
FinTS

Financial Transaction Services

- Schnittstellenspezifikation -

Security

Sicherheitsverfahren HBCI

Herausgeber:

Bundesverband deutscher Banken e.V., Berlin

Deutscher Sparkassen- und Giroverband e.V., Bonn/Berlin

Bundesverband der Deutschen Volksbanken und Raiffeisenbanken e.V., Berlin

Bundesverband Öffentlicher Banken Deutschlands e.V., Berlin

Version: 4.0

Stand: 09.07.2004

Die vorliegende Schnittstellenspezifikation für eine automatisiert nutzbare multibankfähige Banking-Schnittstelle (im Folgenden: Schnittstellenspezifikation) wurde im Auftrag des Zentralen Kreditausschusses entwickelt. Sie wird hiermit zur Implementation in Kunden- und Kreditinstitutssysteme freigegeben.

Die Schnittstellenspezifikation ist urheberrechtlich geschützt. Zur Implementation in Kunden- und Kreditinstitutssysteme wird interessierten Herstellern unentgeltlich ein einfaches Nutzungsrecht eingeräumt. Im Rahmen des genannten Zwecks darf die Schnittstellenspezifikation auch - in unveränderter Form - vervielfältigt und zu den nachstehenden Bedingungen verbreitet werden.

Umgestaltungen, Bearbeitungen, Übersetzungen und jegliche Änderung der Schnittstellenspezifikation sind untersagt. Kennzeichnungen, Copyright-Vermerke und Eigentumsangaben dürfen in keinem Fall geändert werden.

Im Hinblick auf die Unentgeltlichkeit des eingeräumten Nutzungsrechts wird keinerlei Gewährleistung oder Haftung für Fehler der Schnittstellenspezifikation oder die ordnungsgemäße Funktion der auf ihr beruhenden Produkte übernommen. Die Hersteller sind aufgefordert, Fehler oder Auslegungsspielräume der Spezifikation, die die ordnungsgemäße Funktion oder Multibankfähigkeit von Kundenprodukten behindern, dem Zentralen Kreditausschuss zu melden. Es wird weiterhin ausdrücklich darauf hingewiesen, dass Änderungen der Schnittstellenspezifikation durch den Zentralen Kreditausschuss jederzeit und ohne vorherige Ankündigung möglich sind.

Eine Weitergabe der Schnittstellenspezifikation durch den Hersteller an Dritte darf nur unentgeltlich, in unveränderter Form und zu den vorstehenden Bedingungen erfolgen.

Dieses Dokument kann im Internet abgerufen werden unter <http://www.fints.org>.

Versionsführung

Das vorliegende Dokument wurde von folgenden Personen erstellt bzw. geändert:

Name	Organisation	Datum	Version	Dokumente	Anmerkungen
Stein	SIZ	24.10.2002	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI.doc	Frühere Versionen wurden im Rahmen der HBCI-Spezifikation veröffentlicht
	SIZ	02.05.2003	4.0ref	FinTS_4.0_Security_ HBCI.doc	Anpassungen für FinTS 4.0
	SIZ	19.11.2003	4.0 final draft 01	FinTS_4.0_Security_ HBCI.doc	Überarbeitungen
	SIZ	02.04.2004	4.0 final draft 02	FinTS_4.0_Security_ HBCI.doc	Überarbeitungen
	SIZ	09.07.2004	4.0	FinTS_4.0_Security_ HBCI.doc	Überarbeitungen

Änderungen gegenüber der Vorversion

Änderungen sind im Dokument durch einen Randbalken markiert. Hypertextlinks sind in dieser [Farbe](#) markiert. Falls sich die Kapitelnummerierung geändert hat, bezieht sich die Kapitelangabe auf die neue Nummerierung.

Ifd. Nr.	Kapitel	Kapitelnummer	Ken-nung ¹	Art ²	Beschreibung
1	Verfahrensbeschreibung	II.1.1		F	Korrektur des Paddingverfahrens bei RDH-3.
2		II.3.1.3.3, II.6.1, II.6.1.1		K	Profilwechsel bei Schlüsseländerung.
3		II.6.1.1, II.3.1.3.3		K	Schlüsseländerung für maximal zwei Schlüssel.
4		II.6.1.2 a)		K	Verwendung von Schlüsselnummer und -Version.
5		II.6.1.2 a)		E	Belegung Typ-Feld.
6		II.6.1.3 b)		E	Code 3330 in Beispiele für Rückmeldungen hinzu.
7	Chipapplikationen	III.1.1.2		E	Erläuterung zu EF_NOTEPAD.C4.
8		III.1.2.1		F	Keine Addition der Sequenzzähler.
9		III.1.1.2.3, III.2.1.8, III.2.2.8		E	Verweis auf gültige Codierung für Kommunikationsdienst hinzu.

¹ nur zur internen Zuordnung

² F = Fehler; Ä = Änderung; K = Klarstellung; E = Erweiterung

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Inhaltsverzeichnis	Stand: 09.07.2004	Seite: 1

Inhaltsverzeichnis

I.	Einleitung	1
II.	VERFAHRENSBESCHREIBUNG	2
II.1	Allgemeines.....	2
II.1.1	Sicherheitsprofile	2
II.1.2	Sicherheitsklassen	7
II.2	Mechanismen	8
II.2.1	Elektronische Signatur	8
II.2.1.1	Hashing	8
II.2.1.2	Elektronische Signatur bei DDV (DES-basierend)	8
II.2.1.3	Elektronische Signatur bei RDH (RSA-basierend)	9
II.2.2	Verschlüsselung.....	11
II.2.2.1	Verschlüsselung bei DDV (DES-basierend)	14
II.2.2.2	Verschlüsselung bei RDH (RSA-basierend)	15
II.2.3	Komprimierung.....	16
II.2.4	Sicherheitsmedien beim Kundensystem	16
II.3	Abläufe.....	18
II.3.1	Schlüsselverwaltung	18
II.3.1.1	Gemeinsam verwendete Verfahren zur Schlüsselverwaltung	18
II.3.1.2	Symmetrische Schlüssel für DDV	20
II.3.1.2.1	Schlüsselgenerierung	20
II.3.1.2.2	Initiale Schlüsselverteilung	21
II.3.1.2.3	Schlüsseländerungen	21
II.3.1.2.4	Schlüsselverteilung nach Kompromittierung	21
II.3.1.3	Asymmetrische Schlüssel für RDH.....	22
II.3.1.3.1	Schlüsselgenerierung	22
II.3.1.3.2	Initiale Schlüsselverteilung	23
II.3.1.3.3	Schlüsseländerungen	27
II.3.1.3.4	Schlüsselverteilung nach Kompromittierung	29
II.3.2	Schlüsselsperrung	30
II.4	Bankfachliche Anforderungen.....	32
II.5	Formate für Signatur und Verschlüsselung	33
II.5.1	Signatur-Segment	34
II.5.2	Verschlüsselungsdaten	37
II.5.3	Komprimierungsdaten	38
II.6	Key-Management	39
II.6.1	Key-Management-Nachrichten	39

Kapitel:	I	Version:	4.0	Financial Transaction Services (FinTS)
				Dokument: Security - Sicherheitsverfahren HBCI
Seite:	2	Stand:	09.07.2004	Kapitel: Inhaltsverzeichnis

II.6.1.1	Änderung eines öffentlichen Schlüssels des Benutzers.....	40
II.6.1.2	Erstmalige Anforderung der Schlüssel des Kreditinstituts.....	42
II.6.1.3	Erstmalige Übermittlung der Schlüssel des Benutzers	45
II.6.1.4	Schlüsselsperrung durch den Benutzer.....	48

III. CHIPAPPLIKATIONEN51

III.1 Chipapplikation für RDH51

III.1.1	Applikation Notepad.....	51
III.1.1.1	Daten der Applikation Notepad.....	53
III.1.1.1.1	ADF der Applikation Notepad	54
III.1.1.1.2	EF_RULE.....	56
III.1.1.1.3	EF_KEY	60
III.1.1.1.4	EF_KEYD.....	61
III.1.1.1.5	EF_NOTEPAD	63
III.1.1.2	Recordbelegung des EF_NOTEPAD mit einem FinTS- Parameterblock.....	66
III.1.1.2.1	Tag 'F0': FinTS-Parameterblock.....	68
III.1.1.2.2	Tag 'E1': FinTS-Institutparameterblock	68
III.1.1.2.3	Tag 'E2':FinTS - Kommunikationsparameterblock	70
III.1.1.2.4	Tag 'E3':FinTS-Benutzerparameterblock	70
III.1.2	Terminalabläufe	71
III.1.2.1	Verfahren zur Ermittlung der Sequenzzähler	71
III.1.2.2	Beschreibung der Terminalabläufe.....	72
III.1.2.2.1	Signatur einleiten	73
III.1.2.2.2	Nachrichten generieren	75
III.1.2.2.3	Nachrichten signieren	75
III.1.2.2.4	Nachrichten verschlüsseln.....	76
III.1.2.2.5	FinTS-Dialog führen.....	78
III.1.2.2.6	Signatur beenden.....	79
III.1.2.3	Makros	80
III.1.2.3.1	Signatur-Berechnung.....	80
III.1.2.3.2	Signatur-Prüfung.....	81

III.2 Chipapplikation für DDV.....83

III.2.1	Daten der Applikation FinTS-Banking für Typ 0.....	84
III.2.1.1	DF_Banking	85
III.2.1.2	EF_KEY	87
III.2.1.3	EF_KEYD.....	88
III.2.1.4	EF_AUT	89
III.2.1.5	EF_AUTD.....	90
III.2.1.6	EF_PWD1	91
III.2.1.7	EF_PWDD1	93
III.2.1.8	EF_BNK	94
III.2.1.9	EF_MAC	96
III.2.1.10	EF_SEQ.....	97
III.2.2	Daten der Applikation FinTS-Banking für Typ 1.....	98

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Inhaltsverzeichnis	Stand: 09.07.2004	Seite: 3

III.2.2.1	ADF der Applikation FinTS-Banking	99
III.2.2.2	EF_RULE	101
III.2.2.3	EF_KEY	105
III.2.2.4	EF_KEYD	107
III.2.2.5	EF_PWD	109
III.2.2.6	EF_PWDD	111
III.2.2.7	EF_FBZ	112
III.2.2.8	EF_BNK	113
III.2.2.9	EF_MAC	114
III.2.2.10	EF_SEQ	115
III.2.3	Platzbedarf der Applikation im Chip	116
III.2.4	Terminalabläufe (Typ 0 und Typ 1)	118
III.2.4.1	Startdialog	118
III.2.4.2	Nachricht generieren	121
III.2.4.3	Nachricht signieren	123
III.2.4.4	Nachricht verschlüsseln	126
III.2.4.5	FinTS-Dialog führen	129
III.2.5	Makros	130
III.2.5.1	MAC-Berechnung / Prüfung	130
III.2.5.2	Entschlüsselung	133
III.2.6	Übersicht der Chip-Applikations-Parameter (Typ 1)	134

Kapitel:	I	Version:	4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite:	4	Stand:	09.07.2004	Kapitel: Abbildungsverzeichnis

Abbildungsverzeichnis

Abbildung 1: 2-Key-Triple-DES im CBC-Mode.....	12
Abbildung 2: Verschlüsselung bei 2-Key-Triple-DES	12
Abbildung 3: Entschlüsselung bei 2-Key-Triple-DES	13
Abbildung 4: 2-Key-Triple-DES im ECB-Mode	14
Abbildung 5: Verschlüsselung bei RDH-1	15
Abbildung 6: Verschlüsselung bei RDH-2	15
Abbildung 7: Verschlüsselung bei RDH-3 und RDH-4	16
Abbildung 8: Ablauf der Erstinitialisierung bei RDH	26
Abbildung 9: Beispiel für die Gestaltung des Ini-Briefs	27
Abbildung 10: Datenelemente der Applikation „Notepad“	53
Abbildung 11: Datenelemente der Applikation „Banking“	84
Abbildung 12: Datenelemente der Applikation „FinTS“, kontobezogene Karte	98
Abbildung 13: Datenelemente der Applikation „FinTS“, kontoungebundene Karte	98

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Abkürzungen	Stand: 09.07.2004	Seite: 5

