

o_i berechnet sich für $i = 1$ bis m iterativ wie folgt:

$$o_i = M_i \otimes \text{DES}((M_i \otimes o_{i-1} \otimes o_{i-2} \otimes x), k)$$

Dabei ist (in hexadezimaler Darstellung)

$$x = X \text{ '01 23 45 67 89 AB CD EF'}$$

Die beiden letzten entstehenden Blöcke o_{m-1} und o_m seien mit $f_1(M,k)$ und $f_2(M,k)$ bezeichnet. Unter Verwendung von 2 verschiedenen Schlüsseln k_1 und k_2 werden zunächst berechnet:

- $c_1 = f_1(M, k_1)$
- $c_2 = f_2(M, k_1)$
- $c_3 = f_1(M, k_2)$
- $c_4 = f_2(M, k_2)$

Dabei sind (in hexadezimaler Darstellung)

- k_1 : X '902628CBEC461543'
- k_1 mit gerader Parität: X '902728CAED471442'
- k_1 mit ungerader Parität: X '912629CBEC461543'
- k_2 : X '2A41522F4446502A'
- k_2 mit gerader Parität: X '2B41532E4447502B'
- k_2 mit ungerader Parität: X '2A40522F4546512A'

Es wird eine Funktion G definiert, deren Ergebnis in Abhängigkeit von den jeweils 8 Bytes langen Variablen x , y und k ebenfalls 8 Bytes umfaßt. Dabei gilt:

$$G(x,y,k) = \text{DES}((x \otimes y), k) \otimes \text{DES}(x,k) \otimes \text{DES}(y,k) \otimes y$$

So werden gebildet:

$$\text{FP1} = G(G(c_1,c_2,k_1), G(c_3,c_4,k_1), k_1) \text{ und}$$
$$\text{FP2} = G(G(c_1,c_2,k_2), G(c_3,c_4,k_2), k_2).$$

Der 16 Bytes lange, hier mit DFP bezeichnete Fingerprint HASH ist dann die Verkettung dieser beiden Werte:

$$\text{DFP} = (\text{FP1}, \text{FP2})$$

2.2.2.2 Definitionen

Verwendete Symbole	Bedeutung / Wert
N	Modulus eines RSA-Schlüsselsystems
N	Länge von N in Anzahl der Bit
HASH	Fingerprint
THASH	Time stamped HASH
SIGNATUR	Signatur der nachricht
TVP	Zeitstempel (Time Variant Parameter)

2.2.2.3 Voraussetzungen

Für die Anwendung dieser Spezifikationen ist folgendes zu beachten:

- Für die Erzeugung der Elektronischen Unterschrift gilt die Paddingregel des jeweiligen Hash-Algorithmus (gemäß dem oben beschriebenen Hash-Algorithmus).
- Für die Erzeugung und Verifikation der Elektronischen Unterschrift verwendeten asymmetrischen Algorithmen sind oben unter „Asymmetrischer Algorithmus“ näher spezifiziert.
- Die im Folgenden verwandten Längenangaben setzen im Falle der Elektronischen Unterschrift stets eine Fingerprintlänge von 128 Bit voraus (siehe oben unter „Hash-Algorithmus“). Sollte zukünftig ein Hash-Algorithmus mit einer anderen Fingerprintlänge Verwendung finden, so sind die Längenangaben entsprechend anzupassen.
- Der aus dem zu unterschreibenden Text resultierende Fingerprint wird vor der Erzeugung der Elektronischen Unterschrift um den aktuellen Zeitstempel ergänzt. Dieses bewirkt, dass zwei Aufträge, die inhaltlich identisch sind, anhand der Zeitangabe unterschieden werden können. Damit ist der Empfänger in der Lage, zwei identische Aufträge vor der doppelten Einreichung eines Auftrages zu unterscheiden. Eine Manipulation der Zeitwerte während der Datenübertragung durch Außentäter kann durch die Einbeziehung der Werte in die Elektronische Unterschrift erkannt werden.
- Es wird generell ASCII als Zeichensatz verwendet.

2.2.2.4 Sicherung der Nachrichten

Vorgänge beim Sender: Erzeugung der elektronischen Unterschrift:

Es wird ein Fingerprint über die Originalnachricht berechnet. Dieser Fingerprint wird formatiert und das Ergebnis wird signiert.

Fingerprintberechnung:

Padding der Nachricht: Das Padding der Nachricht erfolgt nur temporär, um den Fin-

gerprint berechnen zu können. Die genaue Festlegung der Paddingregel wird in Abhängigkeit von den verwendeten Hash-Algorithmen (siehe oben „Hash-Algorithmus“) definiert.

Anwendung des Hashalgorithmus: Die Fingerprintberechnung erfolgt abhängig von dem verwendeten Hash-Algorithmus. Die Einzelheiten sind oben unter „Hash-Algorithmus“ zu ersehen.

Der Fingerprint hat eine Länge von 128 Bit und wird HASH genannt.

$\text{HASH} = Y_{128}, Y_{127}, \dots, Y_1$

Signatur des Fingerprints:

Der aus dem zu unterschreibenden Text resultierende Fingerprint wird vor der Erzeugung der Elektronischen Unterschrift um den aktuellen Zeitstempel (TVP) ergänzt, der den Zeitpunkt der geleisteten Unterschrift dokumentiert. Anhand des Zeitstempels können zwei inhaltlich identische Aufträge unterschieden werden. Damit ist der Empfänger in der Lage, zwei identische Aufträge von der doppelten Einreichung eines Auftrages zu unterscheiden. Eine Manipulation der Zeitwerte während der Datenfernübertragung durch Außentäter kann durch die Einbeziehung der Werte in die Elektronische Unterschrift erkannt werden.

Der Wert der Elektronischen Unterschrift hängt daher nicht ausschließlich vom Fingerprint der Nachricht ab, sondern auch von den angehängten Bits des Zeitstempels TVP.

Es wird eine Binärfolge THASH mit einer Länge von 256 Bit gebildet, die aus dem aktuellen Zeitstempel der Elektronischen Unterschrift und HASH besteht. Die niedrigwertigen 128 Bit von THASH enthalten den Zeitstempel TVP.

Der Zeitstempel hat das Format: **jjjjmmttX'20'hhppssX'20'**

Symbol	Bedeutung	Anzahl Bytes
jjjj	Jahr	4
mm	Monat	2
tt	Tag	2
X'20'	Leerzeichen	1
hh	Stunde	2
pp	Minute	2
ss	Sekunde	2
X'20'	Leerzeichen	1

Die Gesamtlänge von TVP beträgt 16 Bytes (128 Bit).

Es sei $\text{THASH} = (z_{256}, z_{255}, \dots, z_1)$ mit $\text{TVP} = (z_{128}, \dots, z_1)$ und $\text{HASH} = (z_{256}, \dots, z_{129})$.

Der geheime Schlüssel des RSA-Schlüsselsystems des Erzeugers der Elektronischen Unterschrift wird verwendet, um die Binärfolge THASH digital zu unterschreiben. Das Ergebnis heißt SIGNATUR.

Übermittlung:

An den Kommunikationspartner wird die Originalnachricht (ohne die Bits, die als Padding für die Fingerprintberechnung angefügt wurden) und ggf. die Elektronische Unterschrift des Senders (SIGNATUR) übertragen. Die genauen Formate sind dem Kapitel **2.2.2.5 Formate** zu entnehmen.

Vorgänge beim Empfänger: Verifikation der Elektronischen Unterschrift:

Erhalt des übermittelten Fingerprints:

Unter Verwendung des öffentlichen Schlüssels des Absenders wird aus der Bitfolge SIGNATUR die Bitfolge THASH in der Länge von 256 Bit ermittelt.

Die niedrigstwertigen 128 Bit von THASH werden als Zeitstempel interpretiert. Die Darstellung des Zeitstempels ist in Kapitel 0 definiert.

Die 128 höchstwertigen Bit von THASH bilden den Hashwert HASH, die der Empfänger als Begleitinformation für die Nachricht erhalten hat.

Berechnung des Fingerprints:

Die Berechnung des Fingerprint über die empfangene Nachricht erfolgt wie oben unter Fingerprintberechnung beschrieben.

Dieser Fingerprint heißt HASH´.

Vergleich der Fingerprints:

Abschließend wird verglichen, ob der empfangene Fingerprint HASH und der selbst berechnete Fingerprint HASH´ identisch sind.

Ist dies der Fall, so ist die empfangene Nachricht unversehrt und authentisch.

2.2.2.5 Formate

Für die Sicherheitservices auf dem Nachrichtenlevel werden die folgenden Grundanforderungen definiert:

- Authentikation
- Integrität
- Non-Repudiation

Authentikation, Integrität und Non-Repudiation werden mittels der Elektronischen Unterschrift verwirklicht. Die Elektronische Unterschrift bzw. bei Mehrfachunterschriften die Elektronischen Unterschriften werden in einer separaten „EU-Datei“ abgelegt.

Diese Datei enthält je Elektronischer Unterschrift folgenden 512 Bytes langen Datensatz:

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Inhalt	Länge in Bytes ²⁶	Erläuterung/Belegung
Versionsnummer	an 4	‚A003‘
Länge des Modulos	n 4	‚0768‘
Auftragsart	an 3	Auftragsartkürzel der Originaldatei gemäß den Standards für die Kommunikation Kapitel 1.4 Auftragsartenkennungen) z. B. ‚IZV‘
EU	binär 128	rechtsbündig, ‚0, ..., 0, SIGNATUR‘
UserID	an 8	z. B. ‚A2B2C2D2‘
Originaldatei	an 128	lokaler Dateiname der Originaldatei
Datum/Uhrzeit der Dateierstellung	an 16	jjjjmmttX'20'hhppssX'20' (siehe Signatur des Hashwertes in Kapitel 2.2.2.4 „Sicherung der Nachrichten“.
Datum/Uhrzeit der Unterschrift	an 16	jjjjmmttX'20'hhppssX'20' (siehe Signatur des Hashwertes in Kapitel 2.2.2.4 „Sicherung der Nachrichten“.
frei nutzbares Feld	binär 8	zur Zeit nicht genutzt; X'00‘
Reserve	binär 197	zur Zeit nicht genutzt; X'00‘

2.2.2.6 Beschreibung der Abläufe

Für das Verfahren der Elektronischen Unterschrift werden fünf Funktionen genutzt:

1. Keypaar-Generierung kundenseitig
2. Berechnung eines Fingerprints (Verhashung) bank- und kundenseitig
3. Signieren einer Datei (Erstellung einer EU) kundenseitig
4. Verifizieren der Signatur (Prüfung der EU) bankseitig
5. Änderung des Paßwortes für den Secret Key kundenseitig

Daten zur Identifikation und Legitimation des Kunden:

User-ID: Max. 8stellig alphanummerisch (wird vom jeweiligen Kreditinstitut frei vergeben)

Host-ID: Max. 8stellig alphanummerisch (wird vom jeweiligen Kreditinstitut vergeben)

Original-Passwort: Max. 8stellig alphanummerisch (erlaubt dem User den Zugang zu den verschiedenen Bankrechnern)

Passwort: Für jedes Kreditinstitut gelten unterschiedliche, aus dem Original-Passwort nach Voranstellen der max. achtstelligen Host-ID unter Verwendung der nachfolgend beschriebenen

²⁶ an = alphanummerisch; n = numerisch. Werte im ASCII-Format werden linksbündig eingestellt und rechts mit Blanks (X'20') aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

nen Hashfunktion abgeleitete Passworte (Fingerprint), die nicht auseinander berechenbar sind (Multi-Bankfähigkeit)

Verhasht werden nur die tatsächlich belegten Zeichen von Bankkennung und Passwort. Als Hash-Funktion wird das im Folgenden beschriebene Verfahren verwendet:

- Ein zu bearbeitender Datenstring x (ASCII-Darstellung) der Länge $2n$ wird in die Blöcke x_1, x_2, \dots, x_n von je 2 Bytes Länge unterteilt. Hat der Datenstring x eine ungerade Länge, so wird er mit einem Byte $X'FF'$ aufgefüllt (*Padding*).
- Jeder aus 4 Halbbytes bestehende Block x_i wird nun zu einem Block y_i doppelter Länge transformiert, indem jedem Halbbyte 4 binäre Einsen ($X'F'$) vorangestellt werden (*Zo-ning*).

Beispiel: $x_i = X'A18E' \rightarrow y_i = X'FAF1F8FE'$

- In einem iterativen Prozeß werden nun die einzelnen Blöcke verknüpft, indem Elemente h_i nach folgender Vorschrift gebildet werden (*Quadratur modulo m*):

$$h_1 = y_1^2 \bmod m$$

$$h_i = (h_{i-1} \otimes y_i)^2 \bmod m \quad (i = 2, \dots, n)$$

Als Modulo wird die Primzahl $X'B61A2CA7'$ (3.055.168.679) verwendet. Die Verknüpfung \otimes ist die bitweise Addition modulo 2 (XOR-Operation). Der Fingerprint $H(x)$ des Datenstrings x wird definiert als $H(x) = h_n$

- Dieser binäre Wert von 32 Bit Länge wird nun in eine aufbereitete lesbare Darstellung $H'(x)$ gebracht, indem das übliche Dump-Format gebildet wird, so dass als Ergebnis ein Datenstring von 8 Bytes Länge und dem Wertebereich 0, ... , 9, A, ... , F entsteht (*Presentation*).

Beispiel: $H(x) = X'259AEEB1' \rightarrow H'(x) = \text{„259AEEB1“ (ASCII)}$
 $= X'3235394145454231'$

Die beim Kreditinstitut über das Passwort zugänglichen Funktionen werden von den Kreditinstitut einzeln und User-spezifisch festgelegt. Das Kreditinstitut vergibt an einen neuen Benutzer zunächst ein Initial-Passwort mit dem Wert ‚start‘, das beim Kreditinstitut als entsprechender Hashwert ‚H‘ (Bankkennung und Passwort) gespeichert wird. Mit diesem Passwort ist nur die Berechtigung zu einer Passwort- und Public-Key-Änderung verbunden, die der User als erstes durchzuführen hat (Initialisierung). Bei der Initialisierung (Auftragsart ‚INI‘) wird

- das geänderte Passwort (Fingerprint) dem Bankrechner mitgeteilt,
- eine Datei übertragen, die bei Verfügbarkeit der EU den Public-Key des Users enthält und sonst aus einem Blank besteht,
- am PC ein Ausdruck (INI-Brief) mit folgenden Daten veranlaßt:
Benutzername / Datum / Uhrzeit / Empfänger-Bank / User-ID (externe) / Kunden-ID / EU-Versionsnummer / definierte Längenangabe des Public-Key / Public-Key / Längenangabe des Modulus / Modulus / DFP- Fingerprint über den Public-Key (Exponent und Modulo) in 4 Zeilen à 8 Zeichen (4 Bytes). (Beispiel für den Ausdruck des INI-Briefes am Ende dieses Kapitels 2.2.2.6). Das ausgedruckte Formular (INI-Brief) sendet der Benutzer unterschrieben an die Bank, die daraufhin erst die vereinbarten Transaktionen des Users freigibt. Bei vergessenen Passwörtern setzt sich der Kunde mit der zuständigen Bank in Verbindung, um erneut ein Initial-Passwort mit dem Wert „start“ zu erhalten.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Beispiel:

Im folgenden werden die Anwendungen dieser Funktionen anhand eines Beispiels beschrieben. Parameter für dieses Beispiel:

Parameter	Wert
Kunden-ID	'A1B1C1D1'
User-ID	'A2B2C2D2'
Exponent	0...010001 (Hex)
Lmodulo	'0768'
Modulo	0...bc7bdc...87 (Hex)
Auftragsart Originaldatei	'TST'
Auftragsart Unterschrift	'TST'
Auftragsnummern	'A000'
FTAM spezifische Parameter:	
Host-ID	'A3B3C3D3' (8-stellig, alphanummerisch)
Password	8-stellig, alphanummerisch

Key-Paar-Generierung und Verteilung des Public-Key

Kundenseite Rahmenprogramm:

- Erzeugung und Speicherung von Public-Key und Private Key
- Generierung der Public-Key-Datei:
(Werte im ASCII-Format werden linksbündig eingestellt und rechts mit Blanks X'20' aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt)

Inhalt	Anzahl Bytes	Belegung / Erläuterung
Versionsnummer	4 ASCII	'A003' Dieses Feld dient zur Kennzeichnung des verwendeten EU-Verfahrens (A003).
User-ID	8 ASCII	'A2B2C2D2' Dieses Feld enthält die institutsabhängige User-ID. Dies hat zur Folge, dass das Kundensystem vor dem Versenden der Public Key-Datei die entsprechende User-ID eintragen muss.
LExponent	4 ASCII	'0768', Länge des Exponenten, Wert in Bit
Exponent	128 binär	00...010001 (Hex) Dieses Feld kann Werte bis maximal 1024 Bit enthalten.
LModulo	4 ASCII	'0768', Länge von Modulo, Wert in Bit
Modulo	128 binär	0...bc7bdc...87 Dieses Feld kann Werte bis maximal 1024 Bit enthalten.
Reserve	236 ASCII	Auffüllen mit X'20'

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- Anlegen eines DFÜ-Auftrages des Typs INI (Initialisierung)
A3.A1B1C1D1.INI.DNNNN.A000
FTAM-spezifische Felder
- Übertragung der Public-Key-Datei
- Versand des INI-Briefes an die Bank

Bankseite Rahmenprogramm:

- Lesen Public-Key-Datei
- Eingabe des Fingerprints aus INI-Brief des Kunden (optional)
- Prüfung von Public-Key und Fingerprints über Kryptographie-Funktion (optional)
- Protokollierung des Public-Keys, des Hashwertes sowie des Prüfungsergebnisses (optional) Beispiel: PUB.PTK
- Im Erfolgsfall: Freischaltung des Teilnehmers, Speicherung des Keys
Neuer Status des Teilnehmers: ‚BEREIT‘
- Im Fehlerfall: Eingabewiederholung oder Löschung der Public-Key-Datei
- Protokollierung des Initialisierungsvorganges im Kundenprotokoll A1B1C1D1.PTK

Ablauf der Elektronischen Unterschrift

Kundenseite Rahmenprogramm:

- Bildung eines Fingerprints durch Verhashung der Originaldatei. Betriebssystem-abhängige Zeichen (*bei DOS CR, LF, CRLF und Control-Z*) gehen in die Bildung des Hashwertes nicht mit ein.
- Einfügen des Zeitstempels
- Bildung der Signatur des Fingerprints
- Erstellung der EU-Datei
- Generierung zweier DFÜ-Aufträge:
A3.A1B1C1D1.TST.ONNNN.A000 (für die Originaldatei)
A3.A1B1C1D1.TST.UNNNN.A000 (für die Unterschriftsdatei)
- Übertragung der Dateien innerhalb des bankseitig festgelegten Zeitfensters

Bankseite Rahmenprogramm:

- Falls O-Datei und zugehörige U-Datei vorhanden, erfolgt Prüfung (O-Datei:
O\TST\A1B1C1D1.A000, ggf. U-Datei: U\TST\A1B1C1D1.A000)

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Auf Bankseite wird ein zeitlicher Rahmen definiert, innerhalb dessen zusammengehörige Original- und Unterschriftsdateien eingetroffen sein müssen.

- Bildung des Fingerprints über die O-Datei durch Verhashung
- Erweitern des Fingerprints durch Datum/Uhrzeit der EU aus EU-Datei
- Verifizierung
- EU-Protokoll
(Informationen aus EU-Datei, Public Key, A-/E-Satz der O-Datei, Prüfungsergebnisse)
Beispiel: EU.PTK
- Erfolgsfall bei EU-Prüfung: Weiterleitung der O-Datei
- Fehlerfall bei EU-Prüfung : Ablehnung der O-Datei
- Archivierung nach den einschlägigen handels- und steuerrechtlichen Vorschriften unter Berücksichtigung der Grundsätze ordnungsgemäßer Speicherbuchführung
- Eintrag ins Kundenprotokoll (A1B1C1D1.PTK)

Abholen der Kundenprotokolle

- Einrichten eines DFÜ-Auftrags PTK zum Abholen des Kundenprotokolls
'A3.A1B1C1D1.PTK.DNNNN'/FTAM-spezifische Felder
Dort verzeichnet sind z. B.:
Annahme bzw. Ablehnung des Public Keys, Ergebnis jeder EU-Prüfung, Annahme einer Kundendatei, Ergebnisse aus der Weiterverarbeitung, usw.
- Übertragung des Kundenprotokolls
Beispiel: A1B1C1D1.PTK

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Beispiel für einen INI-Brief DFÜ (A003):

Benutzername system Kundensoftware-interner Name (Angabe freigestellt)

Datum TT.MM.JJJJ Datum der Erstellung des Initialisierungs- bzw. Public-Key-Änderungsauftrages

Uhrzeit HH:MM Uhrzeit der Erstellung des Initialisierungs- bzw. Public-Key-Änderungsauftrages

Empfänger DFÜ-Bank Hostname der Bank (max. 8 Stellen; wird von der jeweiligen Bank mitgeteilt)

User-ID xxxxxxxx (8 Bytes alphanummerisch; beginnend mit einem Alphazeichen; wird von der jeweiligen Bank mitgeteilt)

Kunden-ID yyyyyyyy (8 Bytes alphanummerisch; wird von der jeweiligen Bank mitgeteilt)

EU-Version A003

Öffentlicher Schlüssel (Public-Key) für die Elektronische Unterschrift:

Exponent 0768

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01
```

Modulo 0768

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 D7 4A 2F D5 6C 16 4E C3 2D 82 F3 02 31
CD FF FB 45 77 E4 7E E5 B2 CB 7B 9A 5F 75 7B 32
7C 16 E5 FB 16 41 0B 4A 39 OF 50 47 68 9C 9B 27
D2 A0 9C CA 23 A8 C3 1C AB A5 ED 72 75 9D 0A B8
9B 37 BA 00 CB 68 BB AC C8 D1 C8 D3 35 C8 BF 1F
A3 06 CF 24 5A DC EB 84 64 86 D0 97 8f E4 67 08
```

Hash 5D F9 15 F4
3D 78 69 D5
00 80 60 E5
9A 4C 87 5A

Ich bestätige hiermit den obigen öffentlichen Schlüssel für meine Elektronische Unterschrift.

Ort / Datum Firma / Name Unterschrift

2.2.2.7 Testdaten Elektronische Unterschrift²⁷

1. Testdatei: TEST.DAT – 24 Byte

2. Testdatei: 255.IZV – 640 Byte

Hashfunktion: AR/DFP

k1:	90	26	28	CB	EC	46	15	43
k2:	2A	41	52	2F	44	46	50	2A

Elektronische Unterschrift A003: Beispiel 1

768-Bit RSA-Schlüssel

Modulus:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
a9 75 25 51 ba d1 29 33 b8 c9 31 d8 29 7b 9b 56
64 ce d1 40 d9 3d 79 75 96 8f 53 3e 51 f2 d2 b7
88 1d 38 4d 42 20 37 71 26 2e 11 0c 68 f5 62 c6
58 f3 d3 4c 93 71 37 15 f2 52 e0 22 e7 fc c4 95
fb 68 d9 a7 fc ec 70 fc cb 5d f3 73 c4 be 6b f2
f6 2c 3c d7 b3 1e 70 b5 a1 a7 6e e1 f3 dd 03 c9
```

Öffentlicher Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 01 00 01
```

Geheimer Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
59 d7 8b f3 aa f2 5d bf d8 46 83 30 b1 bd a0 fb
ca f2 83 aa b8 02 89 b4 b8 20 40 e8 17 96 7f f5
62 2a eb c0 ba 40 4e 64 a4 f3 63 82 8a db 1e 84
5f cd 54 6d f0 a8 b5 e4 50 90 2d 9c 63 f6 4f fc
79 0e 63 18 2e 6c b5 83 c1 81 b9 3a 99 66 ee f1
91 05 01 df a5 e8 b6 84 52 f1 07 0d 55 be f1 21
```

Fingerprint (des öffentlichen Schlüssels):

```
3e df 78 c3 17 bc bf df c7 f8 d3 77 ca d3 69 1a
```

²⁷ Die Dateien TEST.DAT und 255.IZV können über DFUE@SIZ.de angefordert werden.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Elektronische Unterschrift der 1. Testdatei:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
24 aa d3 51 12 1a 41 a1 1a 7d af 6a 0a 99 38 4b
f8 05 27 9d 7e 4f 1a 2f 82 3d 1c e6 7a 2e 8a a7
79 b0 48 5f 0c b3 58 1b a6 5d 22 7a 09 42 ae 2e
34 5a ac 14 c4 f9 16 7e 7b 7c 4b 94 5e 1e 09 02
26 39 48 e8 79 fb bd 52 59 43 4e ee 54 07 f8 eb
c3 a5 db 63 8d 3d 61 85 f1 21 e4 b3 12 3f c2 c9

Unterschriebener DSI:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5c 2d eb 15 8b 52 4d 2a bb ab a0 26 3b 4e 64 ea
32 30 30 34 30 32 30 31 20 31 30 30 33 31 34 20

Unterschriebener Fingerprint:

5c 2d eb 15 8b 52 4d 2a bb ab a0 26 3b 4e 64 ea

Timestamp:

20040201 100314

Elektronische Unterschrift der 2. Testdatei:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11 bc c8 85 92 54 11 05 a9 97 1e 74 1a 7e 68 0f
b3 22 c3 e0 44 7e 55 d4 f7 b7 30 91 e5 73 13 40
fa c4 a3 a2 54 da f7 69 aa b2 12 6f 19 9e c6 5b
74 1b 11 62 76 07 e0 db 4b d9 79 6d 55 bf a3 88
27 db 2b 96 2e 72 8e 7d 2e d2 a3 34 7a 73 ea b3
a8 c0 18 c7 32 58 db ea f2 78 66 b9 e3 0f 55 7f

Unterschriebener DSI:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4d 02 c3 09 5f e1 32 88 e8 08 4b ee 52 2c 1e a2
32 30 30 34 30 32 30 31 20 31 30 30 33 31 34 20

Unterschriebener Fingerprint:

4d 02 c3 09 5f e1 32 88 e8 08 4b ee 52 2c 1e a2

Timestamp:

20040201 100314

Elektronische Unterschrift A003: Beispiel 2

768-Bit RSA-Schlüssel

Modulus:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
bb 5f cc 01 3f e3 bf 27 09 9e 90 01 2a 0c 4e 40
e9 02 49 bd f0 5c 65 04 2d df ed e1 19 06 74 a3
1e bd 86 3b 39 61 a6 44 3d 3e fb 0f 3c ae 9d 5d
ec 6c b4 83 43 9d d6 58 a9 19 ad e9 6d 49 f2 51
44 76 34 e8 e5 4b 2d 22 04 7f d5 49 d3 ed 29 9b
66 25 41 26 1f ab 76 31 f3 82 88 a9 ae 79 cc 03
```

Öffentlicher Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 01
```

Geheimer Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
b4 7f 87 e9 fc 3d cc 3a 32 ce 08 32 d6 ea 9b c7
73 ae bc 92 b8 24 89 3b 09 66 19 a5 29 92 4a 71
88 7f 51 fb 63 3f 6a 07 7a 68 5d 39 44 5f 81 3e
ff 0e d9 db fc 10 66 72 56 92 b0 ff ed 0f 52 8e
80 0d bd 55 b7 48 bc 9c df 2b ca 0f 42 da a3 6d
4e b5 49 2b 45 20 38 64 b1 00 4c 5c ad e9 df a1
```

Fingerprint (des öffentlichen Schlüssels):

```
f4 94 e9 1a 32 2c 05 e4 97 df a9 39 b0 e1 d4 36
```