Abkürzungen

Abkürzung	Bedeutung
AC	Access Condition
AEF	Application Elementary File
AID	Application Identifier
BPD	Bankparameterdaten
C	Datenstruktur ist konditional
CBC	Cipher Block Chaining
CID	Cardholders Information Data (Kartenidentifikationsdaten der ZKA-Chipkarte)
CLA	Class Byte
CR	Carriage-Return (Wagenrücklauf)
DDV	DES-DES-Verfahren
DE	Datenelement
DEG	Datenelementgruppe
DES	Data Encryption Standard
DF	Dedicated File
DFÜ	Synonym verwendet für "Datenkommunikation, die in Form von Filetransfer, E-Mail, Online-Nachrichtenaustausch etc. erfolgen kann"
ECB	Electronic Code Book
EF	Elementary File
EU	Elektronische Unterschrift; basiert auf dem asymmetrischen RSA-Verfahren
FCI	File Control Information
FCP	File Control Parameters
FCS	Frame Check Sequence
FinTS	Financial Transaction Services
FMD	File Management Data
GD	Gruppendatenelement
GDG	Gruppendatenelementgruppe
HBCI	Homebanking Computer Interface
I	Information (z. B. Schlüsselart)
ID	Identifikationsmerkmal (Nummer oder alphanumerischer Code)
ISO	International Organisation for Standardisation
IV	Initialisierungsvektor
KGK	Key Generating Key
LF	Line-Feed (neue Zeile)
M	Datenstruktur muss vorhanden sein und ist inhaltlich korrekt zu füllen

Kapitel:	I	Version:	4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite:	6	Stand:	09.07.2004	Kapitel: Abkürzungen

Abkürzung	Bedeutung
MAC	Message Authentication Code; Symmetrisches Verfahren zur Erzeugung einer elektronischen Signatur (derzeit für die ZKA-Chipkarte eingesetzt)
MF	Master File
MFC	Multifunktions-Chipkarte
MIME	Multipurpose Internet Mail Extensions
N	Nachricht
N	Nicht erlaubt (not allowed) (Datenstruktur darf nicht vorhanden sein)
O	Datenstruktur ist optional
PKD	Public-Key-Daten
RDH	RSA-DES-Hybridverfahren
RFC	Request for Comment
RSA	Asymmetrischer Algorithmus für die elektronische Unterschrift (EU) (vgl. MAC), benannt nach den Erfindern Rivest, Shamir und Adleman.
SEG	Segment
SEQ	Sequenznummer
SF	Segmentfolge
SFI	Short File Identifier
SSL	Secure Socket Layer
T	Transaktion (z. B. Schlüsselart)
UN/EDIFACT	s. EDIFACT
UPD	User-Parameterdaten
ZKA	Zentraler Kreditausschuss

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Literaturhinweise	Stand: 09.07.2004	Seite: 7

Literaturhinweise

♦ Allgemeines

- [Formals] Financial Transaction Services (FinTS) – Formals (Allgemeine Festlegungen für multibankfähige Online-Verfahren der deutschen Kreditwirtschaft), Version 4.0 final draft 02, 02.04.2004, Zentraler Kreditausschuss
- [Syntax] Financial Transaction Services (FinTS) – XML-Syntax, Version 4.0 final draft 02, 02.04.2004, Zentraler Kreditausschuss
- [ISO 3166] ISO 3166-1:1996: Code for the representation of names of countries and their subdivisions - Part 1: Country code (<http://www.din.de/gremien/nas/nabd/iso3166ma/> oder <http://www.unece.org/trade/locode/loc99.zip>)

♦ Verfahrensbeschreibung

- [SigG] Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften v. 16. Mai 2001, Bundesgesetzblatt Jahrgang 2001, Teil I Nr. 22
- [SigV] Verordnung zur elektronischen Signatur v. 16. November 2001, Bundesgesetzblatt Jahrgang 2001, Teil I Nr. 59
- [EU-Richtlinie] Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen, Amtsblatt der Europäischen Gemeinschaften v. 19.01.2000
- [DFÜ-Abkommen] Kryptographische Verfahren des deutschen Kreditgewerbes für die Elektronische Unterschrift und für die Verschlüsselung im Rahmen der Kunde-Bank-Kommunikation in: ZKA-Abkommen über die Datenfernübertragung zwischen Kunden und Kreditinstituten (DFÜ-Abkommen) v. 15.03.1995, Anlage 2
- [ISO 9796] ISO 9796:1991: Information technology - Security techniques - Digital signature scheme giving message recovery
- [ISO 9796-1] ISO 9796-1:1999 Information technology - Security techniques - Digital signature scheme giving message recovery – Part 1: Mechanisms using redundancy
- [ISO 9796-2] ISO 9796-2:1997: Information technology - Security techniques - Digital signature scheme giving message recovery – Part 2: Mechanisms using a hash-function

Kapitel:	I	Version:	4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite:	8	Stand:	09.07.2004	Kapitel: Literaturhinweise

- [ISO 10116] ISO 10116:1997 Information technology Security techniques - Modes of operation for an n-bit block cipher algorithm
- [ISO 10118-2] ISO 10118-2:1994 Information technology - Security techniques - Hash functions Part 2: Hash functions using an n-bit block cipher algorithm
- [ISO 10118-3] ISO 10118-3:1998 Information technology - Security techniques - Hash functions Part 3: Dedicated hash-functions, 1998
- [ISO 10126-1] ISO 10126-1:1991: Banking - Procedures for message encipherment (wholesale) – Part 1: General principles
- [ISO 10126-2] ISO 10126-2:1991 Banking - Procedures for message encipherment (wholesale) – Part 2: DEA algorithm
- [X3.92] ANSI X3.92-1981 (R1987): Data Encryption Algorithm
- [X3.106] ANSI X3.106-1983 (R1996): Data Encryption Algorithm, Modes of operation for the
- [X9.19] ANSI X9.19-1996: Financial Institution Retail Message Authentication
- [X9.23] ANSI X9.23-1995 (R1995): Financial Institution Encryption of Wholesale Financial Messages
- [X509] RFC 3039: Internet X.509 Public Key Infrastructure Qualified Certificates Profile
- [PKCS1] PKCS #1: RSA Cryptography Standard, Version 2.0, RSA Laboratories, October 1998
(<http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/>)
- [SHA-1] FIPS 180-1, Secure Hash Standard, Federal Information Processing Standards Publication 180-1, U. S. Department of Commerce / N.I.S.T., National Technical Information Service, 1995
(<http://www.itl.nist.gov/fipspubs/fip180-1.htm>)
- [RSA] R. Rivest, A. Shamir, L. Adleman: A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, vol. 21 no. 2, 1978.
- [RIPEMD] H. Dobbertin, A. Bosselaers, B. Preneel: „RIPEMD-160, a strengthened version of RIPEMD“, Fast Software Encryption - Cambridge Workshop 1996, LNCS, Band 1039, D. Gollmann, Ed., Springer-Verlag, 1996, S. 71-82
(<http://www.esat.kuleuven.ac.be/~bosselae/ripemd160.html>)

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Literaturhinweise	Stand: 09.07.2004	Seite: 9

♦ Chipapplikationen

- [ISO PIN1] ISO 9564-1, Banking – Personal Identification Number Management and Security, Part 1: PIN protection principles and techniques, DIS 1999

- [DAT-MF] Schnittstellenspezifikation für die ec-Karte mit Chip, Dateien des MF, Version 4.2, 01.12.1999

- [LT] Schnittstellenspezifikation für die ec-Karte mit Chip, Ladeterminal, Version 3.0, 02.04.1998

- [DATKOM] Schnittstellenspezifikation für die ZKA-Chipkarte, Datenstrukturen und Kommandos, Version 4.1, 01.07.1999

- [KT-KONZEPT] Schnittstellenspezifikation für die ZKA-Chipkarte, Konzept für die Unterstützung der Signatur-Anwendung der ZKA-Chipkarte durch das Internet-Kundenterminal, Version 1.0, 15. Februar 2002

- [KT-SIG] Schnittstellenspezifikation für die ZKA-Chipkarte, Spezifikation des Internet-Kundenterminals für die Unterstützung der Signatur-Anwendung der ZKA-Chipkarte (ZKA-SIG-API), Version 1.0, 7. Oktober 2002

- [SECCOS] Schnittstellenspezifikation für die ZKA-Chipkarte, Secure Chip Card Operating System (SECCOS), Version 5.0, 5. Juni 2001 mit Errata vom 13. Juni 2001

- [ZKASIG] Schnittstellenspezifikation für die ZKA-Chipkarte, Signatur-Anwendung, Version 1.0, 14. September 2001

- [BP-SIG] Bedienerrelevante Parameter der ZKA-Signaturkarte, Version 1.0, 04.03.2002

- [DINSIG] Chipcards with digital signature application/function according to SigG and SigV, Part 4: Basic Security Services, DIN V66291-4 vom 14. September 2001

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: I
Kapitel: Einleitung	Stand: 09.07.2004	Seite: 1

I. EINLEITUNG

In diesem Dokument wird das Sicherheitsverfahren HBCI beschrieben. Dieses Verfahren beruht auf modernen kryptographischen Methoden und Algorithmen, wie z. B. der Digitalen Signatur und Chipkartentechnologie.

Dieses Sicherheitsverfahren kann in multibankfähigen Online-Banking-Verfahren der deutschen Kreditwirtschaft eingesetzt werden.

Informationen bzgl. Nachrichtenaufbau und Kommunikationsablauf sind dem Dokument [Formals] zu entnehmen.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 2	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Allgemeines

II. VERFAHRENSBESCHREIBUNG

II.1 Allgemeines

Im Rahmen von HBCI-Sicherheit werden zeitgemäße Sicherheitsmechanismen und -methoden eingesetzt, welche den Missbrauch der im Bereich des Homebankings eingesetzten Systeme verhindern.

Das folgende Kapitel ist in sechs Abschnitte gegliedert, welche sich mit den verwendeten Sicherheitsmechanismen, den Abläufen, den bankfachlichen Anforderungen sowie den Segmentformaten für Signatur, Verschlüsselung und Key-Management beschäftigen.

Grundsätzlich kommen im Rahmen von HBCI-Sicherheit zwei verschiedene Sicherheitslösungen zum Einsatz:

- eine auf dem asymmetrischen RSA-Verfahren basierende Lösung in vier Varianten (Sicherheitsprofilen)
- eine auf dem symmetrischen DES-Verfahren basierende Chipkartenlösung

Die beiden Varianten werden RDH (RSA-DES-Hybridverfahren) bzw. DDV (DES-DES-Verfahren) bezeichnet. RDH signiert mit RSA-EU und chiffriert den Einmalschlüssel (verschlüsselungsbezogener Chiffrierschlüssel) mittels RSA, während DDV den MAC als Signatur verwendet und den Einmalschlüssel mittels 2-Key-Triple-DES verschlüsselt.

Die mit FinTS Version 3.0 neu aufgenommene einheitliche Chipkartenlösung für das RDH-Verfahren ist das angestrebte Zielverfahren. Da diese Sicherheitskonzeption momentan aufgrund technischer Restriktionen noch nicht flächendeckend umzusetzen ist, kommt bis zur durchgehenden Verfügbarkeit der RSA-Chipkartenlösung zusätzlich sowohl die DDV-Lösung auf Chipkartenbasis als auch die RDH-Lösung auf reiner Softwarebasis oder auf Basis proprietärer Chipkartenlösungen zum Einsatz.

♦ RDH-Verfahren

Realisierung Kreditinstitut: verpflichtend

Realisierung Kundenprodukt: verpflichtend. Ausgenommen hiervon sind Endgeräte, die eine RSA-EU-Lösung oder RDH-Verschlüsselung noch nicht erlauben (z. B. Smartphones mit MAC-Chipkarte erlauben ggf. keine RSA-EU, PC-basierte Produkte müssen hingegen stets die RSA-EU unterstützen).

♦ DDV-Verfahren

Realisierung Kreditinstitut: optional (empfohlen)

Realisierung Kundenprodukt: optional

II.1.1 Sicherheitsprofile

Die Sicherheitsverfahren RDH und DDV können unterschiedlich parametrisiert werden, wobei Sicherheitsprofile entstehen. Um die Multibankfähigkeit zu gewährleisten, ist bei Kommunikation auf Basis von FinTS 4.0 die Unterstützung des Sicherheitsprofils RDH-3 benutzer- und kreditinstitutsseitig verpflichtend. Aus Kompatibili-

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Allgemeines	Stand: 09.07.2004	Seite: 3

tätsgründen sind die in den bisherigen FinTS-Versionen genutzten Profile RDH-1 und DDV-1 weiterhin gültig. Andere als die unten genannten Profile sind nicht zulässig.

Das Kreditinstitut teilt dem Benutzer die kreditinstitutsseitig unterstützten Profile in den Bankparameterdaten mit. Der Benutzer wählt aus diesen Verfahren das für ihn geeignete Verfahren aus und bildet auf diese Weise Signatur und Verschlüsselung. Das Kreditinstitut antwortet stets mit dem vom Benutzer gewählten Verfahren.



Bankenprofil

Um die Interoperabilität mit dem ZKA Firmenkundenstandard „Abkommen über die Datenfernübertragung zwischen Kunden und Kreditinstituten (DFÜ-Abkommen)“ zu gewährleisten wird analog dem dort beschriebenen Signaturverfahren A004 ein HBCI Bankenprofil organisatorisch festgelegt, das es ermöglicht mit einer physischen Banken-Signaturkarte beide Verfahren zu verwenden.