Elektronische Unterschrift der 1. Testdatei:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
b3 9a 30 ff 83 a8 c8 55 81 b5 4e a1 0e c4 8d 5c
9d 27 43 12 7a 33 6a 4f ca 22 a6 56 b4 75 49 b9
82 82 98 fd ac 40 fe 5b b8 ba d3 d9 89 0e ea a5
87 b3 e8 f8 33 5e 78 3a c3 7b 97 43 e9 09 80 94
fd 51 02 6c f3 cc da 16 f4 df 7d 48 e4 94 42 16
b7 65 35 5c 56 57 93 d2 ae 66 83 3c 87 50 d7 60
```

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Unterschriebener DSI:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5c 2d eb 15 8b 52 4d 2a bb ab a0 26 3b 4e 64 ea
32 30 30 34 30 32 30 31 20 31 36 32 35 35 34 20

Unterschriebener Fingerprint:

5c 2d eb 15 8b 52 4d 2a bb ab a0 26 3b 4e 64 ea

Timestamp:

20040201 162554

Elektronische Unterschrift der 2. Testdatei:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5a 89 d5 71 26 d9 86 66 32 b6 bc a7 61 4a 0b 84
e0 5b 68 84 d2 a2 22 e6 25 93 4d 5e 8f ff 68 99
25 f8 e0 81 55 eb 29 1b 48 36 a9 b5 1d 0a 1d 8c
74 44 58 b3 ce 31 59 2c 91 55 42 f9 f5 e6 4b 35
6d 0a 0f 26 a2 4d ba d7 86 e8 ee ed 7b 27 4f 86
8d 5e f2 40 5c cf e7 b2 c0 78 72 5b 76 7f 3a 36

Unterschriebener DSI:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4d 02 c3 09 5f e1 32 88 e8 08 4b ee 52 2c 1e a2
32 30 30 34 30 32 30 31 20 31 36 32 35 35 34 20

Unterschriebener Fingerprint:

4d 02 c3 09 5f e1 32 88 e8 08 4b ee 52 2c 1e a2

Timestamp:

20040201 162554

2.2.3 Elektronische Unterschrift der Version A004

Unter Beachtung der in Kapitel [2.2.1](#) „Allgemeine Anforderungen“ genannten Anforderungen ist das nachfolgend beschriebene Verfahren für die Elektronische Unterschrift ab dem 1. April 2002 bankseitig verpflichtend zu unterstützen.

2.2.3.1 Einleitung

Die von der ZKA-Chipkarte bereitgestellten asymmetrischen kryptographischen Algorithmen basieren auf dem RSA-Algorithmus mit ungeraden öffentlichen Exponenten ([\[RSA\]](#)).

Im nachfolgenden Kapitel [2.2.3.2](#) „RSA-Schlüsselkomponenten“ werden das Konstruktionsprinzip und die **Schlüsselkomponenten** der öffentlichen und privaten RSA-Schlüssel gemäß Anhang B von [\[EMV B2\]](#), Anhang A von [\[ISO DS2\]](#) und [\[PKCS1\]](#) für ungerade öffentliche Exponenten erläutert.

Ein **Signatur-Algorithmus** besteht aus einem Algorithmus zur Signaturerzeugung und einem hierzu inversen Algorithmus zur Klartextrückgewinnung. Der als Standard von der ZKA-Chipkarte unterstützte Signatur-Algorithmus wird in Kapitel [2.2.3.3](#) unter „Signaturerzeugung“ beschrieben.

Der beschriebene Signatur-Algorithmus auf Basis des RSA-Algorithmus wird durch die ZKA-Chipkarte nur im Rahmen von Signaturverfahren eingesetzt. Ein **Signaturverfahren** legt fest, in welcher Weise eine Nachricht M zu einer Bytefolge aufzubereiten ist, die dann in die Signaturerzeugung eines Signatur-Algorithmus eingeht. Die durch ein Signaturverfahren erzeugte Bytefolge wird als **Digital Signature Input (DSI)** bezeichnet.

In Kapitel [2.2.3.4](#) „Signaturverfahren gemäß DIN-Spezifikation“ des vorliegenden Dokuments wird das Signaturverfahren beschrieben, das durch die für die ZKA-Chipkarte spezifizierte Signatur-Anwendung zur Erzeugung digitaler Signaturen unterstützt wird. Dieses Signaturverfahren entspricht der DIN-Spezifikation [\[DINSIG\]](#) einer Signatur-Anwendung/Funktion nach SigG und SigV.

2.2.3.2 RSA-Schlüsselkomponenten

Ein RSA-Schlüsselpaar besteht aus

- einem öffentlichen Schlüssel P_K und
- einem privaten Schlüssel S_K

Öffentlicher und privater Schlüssel bestehen aus **Schlüsselkomponenten**. RSA-Schlüssel werden auch als **asymmetrische Schlüssel** bezeichnet.

Zur Erzeugung eines RSA-Schlüsselpaares mit einem ungeraden **öffentlichen Exponenten** e werden zwei verschiedene **Primzahlen p und q (Primfaktoren)** verwendet, für die e teilerfremd zu $(p-1)$ und $(q-1)$ ist.

Der zugehörige **private Exponent d** ist dann bestimmt durch

$$e \cdot d \equiv 1 \pmod{\text{kgV}(p-1, q-1)}.$$

Die Primzahlen p und q sowie der private Exponent d müssen geheimgehalten werden.

Das Produkt der Primzahlen $n = p \cdot q$ wird als **Modulus** bezeichnet.

Der **öffentliche Schlüssel** P_K des RSA-Schlüsselpaars besteht aus den Komponenten

- Modulus n und
- öffentlicher Exponent e .

Der **private Schlüssel** S_K des RSA-Schlüsselpaars kann auf zwei Arten durch Komponenten dargestellt werden (vgl. [\[PKCS1\]](#)):

1. Darstellung von S_K durch die Komponenten:

- Modulus n und
- privater Exponent d ,

2. Darstellung von S_K durch die Komponenten:

- Primfaktor p ,
- Primfaktor q ,
- $d_p = d \bmod (p-1)$,
- $d_q = d \bmod (q-1)$ und
- $q_{inv} = q^{-1} \bmod p$.

Nur die Komponente d der ersten Darstellung muss geheimgehalten werden. Die Komponenten der 2. Darstellung werden als **Chinese Remainder Theorem-Parameter (CRT-Parameter)** bezeichnet. Die CRT-Parameter müssen sämtlich geheimgehalten werden.

Die ZKA-Chipkarte unterstützt den RSA-Algorithmus mit beliebigen ungeraden öffentlichen Exponenten, die die Länge des Modulus nicht überschreiten. In der Regel wird der ungerade öffentliche Exponent $F_4 = 2^{16} + 1$ verwendet. In dem vorliegenden Dokument wird die folgende Notation verwendet:

k bezeichnet die Bit-Länge des Modulus n eines RSA-Schlüsselpaars.

k ist durch die Gleichung $2^{k-1} \leq n < 2^k$ eindeutig definiert.

n lässt sich darstellen als Folge von Bit

$$n = b_k b_{k-1} \dots b_1, \text{ wobei } b_k \neq 0 \text{ ist.}$$

Der Wert von n als ganzer Zahl wird dadurch bestimmt, dass das erste, linke Bit b_k das höchstwertige Bit und das letzte, rechte Bit b_1 das niedrigstwertige Bit in der Binärdarstellung von n ist.

Zu k existieren eindeutige Zahlen $N \geq 1$ und $8 \geq r \geq 1$ mit $k = 8 \cdot (N-1) + r$. Dann lässt sich n auch als Folge von Bit schreiben als

$$n = b_r b_{r-1} \dots b_1 b_{8 \cdot (N-1)} \dots b_{8 \cdot (N-2) + 1} \dots b_8 \dots b_1.$$

Wenn $r = 8$ ist, lässt sich n direkt als Folge von N Bytes schreiben:

$$n = B_N B_{N-1} \dots B_1, \text{ wobei } B_N \neq '00' \text{ und } B_N \geq '80' \text{ ist.}$$

Falls $r < 8$ ist, stellt man $8-r$ binäre 0 der Bitfolge $b_r b_{r-1} \dots b_1 b_{8 \cdot (N-1)} \dots b_{8 \cdot (N-2) + 1} \dots b_8 \dots b_1$ voran:

$$n = 0 \dots 0 b_r b_{r-1} \dots b_1 b_{8 \cdot (N-1)} \dots b_{8 \cdot (N-2) + 1} \dots b_8 \dots b_1.$$

und erhält wiederum eine Bytefolge

$$n = B_N B_{N-1} \dots B_1, \text{ wobei } B_N \neq '00' \text{ und } B_N < '80' \text{ ist.}$$

Der ganzzahlige Wert von n wird durch führende 0 in der Binärdarstellung nicht geändert, so dass die Darstellung von n als Folge von N Bytes den durch eine Folge von k Bit dargestellten Zahlwert unverändert lässt.

N ist die Byte-Länge von n.

Die ZKA-Chipkarte verwendet nur Moduli, die mindestens 128 Bytes lang sind. Aus technischen Gründen können durch die ZKA-Chipkarte

- für die Berechnung einer Signatur nur Moduli mit einer maximalen Länge von 256 Bytes und
- für die Prüfung einer Signatur nur Moduli mit einer maximalen Länge von 252 Bytes

verarbeitet werden.

2.2.3.3 Signatur-Algorithmus

Signaturerzeugung

Es sei S_K ein privater RSA-Schlüssel bestehend aus dem Modulus n und dem privaten Exponenten d oder bestehend aus den CRT-Parametern. Der zugehörige öffentliche RSA-Schlüssel P_K bestehe aus dem Modulus n und dem öffentlichen Exponenten e .

Dann können mit S_K binär kodierte Bytefolgen x signiert werden, deren sich aus der Binärdarstellung von x ergebender ganzzahliger Wert zwischen 0 und $n-1$ liegt. x lässt sich somit darstellen als Bytefolge mit einer Länge von N Bytes und als Bitfolge mit einer Länge von k Bit. Das k -te Bit in der darstellenden Byte- oder Bitfolge kann, muss aber nicht, den Wert 1 haben. Sofern vorhanden haben die Bit $b_{8 \cdot N} \dots b_{k+1}$ in der darstellenden Bytefolge den Wert 0.

Für die Signaturerzeugung mit dem aus n und d bestehenden privaten Schlüssel wird die folgende Notation verwendet:

$$\text{sign}(S_K)[x] = x^d \bmod n$$

Falls der private Schlüssel S_K aus den CRT-Parametern besteht, berechnet sich $\text{sign}(S_K)[x] = x^d \bmod n$ wie folgt:

$$\text{sign}(S_K)[x] = s_2 + h \cdot q$$

wobei S_2 und h wie folgt berechnet werden:

$$\begin{aligned} s_1 &= x^{dp} \bmod p \\ s_2 &= x^{dq} \bmod q \\ h &= q \text{Inv}^*(s_1 - s_2) \bmod p. \end{aligned}$$

Hierbei werden die Potenzierungen $x^d \bmod n$, $x^{dp} \bmod p$, $x^{dq} \bmod q$ mit der ganzen Zahl ausgeführt, deren Wert sich aus der Binärdarstellung von x ergibt.

Das Ergebnis der Signaturerzeugung ist wiederum eine Bytefolge s , die sich als Binärdarstellung des ganzzahligen Wertes der Potenz $x^d \bmod n$ bzw. von $s_2 + h \cdot q$ ergibt. Dieser ganzzahlige Wert liegt wieder zwischen 0 und $n-1$. s lässt sich somit darstellen als Bytefolge mit einer Länge von N Bytes und als Bitfolge mit einer Länge von k Bit. Das k -te Bit in der darstellenden Byte- oder Bitfolge kann, muss aber nicht, den Wert 1 haben. Sofern vorhanden haben die Bit $b_{8 \cdot N} \dots b_{k+1}$ in der darstellenden Bytefolge den Wert 0.

Klartextrückgewinnung

Es sei P_K ein öffentlicher RSA-Schlüssel bestehend aus dem Modulus n und dem öffentlichen Exponenten e .

Dann kann mit P_K aus einer binär kodierten Bytefolge s Klartext zurückgewonnen werden, wenn sich der sich aus der Binärdarstellung von s ergebende ganzzahlige Wert zwischen 0 und $n-1$ liegt. s lässt sich somit darstellen als Bytefolge mit einer Länge von N Bytes und als Bitfolge mit einer Länge von k Bit. Das k -te Bit in der darstellenden Byte- oder Bitfolge kann, muss aber nicht, den Wert 1 haben. Sofern vorhanden haben die Bit $b_{8 \cdot N} \dots b_{k+1}$ in der darstellenden Bytefolge den Wert 0.

Für die Klartextrückgewinnung wird die folgende Notation verwendet:

$$\text{recover}(P_K)[s] = s^e \bmod n$$

Hierbei wird die Potenzierung $s^e \bmod n$ mit der ganzen Zahl ausgeführt, deren Wert sich aus der Binärdarstellung von s ergibt.

Das Ergebnis der Klartextrückgewinnung ist eine ganze Zahl, deren Wert zwischen 0 und $n-1$ liegt. Es lässt sich somit darstellen als Bytefolge mit einer Länge von N Bytes und als Bitfolge mit einer Länge von k Bit. Das k -te Bit in der darstellenden Byte- oder Bitfolge kann, muss aber nicht, den Wert 1 haben. Sofern vorhanden haben die Bit $b_{8 \cdot N} \dots b_{k+1}$ in der darstellenden Bytefolge den Wert 0.

Für ein RSA-Schlüsselpaar P_K und S_K gilt:

$$\text{recover}(P_K)[\text{sign}(S_K)[x]] = x$$

2.2.3.4 Signaturverfahren gemäß DIN-Spezifikation

Im Folgenden werden die in ein Signaturverfahren eingehenden Nachrichten M als Bitfolge der Bit-Länge m aufgefasst. Eine Nachricht lässt sich demnach schreiben als Folge von Bit b_i

$$M = b_m b_{m-1} \dots b_1$$

Wenn M als Binärzahl aufgefasst wird, ist das erste, linke Bit b_m das höchstwertige Bit und das letzte, rechte Bit b_1 das niedrigstwertige Bit. Das oder die höchstwertigen Bit einer Nachricht können den Wert 0 haben.

In der Regel ist m ein Vielfaches von 8, so dass sich M auch als Folge von Bytes darstellen lässt. Verfahren zur Kodierung von Nachrichten als Bit- oder Bytefolgen sind nicht Teil der durch die ZKA-Chipkarte unterstützten Signaturverfahren.

Falls ein Teil der zu signierenden Nachricht M als Bytefolge in dem DSI enthalten ist, kann dieser Teil durch die Klartextrückgewinnung des Signatur-Algorithmus aus der Signatur zurückgewonnen werden. In diesem Fall handelt es sich um ein Signaturverfahren **mit Nachrichtenrückgewinnung**. Der **zurückgewinnbare Teil** der Nachricht M wird als M_r bezeichnet. Falls die gesamte signierte Nachricht aus der Signatur zurückgewonnen werden kann, handelt es sich um **vollständige Nachrichtenrückgewinnung** ($M = M_r$), andernfalls um partielle Nachrichtenrückgewinnung. Der **nicht zurückgewinnbare Teil** der Nachricht M wird dann als M_n bezeichnet.

Falls der DSI keinen Teil der Nachricht als Bytefolge enthält, handelt es sich um ein Signaturverfahren **ohne Nachrichtenrückgewinnung** ($M = M_n$).

Zur Prüfung einer mit einem Signaturverfahren erzeugten digitalen Signatur wird außer der Signatur der nicht zurückgewinnbare Teil der signierten Nachricht benötigt.

Das im Folgenden beschriebene Signaturverfahren ist in [\[DINSIG\]](#) spezifiziert. Es ist ein Signaturverfahren ohne Nachrichtenrückgewinnung, das auf [\[ISO DS2\]](#) basiert. Es verwendet zur Erzeugung des DSI

- einen Hash-Algorithmus und
- einen Format-Mechanismus.

Ein Hash-Algorithmus bildet Bitfolgen beliebiger Länge (**Eingabe-Bitfolgen**) auf Bytefolgen einer festen, durch den Hash-Algorithmus festgelegten Länge ab. Das Ergebnis der Anwendung eines Hash-Algorithmus auf eine Bitfolge wird als **Hashwert** bezeichnet.

Das durch die für die ZKA-Chipkarte spezifizierete Signatur-Anwendung zur Erzeugung digitaler Signaturen unterstützte Signaturverfahren verwendet den Hash-Algorithmus **RIPEMD-160**.

Der Hash-Algorithmus RIPEMD-160 ist in [\[RIPEMD\]](#) und [\[ISO HF3\]](#) spezifiziert. RIPEMD-160 bildet Eingabe-Bitfolgen beliebiger Länge auf einen als Bytefolge dargestellten Hashwert von 20 Bytes Länge ab. Teil des Hash-Algorithmus ist das Padding von Eingabe-Bitfolgen auf ein Vielfaches von 64 Bytes. Das Padding erfolgt auch dann, wenn die Eingabe-Bitfolge bereits eine Länge hat, die ein Vielfaches von 64 Bytes ist.

RIPEMD-160 verarbeitet die Eingabe-Bitfolgen in Blöcken von 64 Bytes Länge.

Der Hashwert einer Eingabe-Bitfolge x unter dem Hash-Algorithmus RIPEMD-160 wird wie folgt bezeichnet:

$$\text{RIPEMD}(x).$$

Im Folgenden wird der Format-Mechanismus des Signaturverfahrens gemäß Spezifikation [DINSIG] beschrieben. Die verwendeten Abkürzungen sind in Kapitel 2.2.3.2 „RSA-Schlüsselkomponenten“ definiert.

Berechnen einer Signatur

Die folgenden Schritte werden gemäß Spezifikation [DINSIG] und [ISO DS2] zum Berechnen einer Signatur zu der Nachricht M der Bit-Länge m ausgeführt.

- Der Hashwert RIPEMD(M) mit einer Länge von 160 Bit wird berechnet. Hierbei gehen betriebssystemabhängige Zeichen (*bei Windows CR, LF, CRLF und Control-Z*) in die Bildung des Hashwertes nicht ein.
- Der DSI wird erzeugt. Der DSI ist eine Folge von k Bit, der wie folgt aufgebaut ist:

Bezeichnung	Bit-Länge	Wert
Header	2	0 1
More-Data-Bit	1	1, da , M = M _n ist
Paddingfeld	k-235	k-236 Bit 0, gefolgt von einem Bit 1 (Grenzbit)
Datenfeld	64	Zufallszahl: Die Zufallszahl muss bei jeder Signaturberechnung dynamisch erzeugt und in den DSI eingestellt werden.
Hashwert	160	RIPEMD(M)
Trailer	8	'BC'

Die Nachricht M besteht komplett aus dem nicht zurückgewinnbaren Teil M_n. Die ersten vier Bit des DSI können nur den Wert ,6' annehmen, da k-236 > 0 ist.

- Aus dem DSI wird mit dem Algorithmus zur Signaturerzeugung gemäß Kapitel 2.2.3.3 „Signatur-Algorithmus“ eine Signatur berechnet.

Hierbei ist zu beachten, dass der DSI als Folge von k Bit darstellbar ist, wobei das erste (höchstwertige) Bit den Wert 0 hat. Der sich aus der Binärdarstellung ergebende ganzzahlige Wert des DSI ist damit kleiner als 2^{k-1} und damit kleiner als der Wert des Modulus n.

Ferner lässt sich der DSI als Bytefolge darstellen, ggf. indem maximal 7 Bit mit dem Wert 0 dem ersten Bit der Bitfolge vorangestellt werden. Diese Bytefolge hat den selben ganzzahligen Wert wie die den DSI darstellende Bitfolge.

Die Signatur ist als Bytefolge darstellbar, deren Byte-Länge höchstens N ist. In der Darstellung des Modulus n als Bytefolge hat das Bit b_k den Wert 1 und die Bit $b_{8*N} b_{8*N-1} \dots b_{k+1}$ haben, sofern vorhanden, den Wert 0. In der Darstellung der Signatur als Bytefolge haben die Bit $b_{8*N} b_{8*N-1} \dots b_{k+1}$ ebenfalls den Wert 0.

Prüfen einer Signatur

Die folgenden Schritte werden gemäß Spezifikationen [DINSIG] und [ISO DS2] zum Prüfen einer Signatur ausgeführt. Hierzu müssen die zu prüfende Signatur s und die Nachricht M' der Bit-Länge m' vorliegen.

- Die Signatur muss als Bytefolge darstellbar sein, deren Byte-Länge höchstens N ist. In der Darstellung von s als Bytefolge müssen die Bit $b_{8 \cdot N} b_{8 \cdot N - 1} \dots b_{k+1}$, sofern vorhanden den Wert 0 haben. Ist das nicht der Fall, wird die Signatur abgewiesen. Der sich aus der Binärdarstellung von s ergebende ganzzahlige Wert muss zwischen 0 und $n-1$ liegen. Ist das nicht der Fall, wird die Signatur abgewiesen.

- Auf die Signatur wird der Algorithmus zur Klartextrückgewinnung gemäß Kapitel 2.2.3.3 „Signatur-Algorithmus“ angewandt. Das Ergebnis ist eine Bitfolge $b_k \dots b_1$, die als DSI' bezeichnet wird. DSI' muss den folgenden Anforderungen genügen:

Das niedrigstwertige Byte bestehend aus $b_8 \dots b_1$ hat den Wert 'BC'.

Das Bit b_{k-1} hat den Wert 1 und alle höherwertigen Bit haben den Wert 0.

Das Bit b_{k-2} („More Data“-Bit) hat den Wert 1.

Das Paddingfeld besteht aus $(k - 236)$ Nullen und einer 1 (Grenzbit).

Sind die Anforderungen nicht erfüllt, wird die Signatur abgewiesen.

- Aus dem DSI' wird ein Hashwert' entnommen. Der Hashwert' besteht aus den 160 Bit, die dem Trailer 'BC' vorausgehen.
- Der Hashwert RIPEMD(M') wird berechnet und mit dem Hashwert' verglichen. Falls die Werte gleich sind, war die Prüfung der Signatur erfolgreich. Andernfalls wird die Signatur abgelehnt.

Notation

Für die Berechnung einer Signatur zu einer Nachricht M mit dem Signaturverfahren gemäß [DINSIG], dem Signatur-Algorithmus RSA und dem privaten RSA-Schlüssel S_K wird die folgende Notation verwendet:

$$\text{sign}_{\text{DINSIG}}(S_K)[M].$$

Für die Prüfung einer Signatur s zu einer Nachricht M mit dem Signaturverfahren gemäß [DINSIG], dem Signatur-Algorithmus RSA und dem öffentlichen RSA-Schlüssel P_K wird die folgende Notation verwendet:

$$\text{verify}_{\text{DINSIG}}(P_K)[s, M].$$

2.2.3.5 Referenzen

- [DINSIG] DIN-Spezifikation der Schnittstelle zu Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV, DIN NI-17.4, Version 1.0, 15.12.1998
- [EMV B2] EMV2000, Integrated Circuit Card Specification for Payment Systems, Book 2, Security and Key Management, Version 4.0, EMVCo, December 2000
- [ISO DS2] ISO 9796 - 2, Information technology - Security techniques - Digital signature scheme giving message recovery, Part 2: Mechanisms using a hash function, 1997
- [ISO HF3] ISO 10118 - 3, Information technology - Security techniques - Hash-functions, Part 3: Dedicated hash functions, 1998
- [PKCS1] PKCS #1: RSA Cryptography Standard, Version 2.0, 1.10.1998
- [RIPEMD] H. Dobbertin, A. Bosselaers, B. Preneel, RIPEMD-160: A strengthened version of RIPEMD, 1996
- [RSA] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, vol. 21, n. 2, 1978, 120-126

2.2.3.6 Signaturformat A004

Signaturformat

Die Version A004 der Elektronischen Unterschrift basiert auf RSA-Signaturen, die mit Schlüsseln generiert werden, deren Moduli eine Länge von 1024 Bit haben. Die Signatur A004 unterstützt den RSA-Algorithmus mit beliebigen ungeraden öffentlichen Exponenten. In der Regel wird der ungerade öffentliche Exponent $F_4 = 2^{16} + 1$ verwendet. Als Paddingformat wird das in der „DIN-Spezifikation der Schnittstelle zu Chipkarten mit digitaler Signatur-Anwendung/Funktion nach SigG und SigV“ beschriebene Paddingformat „ISO 9796 Part 2 mit Zufallszahl“ verwendet.

Ermittlung des Hashwertes über die zu unterschreibende Datei

Das in der Version A004 der Elektronischen Unterschrift angewandte Signaturverfahren verwendet den Hash-Algorithmus RIPEMD-160.

RIPEMD-160 bildet Eingabe-Bitfolgen beliebiger Länge auf einen als Bytefolge dargestellten Hashwert von 20 Bytes Länge ab. Teil des Hash-Algorithmus ist das Padding von Eingabe-Bitfolgen auf ein Vielfaches von 64 Bytes. Das Padding erfolgt auch dann, wenn die Eingabe-Bitfolge bereits eine Länge hat, die ein Vielfaches von 64 Bytes ist. RIPEMD-160 verarbeitet die Eingabe-Bitfolgen in Blöcken von 64 Bytes Länge. Das Padding der Nachricht auf die entsprechende Blockgröße wird in der Beschreibung des Hashverfahrens spezifiziert. Der zu verwendende Initialisierungsvektor ist ebenfalls in der Beschreibung des Hashverfahrens festgelegt.

Der im Initialisierungsbrief angegebene Hashwert wird ebenfalls nach diesem Verfahren über die 256 Bytes – bestehend aus öffentlichem Exponenten und Modulus – berechnet.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Aufbau der Unterschriftsdatei

Vor dem Versand einer Elektronischen Unterschrift wird diese in eine separat zu übertragende Unterschriftsdatei eingestellt, die den folgenden Aufbau hat:

Inhalt	Länge in Bytes	Datenformat ²⁸	Belegung	Erläuterung
Versionsnummer	4	an	,A004'	
Länge des Modulus	4	n	,1024'	
Auftragsart	3	an	z.B. 'IZV'	Auftragskürzel der Originaldatei
EU	128	binär	'0, ..., 0, SIGNATUR'	rechtsbündig
User-ID	8	an	z.B. 'A2B2C2D2'	
Originaldatei	128	an		Lokaler Dateiname der Originaldatei ²⁹
Datum/Uhrzeit	16	an	jjjjmmttX'20'hhmmssX'20'	
Datum/Uhrzeit	16	an	jjjjmmttX'20'hhmmssX'20'	
Frei nutzbares Feld	8	binär	X'00'	Zur Zeit nicht benutzt
Reserve	197	binär	X'00'	Zur Zeit nicht benutzt

Der Aufbau der Unterschriftsdatei ist für die EU-Versionen A003 und A004 identisch (vergl. Kapitel [2.2.2.5 „Formate“](#))

²⁸ an = alphanummerisch; n = numerisch; Werte im ASCII-Format werden linksbündig eingestellt und rechts mit Blanks (X'20') aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt.

²⁹ Bei EDIFACT ist der lokale Dateiname der Originaldatei – im Gegensatz zu BCS – wie folgt strukturiert:

35 Bytes	SenderID (linksbündig, rechts mit X'20' auffüllen)
58 Bytes	Interchange Control Reference (Linksbündig, rechts mit X'20' auffüllen)
35 Bytes	EmpfängerID (linksbündig, rechts mit X'20' auffüllen)

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Beispiel für einen INI-Brief DFÜ (A004)

Benutzername	system	Kundensoftware-interner Name (Angabe freigestellt)
Datum	TT.MM.JJJJ	Datum der Erstellung des Initialisierungs- bzw. Public-Key-Änderungsauftrages
Uhrzeit	HH:MM	Uhrzeit der Erstellung des Initialisierungs- bzw. Public-Key-Änderungsauftrages
Empfänger	DFÜ-Bank	Hostname der Bank (max. 8 Stellen; wird von der jeweiligen Bank mitgeteilt)
User-ID	xxxxxxx	(8 Bytes alphanummerisch; beginnend mit einem Alphazeichen; wird von der jeweiligen Bank mitgeteilt)
Kunden-ID	yyyyyyy	(8 Bytes alphanummerisch; wird von der jeweiligen Bank mitgeteilt)
EU-Version	A004	

Öffentlicher Schlüssel (Public-Key) für die Elektronische Unterschrift:

Exponent 1024

00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	01

Modulo 1024

FF	12	03	26	E6	30	90	A5	06	01	EF	16	10	21	EE	D4
77	23	27	A9	14	17	07	F1	71	25	22	D5	91	00	41	0A
D7	4A	2F	D5	6C	16	4E	C3	2D	82	F3	02	31	CD	FF	FB
45	77	E4	7E	E5	B2	CB	7B	9A	5F	75	7B	32	7C	16	E5
FB	16	41	0B	4A	39	0F	50	47	68	9C	9B	27	D2	A0	9C
CA	23	A8	C3	1C	AB	A5	ED	72	75	9D	0A	B8	9B	37	BA
00	CB	68	BB	AC	C8	D1	C8	D3	35	C8	BF	1F	A3	06	CF
24	5A	DC	EB	84	64	86	D0	97	8F	E4	67	08	78	81	07

Hash	D2	FD	56	F3	1E	5C	76	D2	B8	2C
	0B	1E	4C	6A	13	9E	85	87	E8	D3

Ich bestätige hiermit den obigen öffentlichen Schlüssel für meine Elektronische Unterschrift.

Ort / Datum	Firma / Name	Unterschrift
-------------	--------------	--------------

2.2.3.7 Testdaten Elektronische Unterschrift³⁰

1. Testdatei: TEST.DAT – 24 Byte

2. Testdatei: 255.IZV – 640 Byte

Hashfunktion: AR/DFP

k1:	90	26	28	CB	EC	46	15	43
k2:	2A	41	52	2F	44	46	50	2A

Elektronische Unterschrift A004: Beispiel 1

1024-Bit RSA-Schlüssel
Modulus:

```
a1 69 44 f5 ac 4a a0 9e 0d 83 09 92 24 cb df 87
0c bb 66 13 19 e3 52 57 34 39 66 33 63 f8 e6 eb
36 38 9d 45 e6 d9 59 79 c6 df 1f 22 2d 56 6c 1c
ef 0f a2 f8 b8 0f 17 3c cd f8 c4 fa 66 16 40 5c
e0 b0 89 fd 96 8d 8d 00 ad d3 c5 42 7b 76 9c bb
af 20 1f 7e 98 59 a2 e1 3c 6f 5f 03 f2 af 38 33
67 48 23 6f 81 93 4a b0 dc 56 0f 40 49 05 c1 11
1f 9c 63 68 fb 78 58 42 93 c6 06 fd b9 60 7c cf
```

Öffentlicher Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 01 00 01
```

Geheimer Exponent:

```
51 14 27 e7 a0 1b fc 82 68 17 22 7f 9e ac 2a 24
14 69 2d e4 fa 64 0d c0 70 23 45 0b 1b 74 b3 ca
02 fa 7d 45 d7 a7 e6 22 1d 9b 86 70 0c 86 14 d8
93 dd 2e e7 f0 cc e3 c7 4f 4f 5e e2 c8 d4 f4 8e
e6 1d 7f 1f ed bd 67 3b 78 ec 5a 23 eb 84 a1 99
5e 25 6e 0f 33 94 6f f9 d8 ad 40 bb cd c0 7a b6
40 4e 90 ac 2b 77 3d ff 2d 32 1e 0e 73 f4 d0 d6
d3 f9 5b 61 2e 5a 5c 5c c1 b2 98 87 36 8c 4f 41
```

Hashwert (des öffentlichen Schlüssels):

```
e4 b0 08 b9 f7 25 20 ce a1 96 aa 4f b4 97 70 d9
f4 d3 b5 b3
```

³⁰ Die Dateien TEST.DAT und 255.IZV können über DFUE@SIZ.de angefordert werden.

Elektronische Unterschrift der 1. Testdatei:

37 37 51 30 f7 ed cf ec 65 21 da b6 8b ab b9 a4
28 e6 f7 62 50 09 12 23 5e 93 8e 78 6b 89 69 3d
89 fd fe 20 3f 8b 9d 50 ca a4 54 ec 0d c0 48 e8
58 8c 54 26 d9 83 a0 58 b7 e4 d8 39 ce 68 9a b2
c7 a1 27 30 b4 ac 00 5e 72 f1 71 b2 df 63 28 7a
14 92 67 22 1c 0b d4 e5 71 58 b0 cb 13 a4 8c 82
a7 f1 be 5c 53 65 22 d2 4e e3 32 74 65 13 fc 84
46 2d 88 80 33 11 47 35 31 32 4d aa 97 cc 4e 5a

Unterschriebener DSI:

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 01 9b 0c 62 08 aa 03 f2 f6 2c 40 3e d7 f2
21 75 85 29 70 40 21 0f 01 bc c7 27 72 ff 60 bc

Unterschriebener Hashwert:

2c 40 3e d7 f2 21 75 85 29 70 40 21 0f 01 bc c7
27 72 ff 60

Elektronische Unterschrift der 2. Testdatei:

3c 1d d7 1e ec 68 96 9d 6f 9f e3 ed 60 b0 ce 44
60 71 ef f1 95 f5 18 a7 0c 79 ef 15 43 33 fc af
0d 63 82 7f 03 89 83 73 d5 2d 04 62 56 74 e5 77
56 1c 08 f4 de bf c2 10 68 78 8c ca c7 d9 05 73
f9 ef 3d 13 3f 69 d1 cc 69 4d 74 90 4c bf e4 ca
19 3f 7c fd 3d 49 e5 41 0d 86 f1 60 92 69 79 2f
83 56 5a 87 49 6e 28 4b 88 46 a3 c0 1d de 03 c3
b2 c4 58 3b f9 20 d3 17 9c c3 8f c5 21 e5 d1 ac

Unterschriebener DSI:

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 01 93 be ae 46 79 99 c6 65 72 26 48 c9 9c
0d 97 c2 de 90 8b 48 a6 5e 2a 33 16 43 2e 8e bc

Unterschriebener Hashwert:

72 26 48 c9 9c 0d 97 c2 de 90 8b 48 a6 5e 2a 33
16 43 2e 8e

Elektronische Unterschrift A004: Beispiel 2

1024-Bit RSA-Schlüssel

Modulus:

95 f9 55 6e 4f d3 e2 eb 04 39 ab 69 4e d6 2c 17
f6 ef 22 41 eb 8d 94 a2 38 b4 f1 de 90 2b d2 12
f4 08 db 72 b8 57 b9 cc af a4 6e 0f a9 c1 9d a8
09 4b 5c 69 fa 8f 0c 03 ff b1 2b 55 18 bf 93 66
e0 5c 36 39 a4 e8 a7 ba 03 3e 50 13 dc 37 7d ec
03 74 5e 01 9f 24 08 d1 58 8d e0 d7 5a 22 94 99
92 54 b3 c5 97 20 f7 51 e3 83 94 78 ba fa 20 51
c3 ed c5 82 3a ed 83 79 ad c1 48 7e 32 f0 2f a1

Öffentlicher Exponent:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 01 00 01

Geheimer Exponent:

2a d5 81 be f5 f1 d0 97 cb 27 25 7b f8 56 57 55
a8 e3 77 e7 57 fd a1 fc 0b 32 32 fa 9c 36 c7 d3
3d fb e1 a1 8c 61 11 e2 12 30 66 76 f6 c7 23 de
40 79 53 b5 b9 28 6e 08 1e 59 5d c4 fa 42 8d 38
9b fc 0f 82 63 52 14 85 39 a7 8c 12 97 04 da 6d
8d 76 ec 00 09 cb 7b da b3 0e 7f 01 a1 ce c4 d8
01 2d be 1b d0 4c ce c7 b9 00 a3 d7 d3 e6 5b fd
ba 38 16 da e3 78 1f a9 3e cd eb d0 ae 9e 23 79

Hashwert (des öffentlichen Schlüssels):

ae 54 a4 a0 6d 53 6f f3 9b 55 78 42 9b 02 66 75
97 49 61 f0

Elektronische Unterschrift der 1. Testdatei:

34 0f 08 28 cc dd b9 bc fb 3d e0 d8 94 65 20 f0
fb 20 a0 49 cf 7f 5f 85 8f 1d 23 06 51 d0 28 cf
88 03 36 2c ff 4c 86 1a 88 4a c2 ad 9e 93 14 65
85 c1 80 30 2a 2d cc ed f4 a3 7e 24 92 03 ef 8e
c0 d1 dd fb 6e 2f 8e 91 45 1e f2 d5 59 00 4d e5
3b c4 5b 58 fc 34 94 7a 9d 3f 6f 29 b6 38 02 6b
91 0e 8e 57 ec 55 ef f6 6f ce 10 56 f3 8d 41 70
cc f6 c8 0c 67 db 82 62 f0 ca 62 c8 16 64 ba 60

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Unterschriebener DSI:

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 01 2f 24 a4 dd f0 50 9b bb 2c 40 3e d7 f2
21 75 85 29 70 40 21 0f 01 bc c7 27 72 ff 60 bc

Unterschriebener Hashwert:

2c 40 3e d7 f2 21 75 85 29 70 40 21 0f 01 bc c7
27 72 ff 60

Elektronische Unterschrift der 2. Testdatei:

1d 1b b2 c5 6c 10 78 54 be 6f 76 29 0b 1e 23 84
a7 7c fb 7b 20 e3 06 b1 23 44 77 94 c4 43 e4 16
ca 7f a3 8a 28 30 5b 02 bf e7 9b 37 e2 0f eb 0c
ef 03 cf 99 ca 73 8e d9 28 01 73 8a 7a 69 e0 2a
01 c2 4b 58 d4 85 da e3 18 07 e8 ad 6b 3f 12 51
45 5e 68 30 b7 60 b3 ac d6 85 45 d6 c9 72 2b 7c
60 01 f1 75 3c b7 1c b0 39 5c d5 17 cb 14 d2 e2
16 ff cd 03 f5 96 36 99 65 e2 03 26 cb 16 33 57

Unterschriebener DSI:

60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 01 42 55 c9 0e 03 b7 9a 8b 72 26 48 c9 9c
0d 97 c2 de 90 8b 48 a6 5e 2a 33 16 43 2e 8e bc

Unterschriebener Hashwert:

72 26 48 c9 9c 0d 97 c2 de 90 8b 48 a6 5e 2a 33
16 43 2e 8e

2.3 Verschlüsselung

2.3.1 Allgemeine Anforderungen

Kunde und Kreditinstitut vereinbaren, welche Daten von der Bank verschlüsselt bereitgestellt werden sollen. Das Verschlüsselungsverfahren muss in jedem Fall der nachfolgenden Spezifikation entsprechen. Die konkrete technische Lösung muss für die gängigen Betriebssysteme, die beim Unterzeichner bzw. Empfänger zum Einsatz kommen können, kompatibel einsetzbar sein.

2.3.2 Schaffung der Voraussetzungen für die verschlüsselte Kommunikation

Bei dem zur Verschlüsselung eingesetzten Verfahren handelt es sich um ein **hybridisches Verfahren**, in dem der Kunde bei der Vorbereitung des Public-Key-Austausches die VPK-Datei dem Kreditinstitut zur Verfügung stellen muss.

Der Kunde überträgt die VPK-Datei an das Kreditinstitut mittels DFÜ. Zur Autorisierung gegenüber der Bank stehen dem Kunden zwei Möglichkeiten zur Verfügung:

- Signieren der VPK-Datei mittels Elektronischer Unterschrift. Dazu muss das Verfahren der Elektronischen Unterschrift beim Kunden bereits initialisiert sein.
- Generierung des INI-Briefes und Versand der VPK-Datei und des INI-Briefes zum Kreditinstitut.

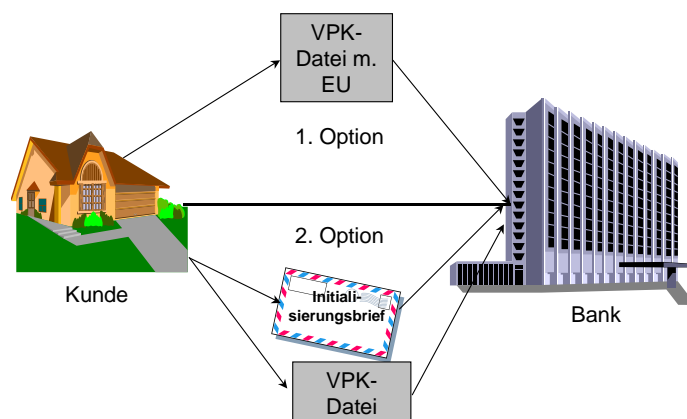


Abbildung 2.2: Generierung und Versand des INI-Briefes/ VPK-Datei zum Kreditinstitut

2.3.3 Vorbereitung der Verschlüsselung / Public-Key-Austausch

Als Voraussetzung für die Datenverschlüsselung muss der jeweilige Empfänger der verschlüsselten Daten dem Sender dieser Daten den öffentlichen Schlüssel seines RSA-

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Verschlüsselungs-Key-Paares bekannt machen.

Im Rahmen der Kunde-Bank-Beziehung übermittelt der Kunde seinen Verschlüsselungs-Public-Key unter Nutzung der neu einzuführenden Auftragsart „VPK“ (= Verschlüsselungs-Public-Key-Kunde) mit folgender Datei:

Inhalt der VPK-Datei	Länge in Bytes ³¹	Erläuterung/Belegung
Versionsnummer	4 ASCII	,V001'
Verschlüsselungsverfahren	2 ASCII	,03' (=Triple DES mit zwei 64 Bit Schlüsseln)
Kunden-ID	8 ASCII	z.B. 'A1B1C1D1'
Länge Exponent	4 ASCII	'0768'
Exponent	128 binär	00 ... 01 00 00 00 0F(Hex) (siehe Kapitel 2.3.6.3 „Exponent“). Gegebenenfalls können auch variable Werte für den Exponenten verwendet werden oder die 4. Fermat'sche Primzahl. Es ist sicherzustellen, dass nicht der schwache Exponent 3 verwendet wird.
Länge Modulo	4 ASCII	'0768'
Modulo	128 binär	Siehe Kapitel 2.3.6.4 "Modulo"
Hashwert des Public Keys	16 binär	Hashwertbildung gemäß Kapitel 2.2.2.2 „Definitionen“
Verwendung	2 ASCII	'05' (=nur für Verschlüsselung)
Zeitstempel der Key-Generierung	20 ASCII	Format: 'TT.MM.JJJJ,X'20'HH:MM:SS' (Datum/Uhrzeit)
Reserve	196 ASCII	Auffüllen mit X'20'
EU-Datei gemäß Kapitel 2.2 „Kryptographische Verfahren ...“	512 binär	Feld wird nur angehängt, wenn Legitimation mit Elektronischer Unterschrift stattfindet ³²

Wenn die Legitimierung des VPK-Auftrages mit der Elektronischen Unterschrift durchgeführt wird, so wird die obige VPK-Datei und die zugehörige Unterschriftsdatei verknüpft und als eine Datei übertragen. Gleichzeitig wird der FTAM-Remote-Filename um die User-ID desjenigen erweitert, der die Elektronische Unterschrift generiert hat. Die User-ID wird in alphanumerisch mit der führenden Kennung „U“ als Auftragsparameter übergeben.

Der bankseitige Verschlüsselungs-Public-Key wird dem Kunden mit der nachfolgend dargestellten Datei - falls die Übermittlung im Wege der Datenfernübertragung erfolgt - unter Nutzung der neu einzuführenden Auftragsart „VPB“ (Verschlüsselungs-Public-Key-Bank) zur Verfügung gestellt.

³¹ an = alphanummerisch; n = numerisch. Alphanummerische Felder werden linksbündig angeordnet und rechts mit Blanks (X'20') aufgefüllt. Numerische Felder werden rechtsbündig angeordnet und links mit Nullwerten (X'30') aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt.

³² Sofern die Autorisierung dieser Verschlüsselungs-Public-Key-Datei mit der Elektronischen Unterschrift stattfindet (alternativ ist die Legitimation per Initialisierungsbrief vorgesehen), wird die VPK-Datei um die entsprechende, nach dem in Kapitel 2 beschriebenen Verfahren für die Elektronische Unterschrift erstellte EU-Datei erweitert. In diesem Fall wird das Auftragsattribut „Dateiart“ im FTAM-Remote-Filename mit „B“ (= Original- und EU-Datei in einer physikalischen Datei) belegt. Im Falle des Versands eines Initialisierungsbriefes wird dieses Attribut auf „D“ gesetzt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Inhalt der VPB-Datei	Länge in Bytes ³³	Erläuterung/Belegung
Versionsnummer	4 ASCII	,V001'
Host-ID	8 ASCII	z.B. 'A3B3C3D3'
Länge Exponent	4 ASCII	'0768'
Exponent	128 binär	00 ... 01 00 00 00 0F(Hex) (siehe Kapitel 2.3.6.3 „Exponent“). Gegebenenfalls können auch variable Werte für den Exponenten verwendet werden oder die 4. Fermat'sche Primzahl. Es ist sicherzustellen, dass nicht der schwache Exponent 3 verwendet wird.
Länge Modulo	4 ASCII	'0768'
Modulo	128 binär	Siehe Kapitel 2.3.6.4 „Modulo“
Hashwert des Public Keys	16 binär	Hashwertbildung gemäß Kapitel 2.2.2.2 „Definitionen“
Verwendung	2 ASCII	'05' (=nur für Verschlüsselung)
Reserve	196 ASCII	Auffüllen mit X'20'

³³ Alphanummerische Felder werden linksbündig angeordnet und rechts mit Blanks (X'20') aufgefüllt. Numerische Felder werden rechtsbündig angeordnet und links mit Nullwerten (X'30') aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt.

2.3.4 Ver- und Entschlüsselung

2.3.4.1 Vorgänge beim Sender

Generierung des geheimen DES-Schlüssels (2-Key-Triple-DES)

Es werden zwei zufällige Bitstrings DEK_{left} und DEK_{right} mit jeweils einer Länge von 64 Bit generiert. Die Verknüpfung von DEK_{left} und DEK_{right} wird als DEK bezeichnet.

Es sei

$$DEK = DEK_{left} || DEK_{right} = x_{127}, \dots, x_0$$

mit $DEK_{left} = x_{127}, \dots, x_{64}$ und $DEK_{right} = x_{63}, \dots, x_0$.

Bei der Interpretation der DES-Keys als natürliche Zahl wird angenommen, dass das jeweils linke Bit (x_{127} bzw. x_{63}) der Schlüssel als höchstwertigstes Bit der Zahl aufgefaßt wird.

Prüfen der geheimen DES-Schlüssel

Die generierten Zufallszahlen, die als rechte und linke Schlüsselhälfte des 2-Key-Triple-DES verwendet werden, sind daraufhin zu überprüfen, dass es sich nicht um einen schwachen oder halbschwachen DES-Schlüssel handelt

Die schwachen Schlüssel des DES							
01	01	01	01	01	01	01	01
FE	FE	FE	FE	FE	FE	FE	FE
1F	1F	1F	1F	0E	0E	0E	0E
E0	E0	E0	E0	F1	F1	F1	F1
Die halbschwachen Schlüssel des DES							
01	FE	01	FE	01	FE	01	FE
FE	01	FE	01	FE	01	FE	01
1F	E0	1F	E0	0E	F1	0E	F1
E0	1F	E0	1F	F1	0E	F1	0E
01	E0	01	E0	01	F1	01	F1
E0	01	E0	01	F1	01	F1	01
1F	FE	1F	FE	0E	FE	0E	FE
FE	1F	FE	1F	FE	0E	FE	0E
01	1F	01	1F	01	0E	01	0E
1F	01	1F	01	0E	01	0E	01
E0	FE	E0	FE	F1	FE	F1	FE
FE	E0	FE	E0	FE	F1	FE	F1

Vorbereitung zur Verschlüsselung von DEK

Der als natürliche Zahl interpretierte 128 Bit DEK wird vor dem höchstwertigsten Bit mit Nullbits auf 768 Bit aufgefüllt. Das Ergebnis heißt **PDEK** (siehe Kapitel 2.3.6.1 „PDEK“).

Verschlüsselung des geheimen DES-Schlüssels

PDEK wird anschließend mit dem öffentlichen Schlüssel des RSA-Schlüsselsystems des Empfängers verschlüsselt und anschließend mit führenden Nullbits auf 1024 Bit erweitert.

Das Ergebnis heißt **EDEK** (siehe Kapitel 2.3.6.2 „EDEK“). Es muss sichergestellt sein, dass EDEK ungleich DEK ist.

Verschlüsselung der Nachrichten

Padding der Nachricht:

Für das Padding der Nachricht wird die Methode **Padding with Octets** nach ANSI X9.23 angewendet, d. h. es werden in jedem Fall Daten an die zu verschlüsselnde Nachricht angefügt.

Anwendung des Verschlüsselungsalgorithmus:

Die Nachricht wird mit dem geheimen Schlüssel DEK nach dem 2-Key-Triple-DES-Verfahren, wie im ANSI X3.92-1981 spezifiziert, im CBC-Mode gemäß ANSI X3.106 verschlüsselt.

Hierbei wird folgender Initialisierungswert „ICV“ verwendet: X '00 00 00 00 00 00 00 00'.

Übermittlung

An den Kommunikationspartner wird der verschlüsselte 2-Key-Triple-DES-Schlüssel (bestehend aus DEK_{left} und DEK_{right}) (=EDEK) in einem der verschlüsselten Nachricht vorangestellten Vorsatz 'V' übertragen. Die zur Übertragung kommende Datei hat damit insgesamt folgenden Aufbau:

Feld	Inhalt	Länge in Bytes ³⁴	Erläuterung/Belegung
1	Satzart	1 an	'V'
2	Versionsnummer	3 an	Derzeit '001'
3	Vorsatzlänge	4 n	'0256'
4	Absender-ID	8 an	Bei Sendeauftrag des Kunden: Kunden-ID (z.B. 'A1B2C3D4') Bei Abholauftrag des Kunden: Host-ID (z.B. 'BANKABCD')
5	Empfänger-ID	8 an	Bei Sendeauftrag des Kunden: Host-ID (z.B. 'BANKABCD') Bei Abholauftrag des Kunden: Kunden-ID (z.B. 'A1B2C3D4')

³⁴ an = alphanummerisch; n = numerisch. Werte im ASCII-Format werden linksbündig eingestellt und rechts mit Blanks (X'20') aufgefüllt. Numerische Felder werden rechtsbündig angeordnet und links mit Nullwerten (X'30') aufgefüllt. Werte im Binär-Format werden rechtsbündig eingestellt und links mit X'00' aufgefüllt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Feld	Inhalt	Länge in Bytes ³⁴	Erläuterung/Belegung
6	Verschlüsselter DEK (EDEK)	128 binär	Siehe Kapitel 2.3.6.2 "EDEK"
7	Hashwert des Verschlüsselungs-Public-Keys des Empfängers	16 binär	Bei Sendeauftrag des Kunden: Hashwert des VPB Bei Abholauftrag des Kunden: Hashwert des VPK (Hashwertbildung gemäß Kapitel 2.2.2.2 „Definitionen“)
8	Reserve	88 an	Zur Zeit nicht genutzt, X'20'

Nach dem Vorsatz folgt die verschlüsselte Nachricht

Feld	Inhalt	Länge in Bytes	Erläuterung/Belegung
9	Verschlüsselte Nachricht	binär variabel	Mit DEK verschlüsselte Originalnachricht

Zusätzlich wird der FTAM-Remote-Filename um den obigen Hashwert des Verschlüsselungs-Public-Keys des Empfängers im Feld „Auftragsparameter“ erweitert. Dieser Hashwert (16 Bytes binär) wird als ASCII-Text in hexadezimaler Schreibweise mit der führenden Kennung „H“ übergeben. Die Länge des Feldes Auftragsparameter im FTAM-Remote-Filename muss daher um 33 Bytes erweitert werden.