Basis hierfür ist das Sicherheitsprofil RDH-3 mit der Einschränkung der Schlüssellänge auf 1024 Bit bis Ende 2004.

Da die meisten Kreditinstitute die zentrale Kryptoinfrastruktur an diesem Bankenprofil ausrichten werden, gilt aus Gründen der Multibankfähigkeit die maximale Schlüssellänge von 1024 Bit auch für das Sicherheitsprofil RDH-2.

Im Zuge der Notwendigkeit größerer Schlüssellängen und der Verfügbarkeit entsprechender Banken-Signaturkarten am Markt werden zukünftig im Standard neue Bankenprofile veröffentlicht werden.

Die Anwendung des Bankenprofils hat auf das FinTS-Protokoll selbst keine Auswirkung.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 4	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Allgemeines

♦ RDH-1 (verpflichtend)

Als Sicherheitsmedien für das Kundensystem sind RSA-Software-Lösungen und RSA-Chipkarten zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	RSA, Hashing nach RIPEMD-160, Padding nach ISO-9796-1	http://www.fints.org/spec/xmlschema/4.0/final/xmldsig#rsa-ripemd160-iso9796-1
Hashalgorithmus	RIPEMD-160	http://www.w3.org/2001/04/xmlenc#ripemd160
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel	S, C
Verschlüsselungsalgorithmus Einmalschlüssel	RSA ohne Padding	http://www.w3c.org/2001/04/xmlenc#rsa
Schlüssellänge	708-768 Bit	
Zertifikatstyp	ZKA UN/EDIFACT X.509	
Zertifikatsinhalt	nicht spezifiziert	

♦ RDH-2 (optional)

Als Sicherheitsmedien für das Kundensystem sind RSA-Software-Lösungen und RSA-Chipkarten zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	RSA, Hashing nach RIPEMD-160, Padding nach ISO-9796-2	http://www.fints.org/spec/xmlschema/4.0/final/xmldsig#rsa-ripemd160-iso9796-2
Hashalgorithmus	RIPEMD-160	http://www.w3.org/2001/04/xmlenc#ripemd160
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel	S, C
Verschlüsselungsalgorithmus Einmalschlüssel	RSA ohne Padding	http://www.w3c.org/2001/04/xmlenc#rsa
Schlüssellänge	1024-2048 Bit	
Zertifikatstyp	ZKA UN/EDIFACT X.509	
Zertifikatsinhalt	nicht spezifiziert	

Financial Transaction Services (FinTS)	Version:	Kapitel:
Dokument: Security - Sicherheitsverfahren HBCI	4.0	II
Kapitel: VERFAHRENSBESCHREIBUNG	Stand:	Seite:
Abschnitt: Allgemeines	09.07.2004	5

♦ RDH-3 (optional, empfohlen)

Als Sicherheitsmedium für das Kundensystem ist nur die Bankensignaturkarte oder eine gleichwertige Signaturkarte zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	mit Signierschlüssel: RSA (inkl. PKCS#1-Padding), Hashing nach RIPEMD-160	http://www.fints.org/spec/xmlschema/4.0/final/xmlsig#rsa-ripemd160
	mit Schlüssel für Digitale Signaturen: RSA, Hashing nach RIPEMD-160, Padding nach ISO-9796-2	http://www.fints.org/spec/xmlschema/4.0/final/xmlsig#rsa-ripemd160-iso9796-2
Hashalgorithmus	RIPEMD-160	http://www.w3.org/2001/04/xmlenc#ripemd160
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel Schlüssel für Digitale Signaturen	S, C, D
Verschlüsselungsalgorithmus Einmalschlüssel	RSA (inkl. PKCS#1-Padding)	http://www.w3c.org/2001/04/xmlenc#rsa-1_5
Schlüssellänge	1024-2048 Bit	
Zertifikatstyp	X.509	
Zertifikatsinhalt	EF_X509.CH.DS abh. von Sicherheitsklasse	

♦ RDH-4 (optional, empfohlen)

Als Sicherheitsmedium für das Kundensystem ist nur die Bankensignaturkarte oder eine gleichwertige Signaturkarte zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	RSA (inkl. PKCS#1-Padding), Hashing nach SHA-1	http://www.w3c.org/2000/09/xmlsig#rsa-sha1
Hashalgorithmus	SHA-1	http://www.w3c.org/2000/09/xmlsig#sha1
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel Schlüssel für Di-	S, C, D

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 6	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Allgemeines

	gitale Signaturen	
Verschlüsselungsalgorithmus Einmalschlüssel	RSA (inkl. PKCS#1-Padding)	http://www.w3c.org/2001/04/xmlenc#rsa-1_5
Schlüssellänge	1024-2048 Bit	
Zertifikatstyp	X.509	
Zertifikatsinhalt	EF_X509.CH.DS abh. von Sicherheitsklasse	

♦ DDV-1 (optional, empfohlen)

Als Sicherheitsmedium für das Kundensystem ist nur die ec-Karte mit Chip vom Typ 0 zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	DES (Retail-MAC)	http://www.fints.org/spec/xmlschema/4.0/final/xmldsig#hmac-ripemd160-type0
Hashalgorithmus	RIPEMD-160	http://www.w3.org/2001/04/xmlenc#ripemd160
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel	S, C
Verschlüsselungsalgorithmus Einmalschlüssel		http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-ecb
Schlüssellänge	128 Bit (112 Bit)	
Zertifikatstyp	nicht zulässig	
Zertifikatsinhalt	nicht zulässig	

♦ DDV-2 (optional, empfohlen)

Als Sicherheitsmedium für das Kundensystem ist nur die ec-Karte mit Chip vom Typ 1 zugelassen.

Parameter	Bedeutung/ Anmerkung	Wert (URI)
Signaturalgorithmus	DES (Retail-MAC)	http://www.fints.org/spec/xmlschema/4.0/final/xmldsig#hmac-ripemd160-type1
Hashalgorithmus	RIPEMD-160	http://www.w3.org/2001/04/xmlenc#ripemd160
Verschlüsselungsalgorithmus	2-Key-Triple-DES (Chiper Block Chaining)	http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-cbc-iso10126
Schlüsselart	Signierschlüssel Chiffrierschlüssel	S, C
Verschlüsselungsalgorithmus Einmalschlüssel		http://www.fints.org/spec/xmlschema/4.0/final/xmlenc#two-key-tripledes-ecb
Schlüssellänge	128 Bit (112 Bit)	
Zertifikatstyp	nicht zulässig	
Zertifikatsinhalt	nicht zulässig	

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Allgemeines	Stand: 09.07.2004	Seite: 7

II.1.2 Sicherheitsklassen

Für die Sicherheitsprofile RDH-3 und RDH-4 sind darüberhinaus Sicherheitsklassen definiert. Die Sicherheitsklasse gibt für jede Signatur den erforderlichen Sicherheitsdienst an. Als Sicherheitsdienst gelten derzeit „Authentifikation“ und „Non-Repudiation“.

Der Sicherheitsdienst „Authentifikation“ erfordert die Signatur mit dem Signierschlüssel (Schlüsselart „S“; Schlüssel auf Benutzerseite: SK.CH.AUT/KE). Der Sicherheitsdienst „Non-Repudiation“ erfordert die Signatur mit dem Schlüssel für Digitale Signatur (Schlüsselart „D“; Schlüssel auf Benutzerseite: SK.CH.DS).

Derzeit sind folgende Sicherheitsklassen zulässig:

Co-de	Bedeutung
1	Authentifikation
2	Non-Repudiation mit fortgeschrittener elektronischer Signatur gemäß §2, SigG
3	Non-Repudiation mit fortgeschrittener elektronischer Signatur gemäß §2, SigG und zwingender Zertifikatsprüfung
4	Non-Repudiation mit qualifizierter elektronischer Signatur gemäß §2, SigG und zwingender Zertifikatsprüfung

Die Sicherheitsklasse gibt für jeden Geschäftsvorfall den erforderlichen Sicherheitsdienst an. Signaturen gemäß der Sicherheitsklasse 2 und höher entsprechen den Anforderungen des Signaturgesetzes und erlauben damit rechtsverbindliche Willenserklärungen unter der Voraussetzung, dass die außerhalb des FinTS-Protokolls liegenden Anforderungen (z. B. Anforderungen an die Zertifizierungsinfrastruktur und an die Endgeräte) ebenfalls erfüllt sind.

Jede Signatur, die im Rahmen von HBCI-Sicherheit generiert wird, muss mindestens der festgelegten Sicherheitsklasse entsprechen:

- Technische Signaturen (Botensignaturen) können generell mit Sicherheitsklasse 1 (Authentifikation) erfolgen.
- Für Herausgebersignaturen von Geschäftsvorfällen kann das Kreditinstitut die mindestens notwendige Sicherheitsklasse individuell festlegen (Die Sicherheitsklasse wird dem Benutzer in den Bankparameterdaten des betreffenden Geschäftsvorfalles mitgeteilt)

Kapitel:	II	Version:	4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite:	8	Stand:	09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen

II.2 Mechanismen

Dieser Abschnitt beschreibt die algorithmischen Grundlagen der Sicherheitsmechanismen. Die Einbindung der hier beschriebenen Vorgänge in das FinTS-Protokoll ist in *II.5 Formate für Signatur und Verschlüsselung* und in [Syntax] beschrieben.

II.2.1 Elektronische Signatur

Die Bildung der elektronischen Signatur erfolgt durch die Vorgänge

- Bildung des Hash-Wertes
- Ergänzen des Hash-Wertes auf eine vorgegebene Länge und
- Berechnung der elektronischen Signatur über den Hash-Wert.

Je nach Sicherheitsverfahren sind die Verarbeitungsschritte jeweils verschieden.

II.2.1.1 Hashing

Als Hash-Funktion können im Rahmen von HBCI-Sicherheit abhängig vom Sicherheitsprofil entweder RIPEMD-160 [RIPEMD] oder SHA-1 [SHA-1] eingesetzt werden.

◆ RIPEMD-160

Der Hash-Algorithmus RIPEMD-160 bildet Eingabe-Bitfolgen beliebiger Länge auf einen als Bytefolge dargestellten Hash-Wert von 20 Byte (160 Bit) Länge ab. Teil des Hash-Algorithmus ist das Padding von Eingabe-Bitfolgen auf ein Vielfaches von 64 Byte. Das Padding erfolgt auch dann, wenn die Eingabe-Bitfolge bereits eine Länge hat, die ein Vielfaches von 64 Byte ist. RIPEMD-160 verarbeitet die Eingabe-Bitfolgen in Blöcken von 64 Byte Länge.

Als Initialisierungsvektor dient die binäre Zeichenfolge X'01 23 45 67 89 AB CD EF FE DC BA 98 76 54 32 10 F0 E1 D2 C3'¹.

◆ SHA-1

Der Hash-Algorithmus SHA-1 bildet Eingabe-Bitfolgen beliebiger Länge auf Bytefolgen von 20 Byte Länge ab. Teil des Hash-Algorithmus ist das Padding von Eingabe-Bitfolgen auf ein Vielfaches von 64 Byte. Das Padding erfolgt auch dann, wenn die Eingabe-Bitfolge bereits eine Länge hat, die ein Vielfaches von 64 Byte ist. SHA-1 verarbeitet die Eingabe-Bitfolgen in Blöcken von 64 Byte Länge.

II.2.1.2 Elektronische Signatur bei DDV (DES-basierend)

1. Hashing der Nachricht

Als Hash-Funktion darf nur RIPEMD-160 eingesetzt werden.

2. Formatierung des Hash-Wertes

Das Padding ist je nach Typ der eingesetzten Chipkarte (Typ 0 oder Typ 1) unterschiedlich:

¹ Little-Endian-Notation

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen	Stand: 09.07.2004	Seite: 9

Bei Typ 0-Karten erfolgt das Padding entsprechend der folgenden Abbildung mit X'00' auf das nächste Vielfache von 8 Byte:

Padding					
Byte-Position:	24	21	20	...	1
	00	00	00	00	H a s h w e r t

Bei Typ 1-Karten erfolgt das Padding entsprechend der folgenden Abbildung auf das nächste Vielfache von 8 Byte: Sei der Hash-Wert = Hash_L | Hash_R, wobei Hash_L die linken 8 Byte und Hash_R die rechten 12 Byte des Hash-Wertes bezeichnet.

	Padding									
Byte-Position:	24	23	22	...	11	10	9	8	...	1
	00	80	H a s h R				0C	81	H a s h L	

Ob eine Karte vom Typ 0 oder Typ 1 vorliegt, kann anhand der Länge der Kartenidentifikationsdaten (CID) ermittelt werden. Für Typ 0-Karten hat die CID eine Länge von 22 Byte, für Typ 1-Karten mindestens eine Länge von 24 Byte.