2.3.4.2 Vorgänge beim Empfänger

Prüfen des Hashwertes des Verschlüsselungs-Public-Keys

Der im FTAM-Remote-Filename übergebene Hashwert des Verschlüsselungs-Public-Keys des Empfängers wird mit dem empfängerseitig hinterlegten Hashwert verglichen. Anhand des Vergleichs wird die Aktualität des Verschlüsselungscodes des Senders überprüft. Wurde ein unzulässiger Hashwert empfangen, so wird die FTAM-Kommunikation beendet.

Handelt es sich beim Empfänger um die Bank, so werden folgende Antwortcodes³⁵ generiert, die kundenseitig die entsprechend beschriebenen Reaktionen auslösen, sofern im FTAM-Remote-Filename die Versionskennung des Anwendungsprotokolls „A3“ verwendet wird:

³⁵ Weitere für andere Zwecke reservierte Antwortcodes sind dem Kapitel 2.3.6 „Interne Datenformate der Funktion „Verschlüsselung““ zu entnehmen.

Antwortcode	Bemerkung
51	Verschlüsselungscode mit Bank muss aktualisiert werden. Auf dem Banksystem wurde ein neuer Verschlüsselungs-Public-Key generiert, der vom Kundensystem abgeholt werden muss. Der Anwender wird hierüber durch eine entsprechende Meldung hingewiesen. Solange der neue Verschlüsselungs-Public-Key bei der Bank noch nicht abgeholt wurde, kann der Kunde keine (verschlüsselten) Dateien an die Bank übermitteln. Der Antwortcode 51 kann auch dann vom Banksystem zurückgeliefert werden, wenn der im FTAM-Remote-Filename der verschlüsselten Datei eingestellte Hashwert des verwendeten Verschlüsselungs-Public-Keys der Bank (VPB) nicht mit dem bankseitig erwarteten Hashwert übereinstimmt. In diesem Fall muss der Kunde ebenfalls einen neuen Verschlüsselungs-Public-Key bei der Bank abholen.
54	Verschlüsselungscode muss neu verschickt werden. Kundenseitig wird nach erfolgreichem Versand des VPK-Auftrages und des dazugehörigen Legitimationsauftrages zunächst davon ausgegangen, dass die entsprechende Authentifizierung bei der Bank mit positivem Ergebnis durchgeführt wurde. Verläuft die bankseitige Unterschriftsprüfung der VPK-Datei jedoch negativ, so erhält das Kundensystem beim Abholversuch von verschlüsselt vereinbarten Auftragsarten den obigen Antwortcode 54.
56	Verschlüsselungscode noch nicht freigegeben. Solange der kundenseitige Verschlüsselungs-Public-Key bankseitig noch nicht authentifiziert wurde, erhält der Kunde bei dem Versuch, Daten mit den Auftragsarten abzuholen, für die die Verschlüsselung mit der Bank vereinbart wurde, den Returncode 56 „Verschlüsselungscode noch nicht freigegeben“.

Entschlüsselung des DES-Schlüssels

Die führenden 256 Nullbits des EDEK werden entfernt und die verbleibenden 768 Bit mit dem geheimen Schlüssel des RSA-Schlüsselsystems des Empfängers entschlüsselt. Anschließend liegt PDEK vor. Aus den niederwertigsten 128 Bits von PDEK erhält man den geheimen DES-Schlüssel DEK, der in die Einzelschlüssel DEK_{left} und DEK_{right} aufgesplittet wird.

Entschlüsselung der Nachricht

Mit dem geheimen DES-Schlüssel (bestehend aus DEK_{left} und DEK_{right}) wird die verschlüsselte Originalnachricht nach dem 2-Key-Triple-DES-Verfahren im CBC-Mode entschlüsselt. Hierbei wird wiederum der Initialisierungswert ICV (siehe Kapitel 2.3.6.1 „PDEK“) verwendet.

Entfernen der Paddinginformationen

Für das Entfernen der Paddinginformationen aus der entschlüsselten Nachricht wird die Methode „Padding with Octets“ nach ANSI X9.23 angewendet. Anschließend liegt die Originalnachricht unverschlüsselt vor.

2.3.5 Beispielhafte Beschreibung der Abläufe

Beispielparameter

Im Folgenden werden die Anwendungen dieser Funktionen anhand eines Beispiels mit folgenden Parametern beschrieben:

Parameter	Wert
Kunden-ID	'A1B1C1D1' (8-stellig, alphanummerisch)
User-ID	'A2B2C2D2' (8-stellig, alphanummerisch)
Auftragsart Originaldatei	'TST'
Auftragsart Unterschrift	'TST'
Auftragsnummern	'A000'
FTAM spezifische Parameter:	
Host-ID	'A3B3C3D3' (8-stellig, alphanummerisch)
Password	8-stellig, alphanummerisch
Verschlüsselungs-Public-Key des Kunden:	'A1B1C1D1'
Exponent	'00 .. 01 00 00 0F' (Hex)
Lmodulo	'0768'
Modulo	'0 ... a7f885 ... 65' (Hex)
Verwendungszweck ³⁶	'05'

Allgemeiner Aufbau des Remote-Filename

Das Feld enthält die inhaltliche Beschreibung des Transferauftrages und gegebenenfalls auftragspezifische Daten. Nur Byte 1 bis 44 werden genutzt. Die einzelnen Parameter sind durch Punkte voneinander getrennt. Aus Kompatibilitätsgründen beim Empfänger-Betriebssystem ist das erste Byte eines Qualifiers immer ein Alphazeichen. Die Grundstellung für nicht belegte Stellen ist 'N'. Der Aufbau ist:

PV.Kunden-ID.Auftragsart.Auftragsattribut[.Auftragsnummer] [.Auftragsparameter]

Erläuterung:

- PV (2 Bytes alphanummerisch):
Versionsnummer des Anwendungsprotokolls; zur Zeit konstant '**A3**'
- Kunden-ID (8 Bytes alphanummerisch):
Bankspezifische Kunden-ID
- Auftragsart-Kennung (3 Bytes alphanummerisch, siehe 1.4 „Auftragsartenkennungen“)

³⁶ Das Feld „Verwendungszweck“ ist zunächst ohne Funktion und wird nur vorläufig belegt.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

- Auftragsattribut (5 Bytes alphanummerisch):

Byte	Inhalt	Auswahlmöglichkeiten
1	Dateiart	O = Originaldatei mit zugehöriger Unterschriftsdatei U = Unterschriftsdatei D = Originaldatei ohne zugehöriger Unterschriftsdatei B = Originaldatei und EU-Datei in einer physikalischen Datei P = Protokolldatei Y = Dummy-Datei
2	Komprimierungsart ³⁷	N = Keine Komprimierung F = FLAM-Komprimierung Z = PKZIP-Komprimierung X = x-Press-Komprimierung ... Die zum Einsatz kommenden Komprimierungsprodukte werden durch die ZKA-Buchstabenkennungen festgelegt.
3	Verschlüsselungsart	N = Keine Verschlüsselung H = Hybrid-Verfahren DES/RSA R = RSA
4	Reserve	
5	Reserve	

- Auftragsnummer (4 Bytes alphanummerisch):
Je Kunde eindeutige Auftragsnummer für die Übertragungsrichtung Kunde/Bank. Dient als Ordnungsbegriff bei der PC-Host-Kommunikation, insbesondere beim Protokollabruf und bei der Synchronisation von Original- und Unterschriftsdatei.

Byte	Inhalt
1	A
2	laufende Nummer des Kunden-DFÜ-PCs
3-4	2 Stellen alphanummerisch (werden aufsteigend vergeben)

- Auftragsparameter:
Auftragsspezifische weitere Parameter, falls für die Auftragsart erforderlich bzw. zugelassen. Zur Zeit:

³⁷ Sofern nichts anderes vereinbart, ist das FLAM-Verfahren das Standard-Komprimierungsverfahren.

Auftragsart	Auftragsparameter
STA	optional Datum-von Datum-bis (VJJMMTT.BJJMMTT)
VPK	Kennung für User-ID des Kunden, der VPK-Auftrag unterschrieben hat Byte 1: U (Kennung für User-ID) Byte 2 – 9: User-ID
alle verschlüsselt zu übertragenden Dateien	Verschlüsselungs-Public-Key-Hashwert (33 Bytes alphanummerisch): Byte 1: H (Kennung für Hashwert) Byte 2-33: Verschlüsselungs-Public-Key in Hex-Darstellung

RSA-Key-Paar-Generierung und Verteilung des Verschlüsselungs-Public-Keys des Kunden zur Verschlüsselung des DEK

Damit der Empfänger verschlüsselter Daten diese wieder entschlüsseln kann, wird der Verschlüsselungsschlüssel zusammen mit der verschlüsselten Originalnachricht übermittelt. Es muss jedoch sichergestellt sein, dass nur der berechtigte Empfänger den Verschlüsselungsschlüssel verwenden kann. Deshalb wird der Verschlüsselungsschlüssel (DEK) seinerseits verschlüsselt. Zu diesem Zweck benötigt der Sender verschlüsselter Daten den öffentlichen RSA-Schlüssel des jeweiligen Empfängers.

Kundenseite Rahmenprogramm:

- Erzeugung und Speicherung von Verschlüsselungs-Public Key (VPK_{Kunde}) und zugehörigem Verschlüsselungs-Secret Key (VSK_{Kunde}).

Im Kundensystem gibt es genau ein RSA-Schlüsselpaar für die Verschlüsselung des Verschlüsselungsschlüssels (DEK). Dies ist im Unterschied zu den RSA-Schlüsselpaaren für die Elektronische Unterschrift zu sehen, bei dem ein Schlüsselpaar pro unterschriftsberechtigtem User eines Kunden generiert wird.

- Generierung der Verschlüsselungs-Public-Key-Datei (VPK_{Kunde})

Die Übertragung der Verschlüsselungs-Public-Key-Datei kann mit einer Elektronischen Unterschrift eines unterschriftsberechtigten Users des Kunden abgesichert werden. In diesem Fall setzt die Übertragung von VPK_{Kunde} die erfolgreiche Ausführung der in Kapitel 2.3.6.1 „PDEK“ beschriebenen Abläufe für mindestens ein für die EU zu verwendendes RSA-Schlüsselpaar voraus. Sollte das Verfahren zur Elektronischen Unterschrift noch nicht initialisiert worden sein, so muss die Generierung des Ini-Briefes erfolgen (siehe auch Kapitel 2.2 „Kryptographische Verfahren des deutschen Kreditgewerbes für die Elektronische Unterschrift im Rahmen der Kunde-Bank-Kommunikation“).

Für die Beschreibung der weiteren Abläufe wird auf Kapitel **2.3.6.1 „PDEK“** verwiesen. Die entsprechenden DFÜ-Aufträge erhalten das Auftragsattribut VPK, d. h. die entsprechenden DFÜ-Aufträge sind:

- Beispiel für einen VPK-Auftrag mit Elektronischer Unterschrift:
A3.KUNDE1.VPK.BNNNN.A001.UUSER1
- Beispiel für einen VPK-Auftrag ohne Elektronischer Unterschrift:
A3.KUNDE1.VPK.DNNNN.A001

Bankseite Rahmenprogramm:

Die entsprechenden bankseitigen Abläufe sind in Kapitel 2.3.6.1 „PDEK“ beschrieben. In Ergänzung dazu erfolgen nach der Verifizierung der EU und Eintrag der Prüfungsergebnisse ins EU-Protokoll die folgenden Schritte:

- Erfolgsfall bei EU-Prüfung: Automatische Freischaltung und Speicherung von VPK_{Kunde}
- Fehlerfall bei EU-Prüfung: Ablehnung der Verschlüsselungs-Public-Key-Datei
- Protokollierung der Ergebnisse

Übertragung von verschlüsselten Daten von der Bank zum Kunden

Bankseitige Bereitstellung der Daten:

Unter den im vorherigen Abschnitt beschriebenen Voraussetzungen zur Bereitstellung und Verschlüsselung der Kundendatensätze auf der Bankseite wird hier folgende Funktionalität bereitgestellt:

- Generierung eines zufälligen DES-Schlüssels DEK
- Verschlüsselung der Originalnachricht mit DEK
- Verschlüsselung von DEK mit VPK_{Kunde}
- Konkatenation von Dateivorsatz und verschlüsselter Originalnachricht zu einer V-Datei
- Bereitstellung der V-Datei für den Abholauftrag

Kundenseitiges Empfangen von Daten:

- Einrichten eines DFÜ-Auftrages zum Abholen der V-Datei, z. B. A3.A1B1C1D1.TST.DNHNN
- Empfang der V-Datei
- Entschlüsselung des verschlüsselten DES-Schlüssels aus dem Dateivorsatz mit VSK_{Kunde}
- Entschlüsselung der Nachricht mit dem so erhaltenen DEK

Übertragung von verschlüsselten Daten vom Kunden zur Bank

Die folgende Beschreibung dokumentiert den Ablauf für den Übertragungsweg vom Kunden zur Bank.