3. Berechnung der elektronischen Signatur

Als Signatur wird ein Retail CBC-MAC gemäß ANSI X9.19 gebildet. Hierzu wird der gepaddete Hash-Wert zunächst in 3 Blöcke der Länge 8 Byte aufgeteilt. Als Zwischenresultat wird ein einfacher CBC-MAC über die ersten 2 Blöcke berechnet. Als Initialisierungsvektor kommt X'00 00 00 00 00 00 00 00' zum Einsatz. Dabei verwendet man als Schlüssel die linke Hälfte des Signierschlüssels. Anschließend erfolgt eine 2-Key-Triple-DES-Verschlüsselung mit dem Signierschlüssel des Benutzers (muss beim Kreditinstitut hergeleitet werden) über die XOR-Summe des Zwischenergebnisses mit dem letzten Nachrichtenblock. Der so erhaltene 8 Byte-Ausgabeblock (=64 Bit) ist der Retail CBC-MAC.

II.2.1.3 Elektronische Signatur bei RDH (RSA-basierend)

1. Hashing der Nachricht

Als Hash-Funktion wird abhängig vom Sicherheitsprofil entweder RIPEMD-160 oder SHA-1 eingesetzt.

2. Formatierung des Hash-Wertes

Die Formatierung des Hash-Wertes erfolgt gemäß ISO 9796-2 (bei RDH-2 und -3) oder gemäß PKCS#1 (bei RDH-4). Übergangsweise ist für das Altverfahren RDH-1 auch die Formatierung gemäß ISO 9796:1991 (siehe [ISO 9796], Kap. 5.1-5.4) zulässig. Der Hash-Wert wird für die nachfolgende Signaturbildung als Langzahl² interpretiert (s. auch die Beispiele in der Anlage zu ISO 9796:1991).

² Unter Langzahl wird dabei die kanonische Darstellung einer natürlichen Zahl in einem Feld [0..n] bezeichnet, wobei die Wertigkeit der Felder von 0 bis n abnimmt.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 10	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen

3. Berechnung der elektronischen Signatur

Der Hash-Wert wird mittels RSA entweder gemäß DIN/ISO 9796-2 (bei RDH-2 und -3) oder gemäß PKCS#1 (bei RDH-4) signiert. Übergangsweise ist für das Altverfahren RDH-1 auch die Signatur gemäß ISO 9796-1 zulässig.³

³ Im Falle von ISO 9796-1 sind auch die dort in den Anhängen A.4 „Signature function“ und A.5 „Verification function“ beschriebenen Operationen durchzuführen und die Anhänge B und C zu berücksichtigen.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen	Stand: 09.07.2004	Seite: 11

II.2.2 Verschlüsselung

Bei jeder Verschlüsselung wird ein separater Einmalschlüssel verwendet. Die Verschlüsselung der FinTS-Nutzdaten erfolgt generell mittels 2-Key-Triple-DES gemäß ANSI X3.92. Der Einmalschlüssel wird entweder mittels 2-Key-Triple-DES (DDV-1..2) oder RSA (RDH-1..4) chiffriert und mit der verschlüsselten Nachricht mitgeliefert.



Der Einmalschlüssel muss für jede Verschlüsselung innerhalb einer FinTS-Kommunikation individuell verschieden sein. Dies muss gewährleistet werden, indem das sendende System den Einmalschlüssel dynamisch generiert.

Die ersten zwei Schritte sind für beide Verfahren identisch:

1. Der Sender erzeugt eine Zufallszahl als Einmalschlüssel und stellt ungerade Parität sicher. Bei der Auswahl der Zufallszahl ist darauf zu achten, dass keiner der folgenden schwachen oder halbschwachen Schlüssel⁴ gewählt wird (vgl. II.3.1.1 *Gemeinsam verwendete Verfahren zur Schlüsselverwaltung*).

Die schwachen Schlüssel des DES-Algorithmus:

```
X' 01 01 01 01 01 01 01 01'
X' FE FE FE FE FE FE FE FE'
X' 1F 1F 1F 1F 0E 0E 0E 0E'
X' E0 E0 E0 E0 F1 F1 F1 F1'
```

Die halbschwachen Schlüssel des DES-Algorithmus:

```
X' 01 FE 01 FE 01 FE 01 FE'
X' FE 01 FE 01 FE 01 FE 01'
X' 1F E0 1F E0 0E F1 0E F1'
X' E0 1F E0 1F F1 0E F1 0E'
X' 01 E0 01 E0 01 F1 01 F1'
X' E0 01 E0 01 F1 01 F1 01'
X' 1F FE 1F FE 0E FE 0E FE'
X' FE 1F FE 1F FE 0E FE 0E'
X' 01 1F 01 1F 01 0E 01 0E'
X' 1F 01 1F 01 0E 01 0E 01'
X' E0 FE E0 FE F1 FE F1 FE'
X' FE E0 FE E0 FE F1 FE F1'
```

2. Dieser Einmalschlüssel wird verwendet, um die Daten mittels 2-Key-Triple-DES im CBC-Modus gemäß ISO 10116 (ANSI X3.106) zu verschlüsseln (vgl. Abbildung 1). Das Padding der Nachricht erfolgt oktettorientiert gemäß ISO 10126 (ANSI X9.23), der Initialisierungsvektor ist X'00 00 00 00 00 00 00 00' (vgl. Abbildung 2 und Abbildung 3).

⁴ Die schwachen und halbschwachen Schlüssel entsprechen denen des DFÜ-Abkommens.

Kapitel:	II	Version:	4.0	Financial Transaction Services (FinTS)
				Dokument: Security - Sicherheitsverfahren HBCI
Seite:	12	Stand:	09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG
				Abschnitt: Mechanismen

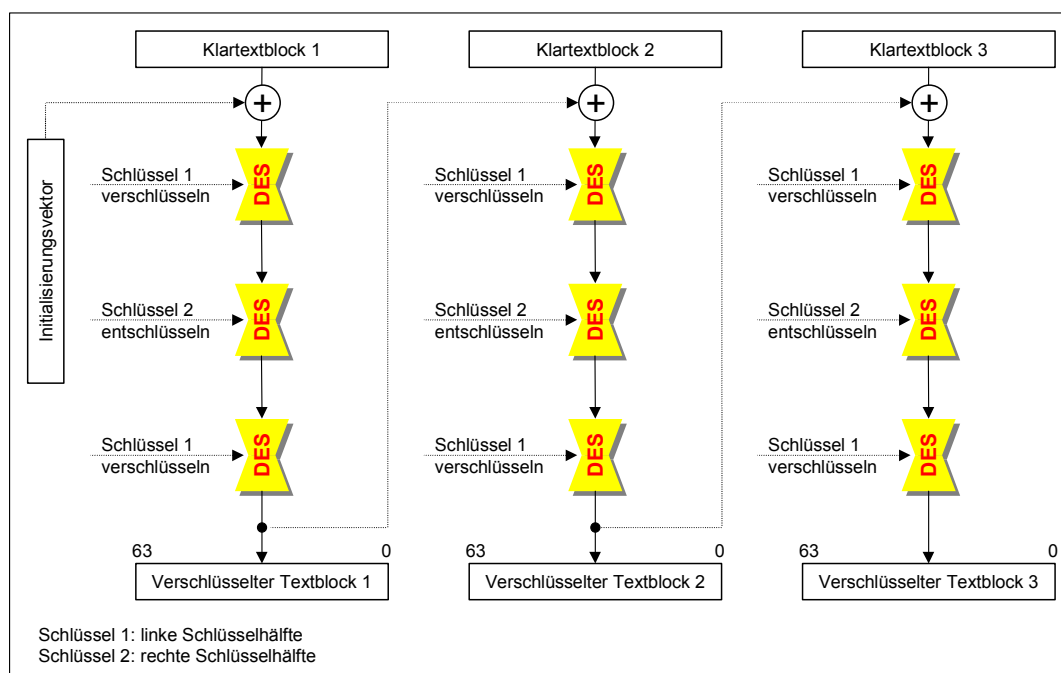


Abbildung 1: 2-Key-Triple-DES im CBC-Mode

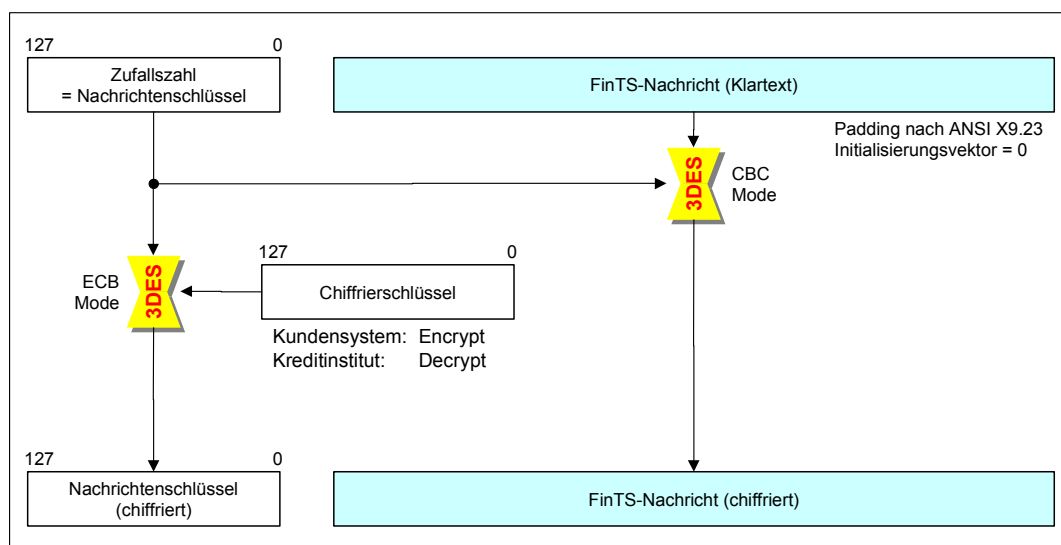


Abbildung 2: Verschlüsselung bei 2-Key-Triple-DES

Financial Transaction Services (FinTS)	Version:	Kapitel:
Dokument: Security - Sicherheitsverfahren HBCI	4.0	II
Kapitel: VERFAHRENSBESCHREIBUNG	Stand:	Seite:
Abschnitt: Mechanismen	09.07.2004	13

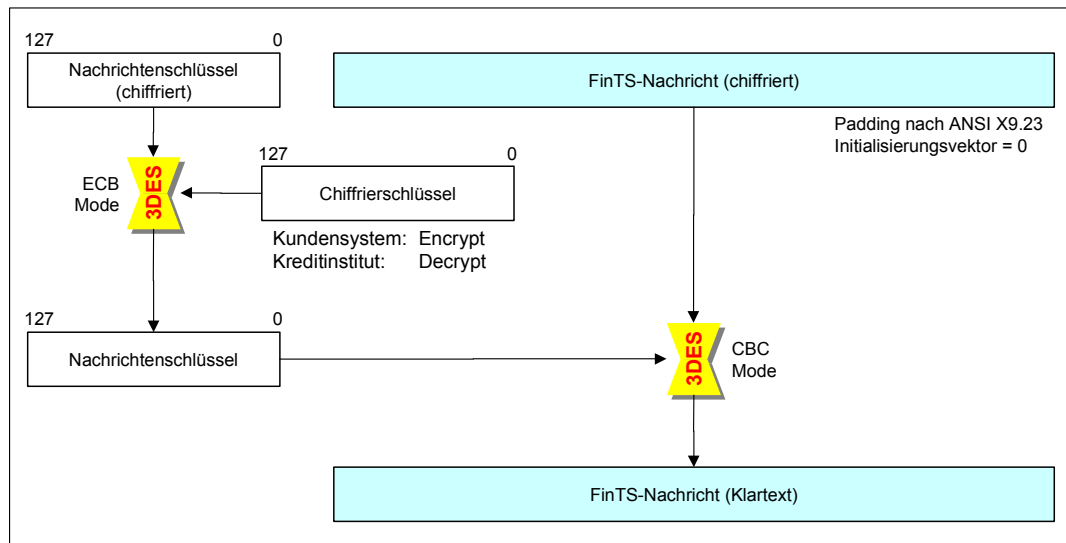


Abbildung 3: Entschlüsselung bei 2-Key-Triple-DES

Die weitere Verarbeitung ist bei DDV und RDH unterschiedlich:

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 14	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen

II.2.2.1 Verschlüsselung bei DDV (DES-basierend)

Der aktuelle Einmalschlüssel für die Chiffrierung der Daten wird vom Kundensystem mit dem kartenindividuellen Chiffrierschlüssel der Chipkarte mittels 2-Key-Triple-DES im ECB-Mode (ISO 10116) verschlüsselt (vgl. Abbildung 4, sowie Abbildung 2 und Abbildung 3).

Aufgrund vorgegebener Verfahren bei der ZKA-Chipkarte wird zum Chiffrieren und Dechiffrieren des Einmalschlüssels, unabhängig von der Übertragungsrichtung, kundensystemseitig immer die Routine „Encrypt“ benutzt, kreditinstitutsseitig immer die Routine „Decrypt“ (vgl. III.2.5.2 Entschlüsselung).