Kundenseitige Bereitstellung der Daten:

Unter den im vorherigen Abschnitt beschriebenen Voraussetzungen zur Bereitstellung und

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Verschlüsselung der Sendeaufträge auf der Kundenseite wird hier folgende Funktionalität bereitgestellt:

- Generierung eines zufälligen DES-Schlüssels DEK
- Verschlüsselung der Originalnachricht mit DEK
- Verschlüsselung von DEK mit VPB_{Bank}
- Konkatenation von Dateivorsatz und verschlüsselter Originalnachricht zu einer V-Datei
- Bereitstellung der V-Datei für den Sendeauftrag
- Starten der DFÜ ³⁸

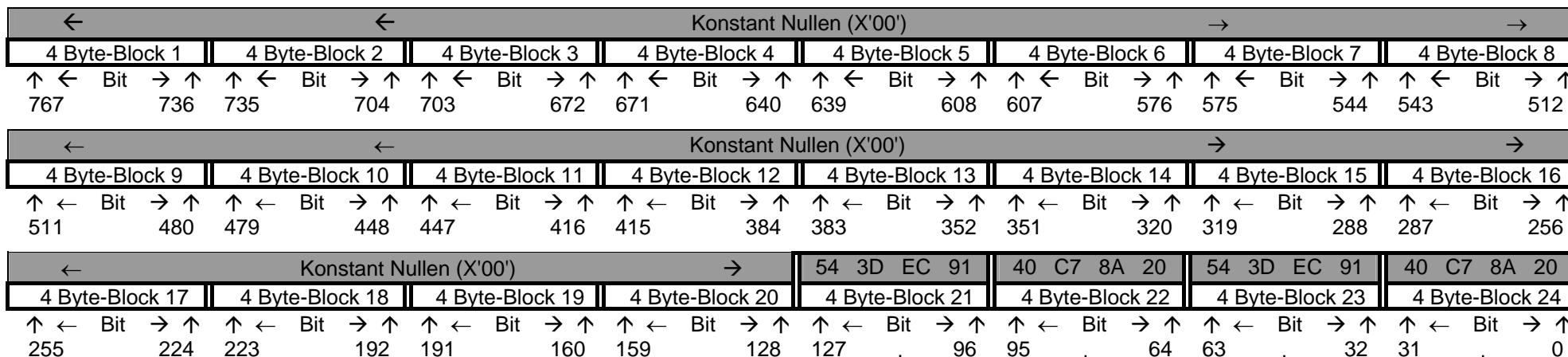
Bankseitiges Empfangen von Daten

- Empfang der V-Datei
- Entschlüsselung des verschlüsselten DES-Schlüssels aus dem Dateivorsatz mit VSB_{Bank}
- Entschlüsselung der Nachricht mit dem so erhaltenen DEK

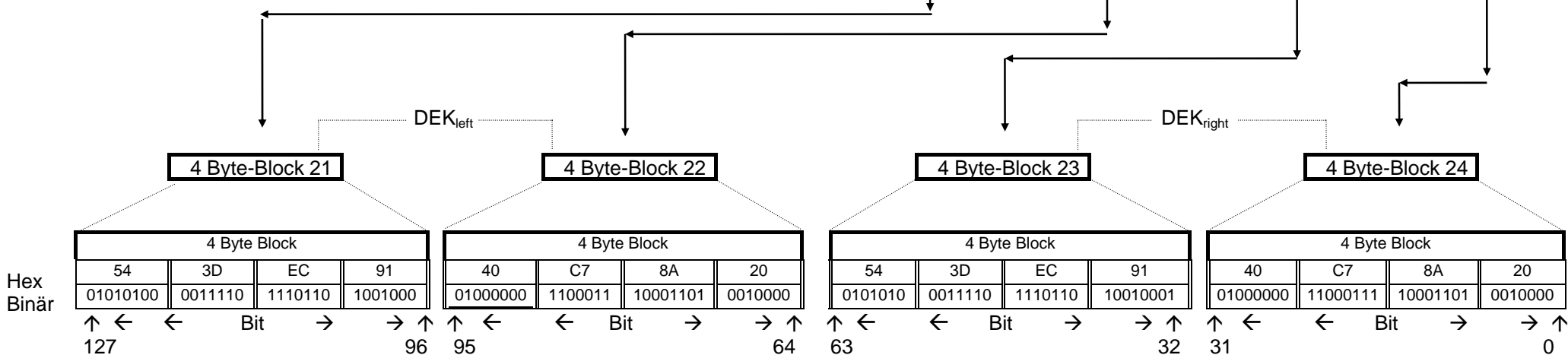
³⁸ Sofern eine DFÜ nicht erfolgreich durchgeführt werden konnte, wird die ggf. vorher bereits komprimierte und verschlüsselte Datei für den nächsten Start der DFÜ gespeichert, so dass ein erneuter Komprimierungs- und Verschlüsselungsvorgang nicht mehr erforderlich ist. Das System sollte erkennen, wenn nach einem erfolglosen DFÜ-Auftrag die Originaldatei gelöscht oder verändert wurde, so dass nicht das Komprimat einer früheren Originaldatei zu einer zwischenzeitlich geänderten Originaldatei verschickt wird.

2.3.6 Interne Datenformate der Funktion „Verschlüsselung“

2.3.6.1 PDEK und DES-Schlüssel (DEK)



ab hier DES-Schlüssel (DEK):



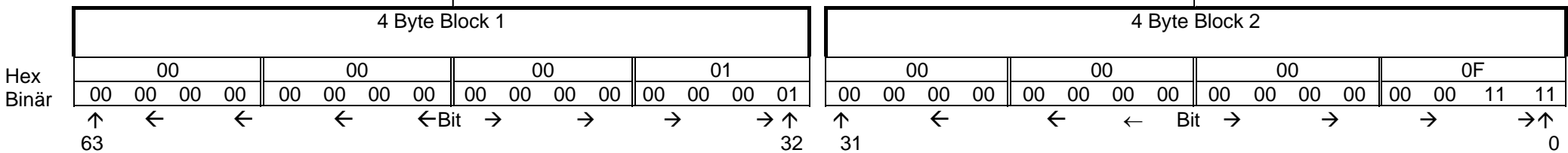
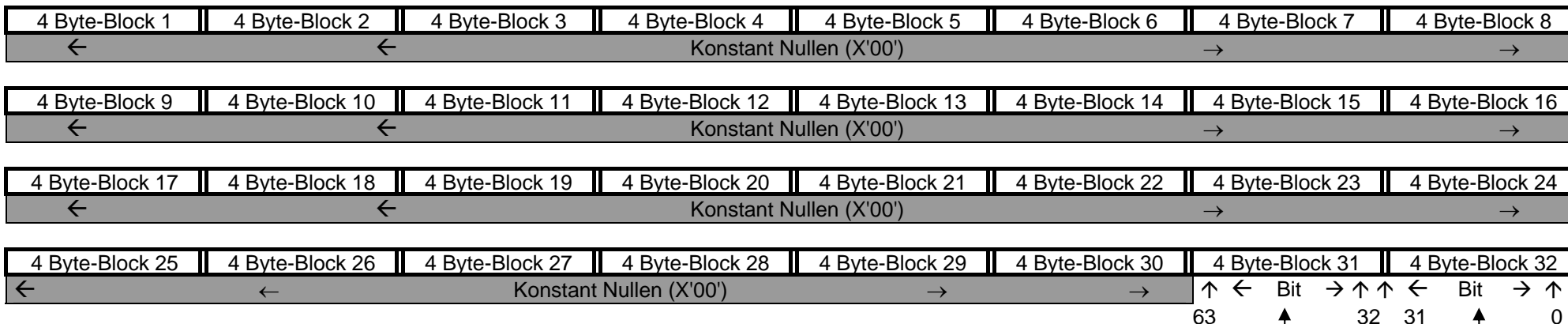
DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

2.3.6.2 EDEK

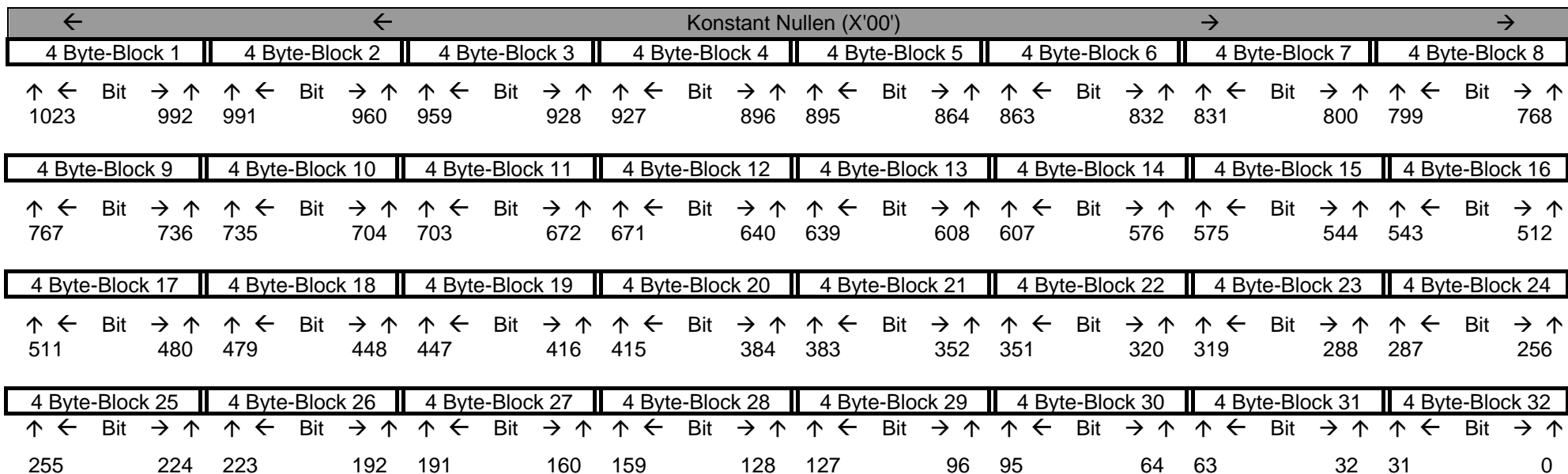
←		Konstant Nullen (X'00')																→	
4 Byte-Block 1		4 Byte-Block 2		4 Byte-Block 3		4 Byte-Block 4		4 Byte-Block 5		4 Byte-Block 6		4 Byte-Block 7		4 Byte-Block 8					
↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑				
1023	992	991	960	959	928	927	896	895	864	863	832	831	800	799	768				
4 Byte-Block 9		4 Byte-Block 10		4 Byte-Block 11		4 Byte-Block 12		4 Byte-Block 13		4 Byte-Block 14		4 Byte-Block 15		4 Byte-Block 16					
↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑					
767	736	735	704	703	672	671	640	639	608	607	576	575	544	543	512				
4 Byte-Block 17		4 Byte-Block 18		4 Byte-Block 19		4 Byte-Block 20		4 Byte-Block 21		4 Byte-Block 22		4 Byte-Block 23		4 Byte-Block 24					
↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑					
511	480	479	448	447	416	415	384	383	352	351	320	319	288	287	256				
4 Byte-Block 25		4 Byte-Block 26		4 Byte-Block 27		4 Byte-Block 28		4 Byte-Block 29		4 Byte-Block 30		4 Byte-Block 31		4 Byte-Block 32					
↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑	↑ ← Bit → ↑					
255	224	223	192	191	160	159	128	127	96	95	64	63	32	31	0				

2.3.6.3 Exponent



Hier wird beispielhaft der Exponent mit dem Wert 00 ... 01 00 00 00 0F (Hex) dargestellt. Gegebenenfalls können auch variable Werte für den Exponenten verwendet werden oder die 4. Fermat'sche Primzahl. Es ist sicherzustellen, dass nicht der schwache Exponent 3 verwendet wird.

2.3.6.4 Modulo



2.3.6.5 Testdaten Verschlüsselung³⁹

1. Testdatei: TEST.DAT – 24 Byte

2. Testdatei: 255.IZV – 640 Byte

Verschlüsselung: Beispiel 1

768-Bit RSA-Verschlüsselungsschlüssel

Modulus:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
cc 84 94 b2 1b ea 9b 0b e9 5d f3 17 09 34 8e 6e
31 64 1f ee 05 43 b5 96 43 75 50 d1 f3 d1 32 d6
5f 13 5b 84 90 e5 24 ae 87 b1 9b 3d 64 d6 33 bc
69 4c 7f fc 30 13 fd c6 ed f4 97 59 b1 4a 8a 53
7b 80 dc 2c 9d 2f 12 07 be 38 c9 fb 97 03 5e 38
5a e4 d8 5b 1d 73 e9 95 eb ce 0a c2 e5 39 a9 2b
```

Öffentlicher Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 0f
```

Geheimer Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
2e 3f 49 a9 69 fc 9c 1a fb 37 3f 42 8d 9c 05 cb
77 d5 86 ef 9c 2c 6e 4f 6f 79 c7 49 12 4a f2 0a
00 66 77 52 d5 80 2e 0e c3 57 3b 4e 5c f0 10 ce
cb 6f f1 08 85 cd ce 95 dc 90 2f 87 d1 91 6a 8c
8d 8b 61 f2 56 dd 70 29 58 94 81 6a d3 15 e2 68
2c b4 3c ea 78 3c 91 11 72 81 c9 a7 d5 86 8c ef
```

Hashwert (des öffentlichen Schlüssels):

```
eb c9 6e b1 1d 86 3b 12 66 30 99 f3 dc 44 b7 3f
```

³⁹ Die Dateien TEST.DAT und 255.IZV können über DFUE@SIZ.de angefordert werden.

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Verschlüsselung der 1. Testdatei:

DES-Key:

dc d3 6b e9 10 0b 26 bc f4 34 4a da d6 c1 b0 3d

Verschlüsselter DES-Key:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12 15 72 0c 73 c7 8f 7a d7 54 37 9b 37 9f fa 09
be 8c f0 db cc 6b ee 71 89 37 33 23 3d 35 cd 7f
db 4a 4c 8d f9 4e 1b 0a a3 b7 3b 07 3a 88 b4 da
1d 59 2b 2f 87 b4 d8 6f 30 2e 2f f5 10 c2 0b 7d
af a9 c8 b4 93 35 12 87 54 56 62 64 24 ce 13 da
c3 e1 13 3b b3 92 58 b2 d3 f9 a8 77 35 b9 a4 fd

Verschlüsselte 1. Testdatei:

56 30 30 31 30 32 35 36 41 31 42 31 43 31 44 31
41 33 42 33 43 33 44 33 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 12 15 72 0c 73 c7 8f 7a
d7 54 37 9b 37 9f fa 09 be 8c f0 db cc 6b ee 71
89 37 33 23 3d 35 cd 7f db 4a 4c 8d f9 4e 1b 0a
a3 b7 3b 07 3a 88 b4 da 1d 59 2b 2f 87 b4 d8 6f
30 2e 2f f5 10 c2 0b 7d af a9 c8 b4 93 35 12 87
54 56 62 64 24 ce 13 da c3 e1 13 3b b3 92 58 b2
d3 f9 a8 77 35 b9 a4 fd eb c9 6e b1 1d 86 3b 12
66 30 99 f3 dc 44 b7 3f 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
32 81 34 58 c1 a4 17 9d d2 c2 14 47 0f 11 7c 30
d1 80 be 67 ba ee fb 0d 30 ec 95 eb fa a5 2e 69

Verschlüsselung der 2. Testdatei:

DES-Key:

da f4 0e 97 86 d5 b0 f7 58 38 d3 94 5b 7a 8c c1

Verschlüsselter DES-Key:

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
b4 62 3d 43 27 90 c7 41 9f 91 0f 35 70 f7 14 1b
ea 23 91 e1 32 86 93 91 4d 72 6e 1d 14 6d 30 b7
10 ee 24 cd d1 d5 e5 14 c0 01 bb 54 11 a9 b5 46
93 c6 11 df 86 48 e2 20 66 0e ea 95 4f f9 87 d1
ff ac 01 bf a6 99 42 11 8b da 9e da f5 6f ad e3
40 a8 c1 db 8f 9f 6d 52 17 89 ce 8a 94 27 1c 9b

Verschlüsselte der 2. Testdatei:

56 30 30 31 30 32 35 36 41 31 42 31 43 31 44 31
41 33 42 33 43 33 44 33 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 b4 62 3d 43 27 90 c7 41
9f 91 0f 35 70 f7 14 1b ea 23 91 e1 32 86 93 91
4d 72 6e 1d 14 6d 30 b7 10 ee 24 cd d1 d5 e5 14
c0 01 bb 54 11 a9 b5 46 93 c6 11 df 86 48 e2 20
66 0e ea 95 4f f9 87 d1 ff ac 01 bf a6 99 42 11
8b da 9e da f5 6f ad e3 40 a8 c1 db 8f 9f 6d 52

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

17 89 ce 8a 94 27 1c 9b eb c9 6e b1 1d 86 3b 12
66 30 99 f3 dc 44 b7 3f 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20
71 08 b4 53 3c 78 7a 37 e5 9f 9c f9 16 dd 99 16
e6 cd fa 91 ba 16 7a bd 69 b2 51 6e ff 32 57 4a
71 0a 61 a0 79 92 ae 6d f0 0a d0 b6 81 41 16 31
18 ab a9 45 91 81 26 f7 44 a3 e7 27 37 cf 03 51
68 b6 b9 e9 b0 96 86 57 55 99 2c c9 62 7c d8 2b
81 17 e2 bf ba ef 5e 97 8c 67 e0 b0 f8 ac 82 56
63 70 c4 f9 94 19 26 64 68 ff 36 0f bf f4 55 c5
7c 4e d0 23 b1 31 ea c4 e3 32 70 85 3e ff e1 73
ff 46 a3 f9 6f c3 74 3f a0 b6 87 d2 ab 33 3e 1d
4b 38 7b 87 eb 19 9c e7 e7 a3 d9 6a db b5 29 a8
88 88 b6 df 32 57 41 d7 b1 38 fa 86 ad d2 fd 40
a8 10 8d f5 18 f2 8b 11 bd c7 3d e8 af 90 9b ed
1c e2 c6 6a 88 d3 52 de a5 c8 18 1e ce dd 17 f4
c1 c0 75 f0 cf 55 d8 a4 66 cb 20 ea b4 3d 0f 77
65 16 10 d6 ae 14 8f 21 94 f7 41 e4 7a e7 00 df
3f 7c 01 04 e3 00 38 65 af ba 88 5a 7e 39 1c 53
db b2 19 f5 55 11 53 d2 d9 03 5a 13 11 65 31 0b
da 60 0b dd 86 7f ba 70 79 c4 26 ac a7 84 dd f9
e2 8a ca a2 fe 4e b5 1c 35 2e e8 43 22 ed 8f ed
97 86 ef 2f 3f 2e 49 f7 95 50 3a d5 10 00 e0 10
2d a7 08 fa 57 9f b7 7c cf 60 0a 46 86 c9 05 e7
76 1e 1b 55 20 73 5b 4a 8a 8d 40 4e 89 38 55 60
36 b2 2e 60 d1 12 7b f8 1b 9b 55 ca e0 af 42 c1
1a ba bc b7 9c 7c 10 da fc 8d 75 91 90 88 da 00
bf c6 10 94 7e 3f db c2 5e 3e 17 38 3f e0 68 15
67 f9 6e d0 c3 4f 1e 6e 90 cc 15 50 64 ef 5f 9e
e4 3f 65 0b 65 12 78 33 35 e9 43 98 f7 36 f1 36
4e 2a 13 0b 20 a9 bc ac 09 b2 5d 7c 72 29 ec f4
9d 4c aa 90 fa a3 e7 86 67 84 57 8f 50 8b 41 66
82 d0 ed 29 89 54 f2 ef c9 75 48 5d 38 b0 30 e0
b3 98 36 84 1e 83 32 6b a1 74 9d d0 3f 83 4e e3
2a 22 f1 d1 c5 26 4b 06 b2 39 18 3e 00 2d ca 25
7c 53 47 90 1c 13 37 92 64 6c a0 27 1c 1b 2f 5b
97 82 e4 77 74 24 89 b8 a1 08 f4 4d 7d 6c 43 b0
85 aa 8f d9 06 8e 91 58 34 d7 59 0e e9 82 b6 99
18 1e 9c 8a 6c 9c f9 35 a8 9f 0a b7 7c 2f 6e 3b
63 78 27 f0 db 72 b5 3e 4c f1 03 57 56 5d 97 bf
96 7f 4f 21 07 c5 44 f2 a5 de b1 de 1c 84 e0 86
ae 33 14 49 ce 03 cf 67 16 94 99 cd 2b 9c a9 3d
ad 9e 50 9b 72 b0 89 13 42 48 77 28 1b 09 6b a2
27 5a 01 c3 8a 83 66 bd

Verschlüsselung: Beispiel 2

768-Bit RSA-Verschlüsselungsschlüssel

Modulus:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
bb 7f 81 9c 78 14 00 df 38 fc f6 22 f1 fc 7f 71
ff fb ec ce 64 f1 3e c4 66 5b 31 85 81 9e 61 74
55 9a 88 48 a9 fb 3e 06 2b 16 73 91 a0 6b aa b0
1d 95 8b ec 5c db 89 0f 75 10 5a 65 e0 3d 37 98
f8 32 51 56 e0 c5 d0 dc c1 61 6d df 62 b7 c2 77
e5 6b a6 13 66 93 46 45 b9 b9 8b 70 be 41 da 37
```