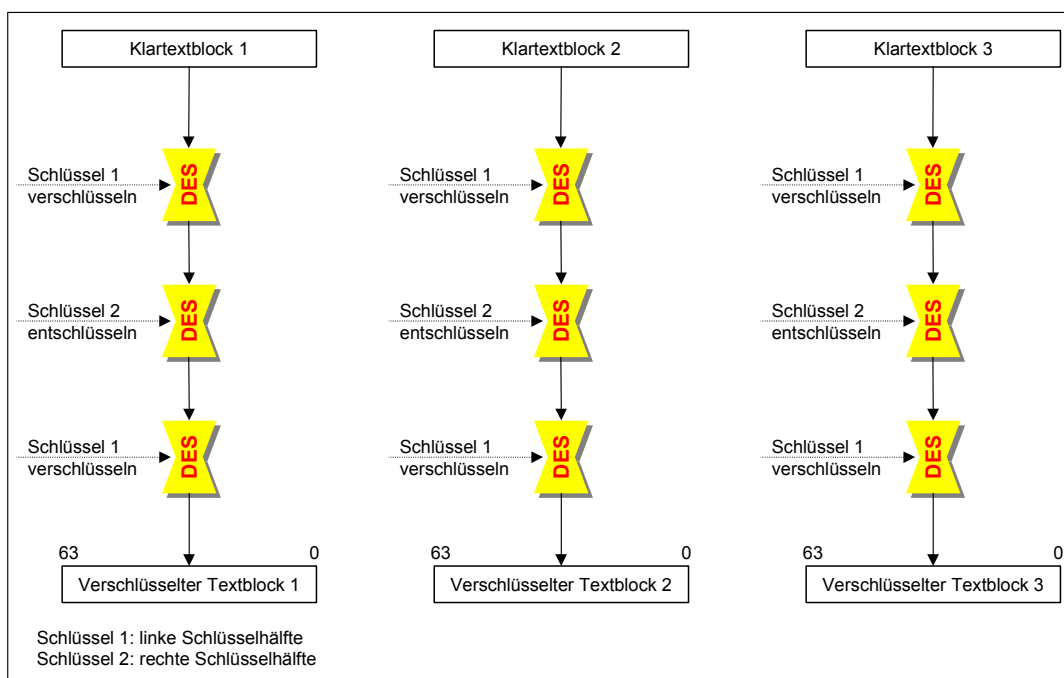


Abbildung 4: 2-Key-Triple-DES im ECB-Mode

Financial Transaction Services (FinTS)	Version:	Kapitel:
Dokument: Security - Sicherheitsverfahren HBCI	4.0	II
Kapitel: VERFAHRENSBESCHREIBUNG	Stand:	Seite:
Abschnitt: Mechanismen	09.07.2004	15

II.2.2.2 Verschlüsselung bei RDH (RSA-basierend)

Der aktuelle Einmalschlüssel wird mit dem öffentlichen Schlüssel des Empfängers chiffriert. Da die Länge des Einmalschlüssels nur 16 Byte, d. h. 128 Bit bei 2-Key-Triple-DES beträgt, muss er entsprechend auf 768 (RDH-1) bzw. 1024-2048 Bit (RDH-2, RDH-3 und RDH-4) ergänzt werden, um die vorgegebene Modulusslänge zu erreichen. Das Padding wird im Verfahren des jeweiligen Sicherheitsprofils vorgenommen, wie in den folgenden Abbildungen gezeigt.

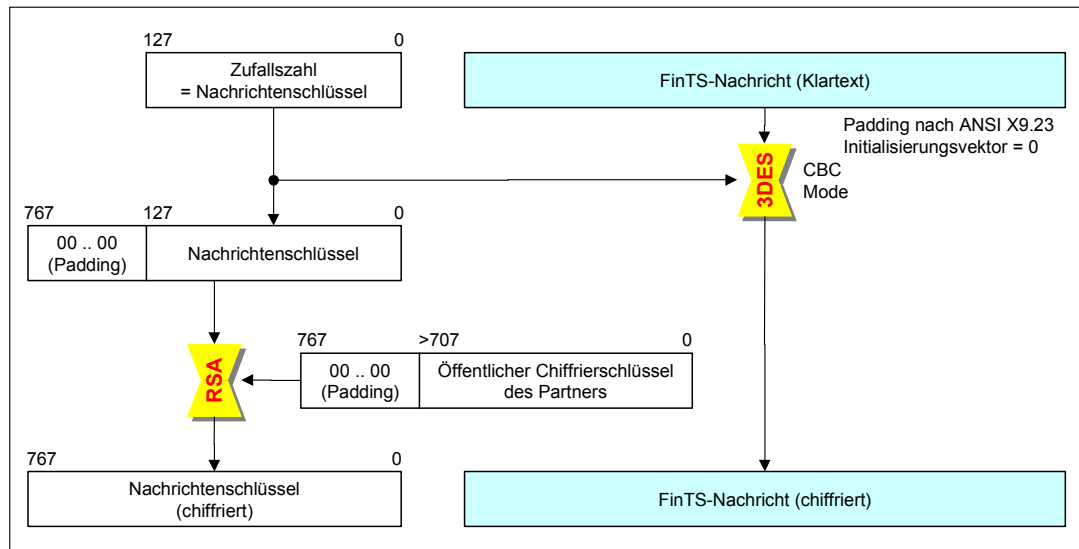


Abbildung 5: Verschlüsselung bei RDH-1

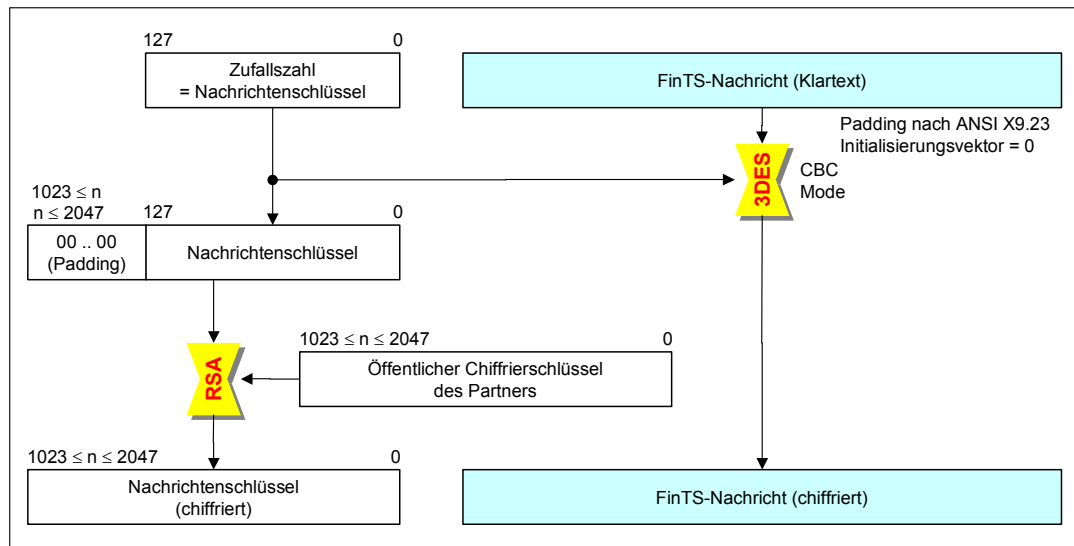


Abbildung 6: Verschlüsselung bei RDH-2

Kapitel:	II	Version:	4.0	Financial Transaction Services (FinTS)
				Dokument: Security - Sicherheitsverfahren HBCI
Seite:	16	Stand:	09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG
				Abschnitt: Mechanismen

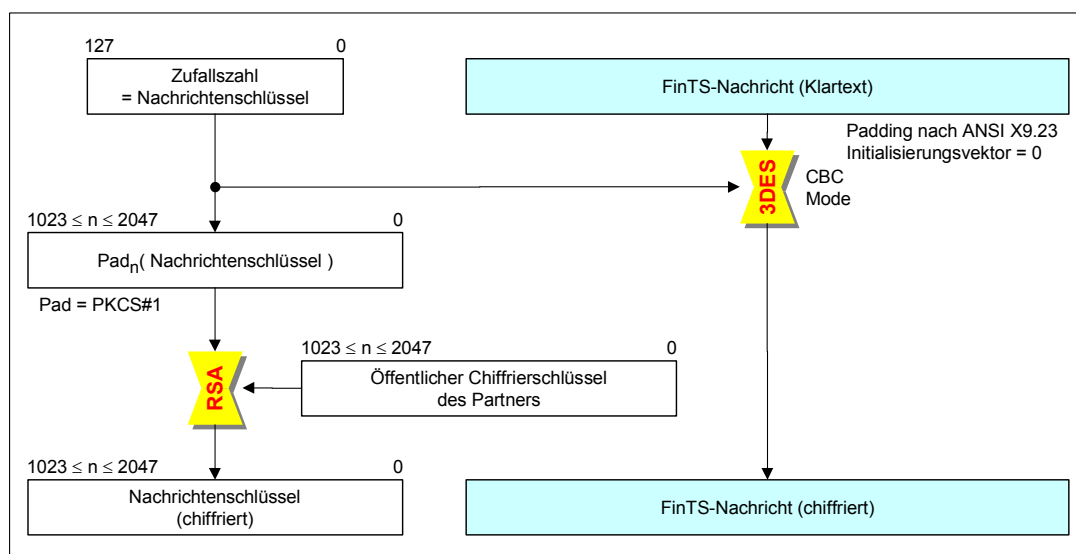


Abbildung 7: Verschlüsselung bei RDH-3 und RDH-4

II.2.3 Komprimierung

Die Komprimierung wird unabhängig vom Sicherheitsprofil nach dem deflate- oder auch GZIP-Algorithmus gemäß [RFC 1951] vorgenommen. Andere Algorithmen können zusätzlich optional angeboten werden. Zum deflate-Algorithmus gibt es eine freie, auch in kommerziellen Produkten einsetzbare Referenzimplementierung sowohl in Source-Form als auch als binäre Bibliothek für alle gängigen Plattformen (<http://www.gzip.org/zlib>).

II.2.4 Sicherheitsmedien beim Kundensystem

Bei Verwendung des symmetrischen Verfahrens (DDV) muss eine vom Kreditinstitut ausgegebene ZKA-Chipkarte eingesetzt werden, welche die Berechnung der kryptographischen Funktionen so durchführt, dass die kartenindividuellen Schlüssel niemals die Chipkarte verlassen.

Werden asymmetrische Verfahren (RDH) eingesetzt, so kann als Sicherheitsmedium eine vom Kreditinstitut ausgegebene RSA-Chipkarte oder eine Datei auf Diskette bzw. Festplatte dienen⁵. Falls eine Chipkarte zum Einsatz kommen soll, wird die in III.1 Chipapplikation für RDH beschriebene Bankensignaturkarte empfohlen. Auf dem Sicherheitsmedium wird unter anderem der private Schlüssel des Benutzers gespeichert. Es ist aber auch möglich, öffentliche Schlüssel des Kreditinstitutes darauf abzulegen oder aber im Falle einer Chipkarte die kryptographischen Operationen damit durchzuführen. Bei Einsatz einer RSA-Chipkarte müssen die geheimen Daten (z. B. private Schlüssel, Passworte) gegen unberechtigtes Auslesen geschützt sein.

⁵ Der Aufbau des Dateiformats auf der Diskette ist bei Bedarf bei der FinTS-Leitstelle erhältlich.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Mechanismen	Stand: 09.07.2004	Seite: 17



Es ist zwingend erforderlich, die Daten auf dem Sicherheitsmedium (kryptographisch) zu schützen. Speziell ist im Rahmen der Speicherung der Schlüsselpaare auf Diskette bzw. Festplatte sicherzustellen, dass die Daten unter Einbeziehung eines Passwortes (Banking-PIN o. ä.) verschlüsselt werden und der Zugriff auf die verschlüsselten Daten nur über die manuelle Eingabe des entsprechenden Passwortes möglich ist.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 18	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

II.3 Abläufe

II.3.1 Schlüsselverwaltung

Bei der Schlüsselverwaltung muss zwischen der Verwendung von symmetrischen Schlüsseln für DDV und asymmetrischen Schlüsseln für RDH unterschieden werden. Auch gibt es die Schlüsselart „D“ (Digitale Signatur) nur für RDH-3 und RDH-4.

Gemeinsam gültig sind hingegen für beide Verfahren die verwendeten Schlüsselarten „S“ und „C“, Schlüsselnamen und die Generierung von Einmalschlüsseln.

II.3.1.1 Gemeinsam verwendete Verfahren zur Schlüsselverwaltung

♦ Schlüsselarten

Bei den Sicherheitsverfahren DDV-1, DDV-2, RDH-1 und RDH-2 können Benutzer und Kreditinstitut über zwei Schlüssel bzw. Schlüsselpaare verfügen:

- einen Signierschlüssel bzw. -schlüsselpaar
- einen Chiffrierschlüssel bzw. -schlüsselpaar

Der Signierschlüssel wird zum Unterzeichnen von Transaktionen verwendet, während der Chiffrierschlüssel zum Verschlüsseln von Nachrichten dient.

Bei den Verfahren RDH-3 und RDH-4 verfügt das Kreditinstitut ebenfalls nur über die obigen beiden Schlüssel bzw. Schlüsselpaare. Benutzer können jedoch über bis zu drei Schlüssel bzw. Schlüsselpaare verfügen:

- einen Schlüssel für digitale Signaturen (DS-Schlüssel)
- einen Signierschlüssel (Authentifikation)
- einen Chiffrierschlüssel

Abhängig von der Personalisierung der Chipkarte können Signier- und Chiffrierschlüssel identisch sein.

Der Signierschlüssel und der DS-Schlüssel werden zum Unterzeichnen von Transaktionen verwendet, während der Chiffrierschlüssel zum Verschlüsseln von Nachrichten dient. Falls kreditinstitutsseitig nur Geschäftsvorfälle angeboten werden, für die gemäß Bankparameterdaten die Unterzeichnung mit dem Signierschlüssel ausreichend ist, ist der DS-Schlüssel nicht erforderlich.



Bei Verwendung von Disketten (Sicherheitsprofil RDH-2) wird dringend empfohlen, dass getrennte Signier- und Chiffrierschlüssel zum Einsatz kommen.

♦ Schlüsselnamen

Der Schlüsselname bei den 2-Key-Triple-DES- und RSA-Schlüsseln setzt sich aus den folgenden alphanumerischen Komponenten zusammen:

- Ländercode
(max. 3 Byte, es wird gemäß ISO 3166 der numerische Ländercode verwendet)
- Kreditinstitut
(max. 30 Byte, normalerweise Bankleitzahl)

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe		Stand: 09.07.2004	Seite: 19

- Benutzerkennung (bei RDH nur für Benutzerschlüssel)
(max. 30 Byte, kann vom Kreditinstitut festgelegt werden, vgl. [Formals], Abschnitt *II.1.1 Nachrichtenelemente*)
- Schlüsselart
(1 Byte, D: DS-Schlüssel (nur Benutzerschlüssel bei RDH-3 und RDH-4); S: Signierschlüssel; C: Chiffrierschlüssel)
- Schlüsselnummer
(max. 3 Byte, numerisch)
- Versionsnummer
(max. 3 Byte, numerisch)

Falls kein öffentlicher Schlüssel des Kreditinstituts vorliegt, so sind Schlüsselnummer und -version wegzulassen. Damit wird kreditinstitutsseitig auf den aktuell gültigen Schlüssel referenziert (Ein Kreditinstitut kann während einer Übergangszeit evtl. mehrere Schlüssel bis zu einem Verfallsdatum vorhalten. Aktuell gültig ist jeweils der neueste Schlüssel).

♦ Generierung von Einmalschlüsseln

Zur Chiffrierung von Nachrichten wird ein dynamisch erzeugter Einmalschlüssel verwendet, der folgendermaßen gebildet wird:

1. Generieren einer 16 Byte langen Zufallszahl
2. Erzeugung von ungerader Parität (optional)
3. Testen, ob erste und zweite Schlüsselhälfte unterschiedlich (optional)
4. Testen nach schwachen und semi-schwachen Schlüsseln (optional) (s. *II.2.2 Verschlüsselung*)

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 20	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

II.3.1.2 Symmetrische Schlüssel für DDV

Für Verschlüsselung und MAC-Berechnung werden, wie unter *II.3.1.1 Gemeinsam verwendete Verfahren zur Schlüsselverwaltung* beschrieben, unterschiedliche Schlüssel für Signatur und Chiffrierung verwendet.

II.3.1.2.1 Schlüsselgenerierung

Beim symmetrischen Verfahren (DDV) sind zur Bildung eines kartenindividuellen Schlüssels beim Kreditinstitut zwei Voraussetzungen zu erfüllen:

- Generierung eines ZKA-weit eindeutigen 2-Key-Triple-DES-Masterkey pro Schlüsselart und Ablegen in einer sicheren Umgebung (Hardwareeinrichtung) als Key Generating Key (KGK).
- Herleiten des jeweiligen kartenindividuellen Schlüssels mittels CID-Feld (Cardholders Information Data = Feld „EF_ID“) auf der ZKA-Chipkarte und entsprechendem 2-Key-Triple-DES-Masterkey.

♦ Generierung eines 2-Key-Triple-DES-Masterkey:

Für die Generierung von ZKA-weit einheitlichen 2-Key-Triple-DES-Masterkeys (KGK = Key Generating Key), die als Basis für die Herleitung der kartenindividuellen Signier- und Chiffrierschlüsseln dienen, ist folgendes Verfahren, analog der ZKA-Chipkarte, zu verwenden:

1. Generieren einer 16 Byte langen Zufallszahl
2. Erzeugung von ungerader Parität
3. Testen, ob erste und zweite Schlüsselhälfte unterschiedlich
4. Testen nach schwachen und semi-schwachen Schlüsseln (s. *II.2.2 Verschlüsselung*)

♦ Herleitung von Kartenschlüsseln:

Zur eindeutigen Herleitung der symmetrischen Signier- und Chiffrierschlüssel wird das Feld „EF_ID“ im Master File (MF) der ZKA-Chipkarte (Cardholders Information Data (CID) ohne Padding) zusätzlich übertragen.

Ein kartenindividueller Schlüssel KK von 16 Byte Länge wird aus

- KGK (Key Generating Key, 16 Byte)
- CID (vollständiger Inhalt von EF_ID, mit X'00' auf das nächste Vielfache von 8 Byte Länge aufgefüllt) und
- dem öffentlich bekannten Initialwert I = X'52 52 52 52 52 52 52 52 25 25 25 25 25 25 25' (16 Byte)

zu

$$KK = P(d * KGK(H(I, CID)))$$

berechnet.

Hierbei bezeichnen

- 'P' die Funktion "Parity Adjustment" auf ungerade Parität, die wie folgt definiert ist:
Sei b_1, \dots, b_8 die Darstellung eines Byte als Folge von 8 Bit. Dann setzt P das

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe		Stand: 09.07.2004	Seite: 21

niedrigstwertige Bit b_8 jedes Byte auf ungerade Parität, d. h. b_8 wird in jedem Byte so gesetzt, dass es eine ungerade Anzahl von 1-Bits enthält.

- 'd * KGK' die 2-Key-Triple-DES-Entschlüsselung im ECB-Mode (ISO 10116) mit dem Schlüssel KGK.
- 'H' die in ISO 10118-2 definierte Hash-Funktion.

II.3.1.2.2 Initiale Schlüsselverteilung

Die initiale Schlüsselverteilung erfolgt implizit mit der Verteilung der Chipkarte.

II.3.1.2.3 Schlüsseländerungen

Beim symmetrischen Verfahren (DDV) ist wegen der Verknüpfung mit der Chipkarte auf elektronische Weise keine Änderung einzelner kartenindividueller Schlüssel möglich. Im Falle einer vermuteten Kompromittierung muss daher ein Kartenaustausch oder ein Ersatz aller Schlüssel und des Feldes „EF_ID“ erfolgen.

Bei einer Schlüsseländerung wird die Signatur-ID (Sequenzzähler der Chipkarte) auf 1 zurückgesetzt. Die im Kreditinstitut geführte Liste der eingereichten bzw. noch nicht eingereichten Signatur-IDs (s. *II.4 Bankfachliche Anforderungen*) wird gelöscht.

II.3.1.2.4 Schlüsselverteilung nach Kompromittierung

Die Schlüsselverteilung nach einer Kompromittierung erfolgt ebenfalls mittels Vergabe einer neuen Chipkarte bzw. Ersatz aller Schlüssel und des EF-ID-Feldes. Die alte Chipkarte bzw. deren Schlüssel werden gesperrt.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 22	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

II.3.1.3 Asymmetrische Schlüssel für RDH

Grundsätzlich können Benutzer und Kreditinstitut beim asymmetrischen Verfahren (RDH) über maximal drei bzw. zwei Schlüsselpaare verfügen:

- ein Signierschlüsselpaar
- ein Chiffrierschlüsselpaar
- ein Schlüsselpaar für die Erzeugung Digitaler Signaturen (DS) (nur für Benutzer bei RDH-3 und RDH-4)

Der Signierschlüssel sowie der DS-Schlüssel werden zum Unterzeichnen von Nachrichten und Geschäftsvorfällen verwendet, während der Chiffrierschlüssel zum Verschlüsseln von Nachrichten und Geschäftsvorfällen dient (vgl. *II.2 Mechanismen*).

Falls ein Kreditinstitut seine Nachrichten nicht signiert, kann es auf das Signierschlüsselpaar verzichten.

II.3.1.3.1 Schlüsselgenerierung

Die Schlüsselpaare des Benutzers sind vom Kundensystem bzw. von der Chipkarte zu erzeugen. Die Schlüsselpaare des Kreditinstituts sind vom Kreditinstitut zu erzeugen. Die privaten Schlüssel sind jeweils geheim zu halten.

Die Schlüsselgenerierung hat gemäß dem folgenden Ablauf stattzufinden:⁶

1. Es wird ein konstanter öffentlicher Exponent e und ein für jeden Benutzer individueller Modulus n für jedes eingesetzte RSA-Schlüsselsystem verwendet.
2. Der konstante öffentliche Exponent e wird auf die 4. Fermat'sche Primzahl festgelegt: $e = 2^{16} + 1$
3. 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: $2^{N-1} \leq n < 2^N$
4. Der Zielwert für N ist bei RDH-1 768, wobei eine aus der Suche nach starken Primzahlen resultierende Unterschreitung dieses Wertes um maximal 60 Bit zulässig ist. Bei RDH-2, RDH-3 und RDH-4 liegt der Zielwert für N zwischen 1024 und 2048.
5. 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 Primteiler⁷ r
 - $p + 1$ hat einen großen Primteiler s
 - $r - 1$ hat einen großen Primteiler

Die entsprechenden Forderungen werden an q gestellt.

Die Längen von p und q sollen sich um höchstens 12 Bits unterscheiden.

Bei der Wahl von p und q ist sicherzustellen, dass e kein Primfaktor von $p - 1$ oder $q - 1$ ist.

⁶ Das Verfahren entspricht dem des DFÜ-Abkommens.

⁷ Der Primteiler sollte dabei ungefähr der Länge des Schlüssels entsprechen.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe	Stand: 09.07.2004	Seite: 23

II.3.1.3.2 Initiale Schlüsselverteilung

Der Benutzer benötigt für das Einrichten eines neuen Zugangs folgende Initialinformationen:

- seine Benutzerkennung
- Informationen zum Kommunikationszugang

Die Übermittlung dieser Informationen ist auf folgenden Wegen denkbar:

- Schriftstück des Kreditinstitutes (Benutzerkennung und Zugangsdaten müssen manuell vom Benutzer eingegeben werden)
- Diskette des Kreditinstitutes mit folgendem Inhalt:
 - allgemeiner Teil der UPD inkl. Benutzerkennung
 - BPD oder BPD-Ausschnitt (Zugangsdaten-Segment) mit den Kommunikationszugangsdaten des jeweiligen Instituts
- Chipkarte des Kreditinstitutes, die die Kommunikationszugangsdaten in der Applikation EF_NOTEPAD enthält.

Zu Beginn muss ein gegenseitiger Austausch der öffentlichen Schlüssel von Benutzer und Kreditinstitut erfolgen. Zukünftig soll dieser Austausch durch eine Anforderung der Zertifikate bei den jeweiligen Zertifizierungsinstanzen erfolgen. Dieser Prozess findet außerhalb des FinTS-Protokolls statt und wird daher hier nicht näher beschrieben. Übergangsweise kann der Schlüsselaustausch auch im Rahmen einer FinTS-Kommunikation erfolgen.

Hierzu ist folgender Ablauf vorgesehen:

1. Das Kreditinstitut übermittelt seinen öffentlichen Chiffrierschlüssel an den Benutzer. Falls es Nachrichten signiert, übermittelt es ebenfalls seinen öffentlichen Signierschlüssel. Hierzu gibt es zwei Möglichkeiten:
 - Zusenden bzw. Aushändigung der Schlüssel und anderer relevanter Daten auf einem Medium (z. B. Diskette⁸, Chipkarte) bei Vertragseröffnung.
Falls dem Benutzer eine Diskette zugesendet wird, hat diese folgende Daten zu enthalten:
 - Datei mit bis zu zwei FinTS-Segmenten, die jeweils einen öffentlichen Schlüssel des Kreditinstitutes enthalten
 - BPD des Kreditinstitutes
 - Übertragung der Schlüssel beim Erstzugang
 - (1) Der Benutzer fordert die öffentlichen Schlüssel und die BPD mit Hilfe der Key-Management-Nachricht „Erstmalige Anforderung der Schlüssel des Kreditinstituts“ (s. II.6.1.2 *Erstmalige Anforderung der Schlüssel des Kreditinstituts*) an. Diese Nachricht ist weder signiert noch chiffriert.