Öffentlicher Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 0f
```

Geheimer Exponent:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
53 ef ab 44 24 4b 4e 6e 5c 58 ef 8b eb 6a 1d cf
e2 75 19 07 8d e8 12 8e d1 36 fe 29 e5 c1 d3 a7
6b a5 69 4d 35 f7 b2 1e a1 dd 26 95 ae 4a a8 ea
47 31 ce 4e cc da 43 8a 25 87 dc de 47 60 de 27
88 60 2d 73 85 48 be dd cf ba b6 39 4e cd 81 cc
56 f0 d2 02 88 82 cb 12 b6 d5 ad 67 f5 a9 83 6f
```

Hashwert (des öffentlichen Schlüssels):

```
9e 7c 62 75 16 92 04 c8 b2 06 8a bb 79 1c 97 83
```

Verschlüsselung der 1. Testdatei:

DES-Key:

```
26 08 e0 ec 62 fb 62 b3 29 73 19 9b 49 b6 e3 e3
```

Verschlüsselter DES-Key:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18 eb 82 2c 99 75 71 f7 b1 9d f0 16 98 19 ea fe
0a fe 69 2a 0a 51 59 07 d1 e2 a9 64 bf 36 3d 72
0b 5b 50 8c 2b 1f 0b 17 d2 6e 99 76 71 90 74 43
c2 d5 5a 3b a5 8f 55 1a 64 e6 ac 51 7a 59 3b 07
c9 3e 35 e9 fe 8e 48 b8 a2 95 b4 ee 18 c0 32 50
b8 4a 45 e1 33 87 fb ef fe 0d 3f 5a ec 14 61 c4
```

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

Verschlüsselte 1. Testdatei:

```
56 30 30 31 30 32 35 36 41 31 42 31 43 31 44 31
41 33 42 33 43 33 44 33 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 18 eb 82 2c 99 75 71 f7
b1 9d f0 16 98 19 ea fe 0a fe 69 2a 0a 51 59 07
d1 e2 a9 64 bf 36 3d 72 0b 5b 50 8c 2b 1f 0b 17
d2 6e 99 76 71 90 74 43 c2 d5 5a 3b a5 8f 55 1a
64 e6 ac 51 7a 59 3b 07 c9 3e 35 e9 fe 8e 48 b8
a2 95 b4 ee 18 c0 32 50 b8 4a 45 e1 33 87 fb ef
fe 0d 3f 5a ec 14 61 c4 9e 7c 62 75 16 92 04 c8
b2 06 8a bb 79 1c 97 83 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
e9 44 14 c3 9d 91 ca 5c 50 a9 e9 03 7b 7e aa c4
cd c5 a3 cf 03 ec 90 f7 a2 8e 28 b6 20 7a aa 12
```

Verschlüsselung der 2. Testdatei:

DES-Key:

```
62 e9 37 4c 8f 61 9e ea 9d 6d 62 f1 16 92 6b 91
```

Verschlüsselter DES-Key:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 9c 34 3e f3 bd 3d 59 0d 08 d5 07 ce 42 67 c9
52 df e0 ec 68 18 98 05 08 ec 3a 49 a8 45 63 99
61 49 e0 34 2b fc 69 ce ca 83 95 20 d5 9d 7c a8
3c f9 31 cf 8d 60 9a 14 b8 56 35 f5 f5 4c 13 ac
3b 94 54 cc c8 97 19 96 98 61 99 9d 60 06 8b a9
cd 9b 52 a6 a0 75 5f db 43 d6 a6 3e 8b c7 60 df
```

Verschlüsselte der 2. Testdatei:

```
56 30 30 31 30 32 35 36 41 31 42 31 43 31 44 31
41 33 42 33 43 33 44 33 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 10 9c 34 3e f3 bd 3d 59
0d 08 d5 07 ce 42 67 c9 52 df e0 ec 68 18 98 05
08 ec 3a 49 a8 45 63 99 61 49 e0 34 2b fc 69 ce
ca 83 95 20 d5 9d 7c a8 3c f9 31 cf 8d 60 9a 14
b8 56 35 f5 f5 4c 13 ac 3b 94 54 cc c8 97 19 96
98 61 99 9d 60 06 8b a9 cd 9b 52 a6 a0 75 5f db
43 d6 a6 3e 8b c7 60 df 9e 7c 62 75 16 92 04 c8
b2 06 8a bb 79 1c 97 83 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
5b c0 51 26 b4 60 ca a9 3a 80 e3 15 13 2e 10 a7
a0 59 62 3d 50 0e 9b d9 ed c9 c1 66 b4 01 61 a4
05 09 fe 65 68 fa 61 ee 40 50 20 45 ab 09 2a d4
df 4a 63 56 e9 08 05 e5 57 f4 76 e5 e4 ea 43 3e
80 38 98 97 7c 00 40 20 e7 5d 68 16 0a 6e 93 9b
d7 01 55 54 a4 62 19 ed fc 7c 48 03 f0 c6 23 e1
18 4d 7c c7 6f c8 91 dc 19 2b 43 2e 68 f1 b3 47
```

DFÜ – Abkommen

Anlage 2: Spezifikation für die FTAM-Anbindung

a0 12 e1 71 08 95 94 ec d5 23 6b b5 90 da d3 12
1e 53 31 bd 98 5d 6f 68 be 21 d0 2d dc 9d dd 52
35 94 dd b5 79 25 94 79 54 2b b9 ba b1 af f0 ff
3f 93 c3 cc e0 f2 08 0e 70 96 2a 0f a4 40 42 5d
86 6d 4f 69 f4 f4 cd f4 c5 ab 1b db 66 d8 6c 5e
c3 f2 b7 4d 91 b9 14 c4 2d 92 a0 54 33 ee fe d5
13 88 1c a4 8d 73 de 53 6b 1c 3b 00 85 3d 10 af
fd c7 82 f6 c0 3b 41 c9 9a e4 8f 79 a5 43 00 25
4b 86 47 e1 c5 93 23 2f bb 90 b8 71 24 f5 86 4d
de 82 d6 2b 87 48 6d f3 dc d4 40 9f d7 6f 0e 72
8b 16 ce 76 55 1b 88 67 8f d6 11 49 6a 2a 7e c2
ac e3 75 c4 b3 60 5e d4 33 69 96 2b 38 86 06 db
ef f8 0c b4 60 d5 68 dc 3e 08 b6 b2 53 32 fa 27
57 8f 50 8e 5e 37 5a 08 63 6b c8 26 05 29 37 c6
9f 17 f5 75 42 84 7f 00 a7 8f f0 18 85 d2 ee 00
d7 61 e3 b8 03 e5 b5 7e 87 3e 27 fa da 5a 92 0f
f1 10 5f 12 8f 17 1c a6 f4 c0 7f 8b 16 fd a7 cc
9d 40 ba 8d 91 78 37 47 82 51 90 d1 06 3c eb 66
5d 55 85 85 4b 7b 8a e3 f3 9d 04 9a 68 8a 2f 2d
03 ac a3 69 65 ad cf 18 ac f7 bc 07 13 e0 b9 71
f0 49 52 d4 0e ff 32 b8 0a 61 0e a2 18 59 07 5d
87 98 fc 04 7a cf 1d 93 9e 11 44 8f 02 30 74 2d
21 cb 87 ed 51 d5 a7 28 4d bf d6 65 6a a9 af 6c
4b 9a bf 4b 61 b0 d2 3e 15 73 f0 ee 8e 0a 5b c3
2e 36 41 44 9c 66 d5 48 33 3e 0b 40 2f 58 30 7c
b0 c8 49 32 38 e2 7c cf cf 51 6d 86 04 c9 b1 b2
e2 1c f9 90 6a 14 63 3d 00 f3 8b 54 95 8a 8b 0f
52 82 26 5c b7 fb 7b 5b f3 1c 10 21 8c f0 b8 87
be 0a 6f 05 70 f7 ae 5a 9c 6e 3a 2f de 33 7d 61
e8 ba 4c 1c 3b 72 1d 5c 00 9a 76 e4 21 21 4a 18
e9 98 18 f5 ab a6 af 34 42 4f 32 cd 7d d9 42 49
73 ba 93 a8 9c d1 87 68 9b c1 89 7f a5 60 89 80
90 61 f1 20 b5 8e 2b 98 86 6d 49 53 73 5c 4b 94
ce fd 3a 27 1c 8a c9 27

2.3.7 Liste der reservierten Antwortcodes

Antwortcode	Bemerkung
50	Aktion erfolgreich - Neue Bankparameterdaten abholen Das Banksystem hat neue Bankparameterdaten zur Verfügung gestellt. Das Kundensystem generiert automatisch einen BPD-Abholauftrag und informiert den Anwender, dass der Abholauftrag zu starten ist.
51	Verschlüsselungscode mit Bank muss aktualisiert werden. Auf dem Banksystem wurde ein neuer Verschlüsselungs-Public-Key generiert, der vom Kundensystem abgeholt werden muss. Der Anwender wird hierüber durch eine entsprechende Meldung hingewiesen. Solange der neue Verschlüsselungs-Public-Key bei der Bank noch nicht abgeholt wurde, kann der Kunde keine (verschlüsselten) Dateien an die Bank übermitteln. Der Antwortcode 51 kann auch dann vom Banksystem zurückgeliefert werden, wenn der im FTAM-Remote-Filename der verschlüsselten Datei eingestellte Hashwert des verwendeten Verschlüsselungs-Public-Keys der Bank (VPB) nicht mit dem bankseitig erwarteten Hashwert übereinstimmt. In diesem Fall muss der Kunde ebenfalls einen neuen Verschlüsselungs-Public-Key bei der Bank abholen.
52	Daten müssen verschlüsselt abgeholt werden Diese Meldung kann nur auftreten, wenn der Kunde versucht, trotz einer aktivierten bankseitigen Verschlüsselung Daten unverschlüsselt abzuholen.
53	Daten müssen unverschlüsselt abgeholt werden. Diese Meldung kann nur auftreten, wenn der Kunde versucht, Daten verschlüsselt abzuholen, obwohl diese bankseitig unverschlüsselt bereitgestellt werden.
54	Verschlüsselungscode muss neu verschickt werden. Kundenseitig wird nach erfolgreichem Versand des VPK-Auftrages und des dazugehörigen Legitimationsauftrages zunächst davon ausgegangen, dass die entsprechende Authentifizierung bei der Bank mit positivem Ergebnis durchgeführt wurde. Verläuft die bankseitige Unterschriftsprüfung der VPK-Datei jedoch negativ, so erhält das Kundensystem beim Abholversuch von verschlüsselt vereinbarten Auftragsarten den obigen Antwortcode 54.
55	User nicht EU-berechtigt Diese Meldung wird vom Banksystem zurückgegeben, wenn beim Versand einer VPK-Datei mit Elektronischer Unterschrift festgestellt wird, dass der Unterzeichner bei der Bank keine Unterschriftsberechtigung besitzt. Dies kann bereits vor der Übertragung der Datei selbst ermittelt werden, da die User-ID des Unterzeichnenden im FTAM-Remote-Filename übergeben wird.
56	Verschlüsselungscode noch nicht freigegeben. Solange der kundenseitige Verschlüsselungs-Public-Key bankseitig noch nicht authentifiziert wurde, erhält der Kunde bei dem Versuch, Daten mit den Auftragsarten abzuholen, für die die Verschlüsselung mit der Bank vereinbart wurde, den Returncode 56 „Verschlüsselungscode noch nicht freigegeben“.

2.3.8 Abkürzungsverzeichnis

Abkürzung	Bedeutung
CBC	Cipher Block Chaining
DEK	geheimer DES-Schlüssel (Data Encryption Key)
DES	Data Encryption Standard
DFP	Digital Fingerprint
EDEK	verschlüsselter DEK (Encrypted Data Encryption Key)
FP	Fingerprint
HASH	Hashwert
ICV	Initial Chaining Value
n	Modulus eines RSA-Schlüsselsystems
N	Länge von n in Anzahl der Bit
PDEK	Padded Data Encryption Key
SIGNATUR	Signatur einer Nachricht
THASH	um den Zeitstempel ergänzter Hashwert (Time stamped HASH)
TVP	Zeitstempel (Time Variant Parameter)
VPB	Verschlüsselungs-Public-Key-Bank
VPK	Verschlüsselungs-Public-Key-Kunde
VSB	Verschlüsselungs-Secret-Key-Bank
VSK	Verschlüsselungs-Secret-Key-Kunde