⁸ Es kann sich hierbei um dieselbe Diskette handeln, mit der dem Benutzer seine Benutzerkennung mitgeteilt wird (s.o.).

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 24	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

- (2) Der weitere Ablauf ist abhängig davon, ob das Kreditinstitut seine Antwortnachrichten signiert.

Fall A: Das Kreditinstitut signiert

Der Benutzer erhält die öffentlichen Schlüssel des Kreditinstituts zurückgemeldet. Während die Authentizität des Chiffrierschlüssels dabei durch die Signatur gesichert ist, ist die Authentizität des Signierschlüssels nicht gesichert, da das Kundensystem die Echtheit der Signatur noch nicht prüfen kann.

Fall B: Das Kreditinstitut signiert nicht

Der Benutzer erhält nur den öffentlichen Chiffrierschlüssel zurückgemeldet. Dessen Authentizität ist dabei nicht gesichert.

- (3) Die Sicherung der Authentizität dieser Schlüssel kann über folgende Mechanismen erfolgen:

Fall A: Ini-Brief

Diese Nachricht wird von einem Ini-Brief an den Benutzer begleitet. Die Gestaltung ist dem Kreditinstitut freigestellt, sollte sich aber am Muster in Abbildung 9 orientieren. Der Ini-Brief enthält für den Fall A Exponent und Modulus des Signierschlüssels sowie dessen Hash-Wert und für den Fall B Exponent und Modulus des Chiffrierschlüssels sowie dessen Hash-Wert. Exponent und Modulus sind dabei mit führenden Nullen (X'00') auf den Zielwert für die Schlüssellänge (1024 Bit bei RDH-1, 2048 Bit bei RDH-2, -3 und -4) zu ergänzen. Ferner enthält der Ini-Brief den jeweiligen Schlüsselnamen.

Bei der Hash-Wertbildung ist wie folgt vorzugehen:

- Padding der höchstwertigen Bits von Exponent und Modulus des Schlüssels mit Nullen (X'00') auf den Zielwert der Schlüssellänge
- Konkatenierung von Exponent und Modulus (Exponent || Modulus)
- Bildung des Hash-Wertes mittels RIPEMD-160 entsprechend Bankenprofil (siehe *II.1.1 Sicherheitsprofile*) gemäß *II.2.1.1 Hashing* über diesen Ausdruck

Nach Erhalt des Ini-Briefs führt der Benutzer einen Vergleich des im Ini-Brief aufgeführten Hash-Wertes mit dem Hash-Wert des vom Kreditinstitut übermittelten Schlüssels durch.

Bei Übereinstimmung der Hash-Werte gelten der bzw. die öffentlichen Schlüssel des Kreditinstituts als authentisiert.



Das Kundensystem sollte den Hash-Wert-Vergleich für den Benutzer in geeigneter Weise unterstützen.

Fall B: Übermittlung des Hash-Wertes auf der Chipkarte

Auf der Karte befindet sich in der Applikation EF_NOTEPAD (s. *III.1.1 Applikation Notepad*) für Fall A der Hash-Wert des öffentli-

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe	Stand: 09.07.2004	Seite: 25

chen Signierschlüssels des Kreditinstituts und für Fall B der Hash-Wert des öffentlichen Chiffrierschlüssels des Kreditinstituts. Die Hash-Wert-Bildung erfolgt wie in Fall A.

Dieser Hash-Wert wird vom Kundensystem mit dem Hash-Wert des in der Nachricht übermittelten Schlüssels verglichen.



Das Kundensystem sollte den Benutzer über das Ergebnis des Hash-Wertvergleichs informieren.

Bei Übereinstimmung der Hash-Werte gelten der bzw. die öffentlichen Schlüssel des Kreditinstituts als authentisiert.

Fall C: Prüfung des übermittelten Zertifikates

Falls das Kreditinstitut über zertifikatsbasierte Schlüssel verfügt, übermittelt es das jeweilige Zertifikat in der Nachricht zusammen mit dem öffentlichen Schlüssel. Somit ist der Benutzer in der Lage, das Zertifikat bei der jeweiligen Zertifizierungsinstanz zu verifizieren. Diese Verifikation findet außerhalb des FinTS-Protokolls statt und wird daher hier nicht näher beschrieben.

Ein Hash-Wert-Vergleich wie in den beiden anderen Fällen ist nicht erforderlich.



Das Kundensystem sollte den Benutzer über das Ergebnis der Zertifikatsprüfung informieren.

- Der Benutzer übermittelt alle seine öffentlichen Schlüssel, die mit dem privaten Signierschlüssel unterzeichnet wurden, im Rahmen der Key-Management-Nachricht „Erstmalige Übermittlung der Schlüssel des Benutzers“ an das Kreditinstitut (vgl. *II.6.1.3 Erstmalige Übermittlung der Schlüssel des Benutzers*). Diese Nachricht muss sowohl signiert als auch chiffriert sein.
- Um die Authentizität der Schlüssel zu gewährleisten, sind folgende Mechanismen möglich:

Fall A: Ini-Brief

Der Benutzer erfährt anhand des Rückmeldungscode 3310 („Ini-Brief erforderlich“) in der Kreditinstitutsnachricht, dass diese Nachricht durch einen Ini-Brief gemäß dem in Abbildung 9 aufgeführten Muster begleitet werden muss. Im Ini-Brief bestätigt der Benutzer ausschließlich den öffentlichen Signierschlüssel mit handschriftlicher Unterschrift. Eine Bestätigung des öffentlichen Chiffrierschlüssels ist nicht erforderlich, da dieser mit dem Signierschlüssel signiert wird und damit authentifiziert ist. Neben dem Schlüssel und dem Schlüsselnamen wird im Ini-Brief der Hash-Wert des Schlüssels aufgeführt. Dieser wird ebenso gebildet wie der Hash-Wert im Ini-Brief des Kreditinstituts (s. o.).

Im Kreditinstitut findet ein Vergleich zwischen dem im Ini-Brief aufgeführten Hash-Wert und dem Hash-Wert des vom Benutzer übermittelten öffentlichen Signierschlüssels statt.

Falls dieser Vergleich positiv verläuft, werden die öffentlichen Schlüssel des Benutzers freigeschaltet.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 26	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

Fall B: Prüfung des übermittelten Zertifikates

Der Benutzer erfährt anhand des Rückmeldungscode 3320 („Ini-Brief nicht erforderlich“) in der Kreditinstitutsnachricht, dass das Kreditinstitut die Prüfung der Authentizität der Schlüssel auf Basis eines Zertifikates vornehmen kann.

Falls der Benutzer über zertifikatsbasierte Schlüssel verfügt, übermittelt er daher das jeweilige Zertifikat in der Nachricht zusammen mit dem öffentlichen Schlüssel. Somit ist das Kreditinstitut in der Lage, das Zertifikat bei der jeweiligen Zertifizierungsinstanz zu verifizieren. Diese Verifikation findet außerhalb des FinTS-Protokolls statt und wird daher hier nicht näher beschrieben.

Ein Hash-Wert-Vergleich wie in Fall A ist nicht erforderlich.

Bei erfolgreicher Zertifikatsprüfung werden die öffentlichen Schlüssel des Benutzers freigeschaltet.

- Bei softwarebasiertem RDH hat eine Synchronisierung der Kundensystemkennung zu erfolgen (s. [Formals], Abschnitt *III.3 Synchronisierung*).
- Nachdem die Erstinitialisierung abgeschlossen ist, kann der Benutzer Auftragsnachrichten senden.

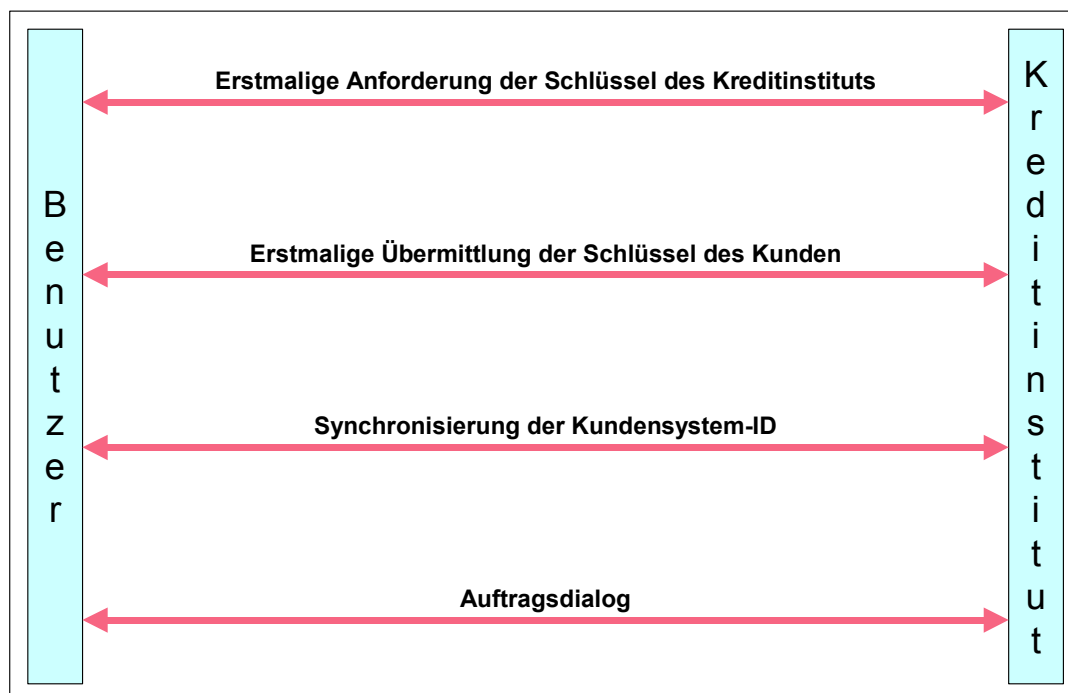


Abbildung 8: Ablauf der Erstinitialisierung bei RDH

Um die Multibankfähigkeit verschiedener Kundensysteme zu sichern, gelten für die Ini-Diskette folgende Namenskonventionen:

- UPD allgemeiner Teil: <Benutzerkennung>.UPA
- Datei mit den öffentlichen Schlüsseln: <Benutzerkennung>.PKD
- BPD: <Bankleitzahl>.BPD
- Segment mit Kommunikationszugang: <Bankleitzahl>.KOM

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 28	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

Falls die Karte die Generierung neuer Schlüssel zulässt oder im Falle anderer Speichermedien (Diskette) ändert der Benutzer seine Schlüsselpaare unabhängig voneinander.

Der Benutzer sendet je Kreditinstitut im Rahmen einer FinTS- Kommunikation eine Nachricht, in welcher dieses über einen neuen öffentlichen Schlüssel informiert wird (vgl. *II.6.1.1 Änderung eines öffentlichen Schlüssels des Benutzers*). Die Nachricht ist mit dem alten (bei Wechsel des Signierschlüssels), respektive dem aktuellen (bei Wechsel des Chiffrierschlüssels) privaten Signierschlüssel des Benutzers zu signieren und mit dem aktuellen Chiffrierschlüssel des Kreditinstituts zu chiffrieren. Das Kreditinstitut speichert diesen neuen öffentlichen Schlüssel des Benutzers und verwendet ihn, mit Ausnahme bei einem Profilwechsel (s. Kap. Fehler! Verweisquelle konnte nicht gefunden werden.), ab sofort (d. h. bereits in der Antwortnachricht) für alle Verschlüsselungen bzw. Verifikationen von Signaturen. Gleichzeitig wird der alte Schlüssel gesperrt.

Falls die Übermittlung der neuen Schlüssel aus irgendeinem Grunde fehlschlägt, kann der Benutzer den Vorgang beliebig wiederholen.

Bei einer Schlüsseländerung wird die Signatur-ID auf 1 zurückgesetzt. Die Liste der eingereichten bzw. noch nicht eingereichten Signatur-IDs (s. *II.4 Bankfachliche Anforderungen*) wird gelöscht.

♦ Routinemäßige Schlüsseländerung des Kreditinstituts

Ein Kreditinstitut generiert bei Bedarf ein neues Schlüsselpaar.

Falls das Kreditinstitut über aktuellere öffentliche Schlüssel verfügt als der Bote der Initialisierungsnachricht, werden diese in der Kreditinstitutsantwort auf die Initialisierungsnachricht mit übertragen (vgl. [Formals], Abschnitt *II.15 Initialisierung*). Die neuen Schlüssel gelten ab sofort, d. h. bereits für die erste Folgenachricht. Da das Kreditinstitut i. d. R. aber auch noch die alten Schlüssel aktiv hält, werden für einen begrenzten Zeitraum auch noch Nachrichten akzeptiert, die mit den alten Kreditinstitutsschlüsseln chiffriert wurden.

Zur Verifikation des kreditinstitutsseitigen öffentlichen Schlüssels auf dem Kundensystem kann das entsprechende Kreditinstitut die Kreditinstitutsnachricht mit dem alten Signierschlüssel signieren (wenn eine kreditinstitutsseitige Signatur vorgesehen ist) oder den Hash-Wert des öffentlichen Schlüssels analog der initialen Schlüsselverteilung an den Benutzer übermitteln. Die Verifikation ist grundsätzlich optional.

Für den Fall, dass der alte Kreditinstitutsschlüssel nicht mehr zur Verfügung steht oder gesperrt werden musste, wird dem Benutzer – falls er den alten Kreditinstitutsschlüssel zur Chiffrierung der Initialisierung verwendet – der Rückmeldungscode "9030" mit dem Hinweis "Fehler beim Entschlüsseln" gesendet. Ggf. kann die Initialisierung vom Kreditinstitutssystem auch gar nicht verarbeitet werden, so dass keine Antwort gesendet wird. Daraufhin sollte das Kundensystem über eine anonymen Kommunikation mit Hilfe der Nachricht „Erstmalige Anforderung der Schlüssel des Kreditinstituts“ (s. *II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts*) die neuen Kreditinstitutsschlüssel anfordern. Zur Verifikation der neuen Schlüssel muss dem Benutzer in diesem Fall zusätzlich ein Ini-Brief mit dem Hash-Wert des neuen Kreditinstitutsschlüssels zugeschickt werden.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe	Stand: 09.07.2004	Seite: 29

II.3.1.3.4 Schlüsselverteilung nach Kompromittierung

Die Verteilung der Schlüssel nach einer Kompromittierung erfolgt analog der Schlüsselverteilung bei der Initialisierung. Es findet immer ein Austausch aller Schlüssel statt, auch dann, wenn nur einer der Schlüssel kompromittiert wurde.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 30	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe

II.3.2 Schlüsselsperrung

Bei der Schlüssel- bzw. Benutzersperrung muss zwischen folgenden Fällen unterschieden werden:

- Kompromittierung des eigenen Schlüssels
- Verlust des eigenen Schlüssels
- Überschreiten der Anzahl der Falschsignaturen

Zusätzlich müssen bei der Sperrung noch folgende Punkte berücksichtigt werden:

- Information des Benutzers
- Entsperrung

Die Sperrung anderer Benutzer wird als eigenständiger Auftrag behandelt und zu einem späteren Zeitpunkt realisiert.

◆ Kompromittierung des eigenen Schlüssels

Bei Verdacht auf Kompromittierung des eigenen Schlüssels kann die Sperrung mittels einer speziellen Nachricht (vgl. *II.6.1.4 Schlüsselsperrung durch den Benutzer*) erfolgen, welche signiert sein muss.

◆ Verlust des eigenen Schlüssels

Bei einem Verlust (inkl. Diebstahl) des eigenen Schlüssels (respektive des Speichermediums) muss der Benutzer Schlüssel bzw. Medium sperren und beim Kreditinstitut ein anderes Medium inkl. Schlüssel beantragen.

Eine nicht-signierungspflichtige Sperrmöglichkeit ist nicht vorgesehen, da hierdurch die Gefahr des Missbrauchspotential gegeben ist (absichtliche Sperrung fremder Anschlüsse).

Eine Sperrung auf anderem Weg (z. B. telefonische Sperrung über Servicezentralen) muss immer möglich sein (z. B. Verlust der eigenen Infrastruktur).

◆ Überschreiten der Anzahl der Falschsignaturen

Wird beim Einreichen von Aufträgen durch fehlerhafte Signaturen die festgelegte Anzahl von n Falschsignaturen in Folge überschritten, werden kreditinstitutsseitig die Schlüssel gesperrt. Als Falschsignaturen werden dabei fehlgeschlagene kryptographische Operationen, jedoch z. B. keine fehlerhaften Berechtigungen verstanden.

Bei einer Sperrung aufgrund zu vieler Fehlsignaturen werden alle Benutzerschlüssel gesperrt. Sofern die Nachricht lediglich von einem einzigen Benutzer signiert wurde oder falls bei einer mehrfach signierten Nachricht der Bote von der Fehlsignatursperre betroffen ist, wird die Kommunikation beendet. Der Kommunikationsabbruch erfolgt dabei kreditinstitutsseitig im Anschluss an die Antwortnachricht, d. h. ein Austausch von Endenachrichten findet nicht statt. Die Antwort ist beim DDV-Verfahren weder signiert noch verschlüsselt. Beim RDH-Verfahren ist die Antwort signiert (sofern kreditinstitutsseitig signiert wird) aber nicht verschlüsselt. In der Antwortnachricht teilt das Kreditinstitut lediglich den Grund der Kommunikation sendes mit. Antworten auf Aufträge dürfen nicht mitgesendet werden, da diese aufgrund der Sperrung nicht abgesichert werden können.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Abläufe	Stand: 09.07.2004	Seite: 31

◆ Information des Benutzers

Im Falle einer Sperrung aufgrund von Schlüsselkompromittierung oder Schlüsselverlust erhält der Benutzer auf die Sperrnachricht eine Antwortnachricht (vgl. *II.6.1.4 Schlüsselsperrung durch den Benutzer*), welche ihm die Sperrung bestätigt. Bei einer Sperrung wegen Überschreitung des Maximalwertes möglicher Falschsignaturen erhält er lediglich einen entsprechenden Rückmeldungscode. In jedem Fall erhält er jedoch entsprechende Fehlermeldungen bei der Einreichung nachfolgender Nachrichten.

◆ Entsperrung der Benutzerkennung

Eine Entsperrung erfolgt nur gegen handschriftliche Unterschrift des Benutzers.

Ist der Schlüssel kompromittiert oder nicht mehr auffindbar, so wird für den Benutzer eine neue Chipkarte, respektive neue Schlüssel und ein neues EF_ID (DDV), oder ein neues Schlüsselpaar (RDH) erzeugt und der alte Schlüssel bleibt gesperrt. Es werden in jedem Falle alle Schlüsselpaare neu vergeben, auch wenn nur ein Schlüsselpaar kompromittiert sein sollte. Damit ein Benutzer nach einer Sperrung wieder zum Zugang zum System autorisiert werden kann, darf er in diesem Fall ausnahmsweise einer erneute Erstinitialisierung durchführen und seine Schlüssel über einen Ini-Brief freischalten lassen.

In den übrigen Fällen kann der Schlüssel einfach durch das Kreditinstitut entsperrt werden.

Kapitel:	II	Version:	4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite:	32	Stand:	09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Bankfachliche Anforderungen

II.4 Bankfachliche Anforderungen

◆ Zu signierende Nachrichten

Grundsätzlich sind alle Benutzernachrichten mindestens mit Botensignatur zu signieren, bei Sicherheitsprofil RDH-3 und RDH-4 gemäß den in den BPD vorgegebenen Sicherheitsklassen. Ausnahmen gelten für den anonymen Zugang ([Formals], Abschnitt *II.17 Anonymer Zugang*), für die Schlüsselanforderung im Rahmen der Erstinitialisierung (*II.6.1.1 Änderung eines öffentlichen Schlüssels des Benutzers*) und für die Lebendmeldung ([Formals], Abschnitt *III.6 Lebendmeldung in Dialogen*). Die Signatur von Kreditinstitutsnachrichten ist optional.

◆ Doppeleinreichungskontrolle

Die Doppeleinreichungskontrolle wird mittels eines Zählers pro Signatur realisiert (Signatur-ID), dessen Inhalt jeweils in die Signatur(en) der Nachricht einfließt. Falls als Sicherheitsmedium keine Chipkarte verwendet wird, wird zur Doppeleinreichungskontrolle zusätzlich zur Signatur-ID die Kundensystemkennung benötigt.

Bei der Doppeleinreichungskontrolle (Verhinderung von Replay-Attacken) ist zu berücksichtigen, dass die sequentiell erzeugten Referenznummern (=Signatur-IDs) beim Kreditinstitut nicht in derselben Reihenfolge eintreffen müssen, da diese benutzerseitig auch offline (d. h. zeitlich voneinander unabhängig) generiert werden können. Das Kreditinstitut muss deshalb sicherstellen, dass innerhalb eines bestimmten Zeitraums keine Sequenznummer mehrfach erscheint.

Aus diesem Grund muss beim Kreditinstitut eine Liste mit den eingereichten (Positivliste) oder noch nicht eingereichten (Negativliste) Signatur-IDs geführt werden. Nach einer festgelegten Aufbewahrungsfrist wird eine Referenznummer nicht mehr akzeptiert. (Konkret wird ein Kreditinstitut eine Nachricht abweisen, welche länger als die vereinbarte Frist nach einer Nachricht mit höherer Signatur-ID eintrifft). Diese Liste muss je Signaturschlüsselpaar geführt werden, d. h., falls der Benutzer sowohl mit dem Signierschlüssel- als auch mit dem DS-Schlüssel unterschreibt, sind zwei Listen erforderlich.

◆ Mehrfachsignaturen

Bei Mehrfachsignaturen ist die Reihenfolge der Signaturen bedeutungslos.

Sind die Berechtigungsprofile mehrerer signierender Benutzer zueinander inkonsistent, so liegt es im Ermessen des Kreditinstituts, ob es die Nachricht annimmt oder ablehnt (Beispiel: Der Herausgeber eines Auftragsteils, für deren Aufträge drei Signaturen erforderlich sind, liefert nur eine zweite Signatur eines Benutzers mit, der über das Recht verfügt, die Aufträge alleine zu signieren).

Ob es zulässig ist, dass bei Mehrfachsignaturen verschiedene Signaturverfahren eingesetzt werden, gibt das Kreditinstitut in den BPD im Segment „Sicherheitsverfahren“ (siehe [Formals], Abschnitt *IV.2.3 Sicherheitsverfahren*) an.

◆ verteilte Signaturen

Falls ein Auftrag mehrere Signaturen benötigt, diese jedoch nicht sofort erstellt und innerhalb der gleichen Nachricht übermittelt werden können, so besteht die Möglichkeit, die Signaturen auch zeitlich und räumlich getrennt zu erstellen. Die zugehörigen Abläufe sind in [Formals], Abschnitt *III.8 Verteilte Signaturen* beschrieben. Die hier vorgestellten Geschäftsvorfälle sind in den BPD als zulässig zu hinterlegen, ebenfalls die Geschäftsvorfälle, die verteilt signiert werden sollen.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung	Stand: 09.07.2004	Seite: 33

II.5 Formate für Signatur und Verschlüsselung

Für die Speicherung der Sicherheitsinformationen für die Signatur(en) werden dem XML-Signature-Standard entsprechend Signatur-Segmente in die bestehende Nachricht eingefügt. Diese Segmente enthalten jeweils XML-Referenzen auf die signierten Teile der Gesamtnachricht. Dies muss im Falle einer Botensignatur die Gesamtnachricht umfassen, oder kann sich im Falle einer Signatur durch Herausgeber und Zeugen von Aufträgen auf die entsprechenden Aufträge beschränken (siehe auch [Syntax]).

Bei der Verschlüsselung wird der zu verschlüsselnde Teil der Nachricht gegen ein nach dem XML-Encryption-Standard aufgebautes Verschlüsselungsdaten-Segment ausgetauscht, welches neben den für die Entschlüsselung benötigten Informationen auch die verschlüsselten Daten selbst enthält.

Weitere Informationen zum Signieren und Verschlüsseln sind unter [Formals], Abschnitt *II.12.4 Vorgehensweise beim Signieren und Verschlüsseln* zu finden.

Für die Übermittlung der sicherheitsrelevanten Informationen werden die folgenden Segmente und Datenelementgruppen übertragen.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 34	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung

II.5.1 Signatur-Segment

♦ Beschreibung

Das Signatur-Segment enthält den Signaturwert sowie Zusatzinformationen, die allgemein für alle im FinTS-Protokoll verwendbaren Sicherheitsverfahren benötigt werden, oder die für das jeweilige Sicherheitsverfahren spezifisch sind. Je nach verwendeter Syntax können weitere syntaxspezifische Zusatzinformationen enthalten sein (z. B. Algorithmenbezeichner).

Je nach Erfordernissen der verwendeten Syntax werden im Signatursegment auch der Hashwert und Referenzen auf die signierten Nachrichtenteile abgelegt. Auch ist es möglich, dass die verwendete Syntax ein mehrstufiges Hashing realisiert, wobei unterschiedliche Teile der Nachricht einzeln gehasht und anschließend nochmals ein Gesamt-Hashwert gebildet wird. In diesem Fall werden alle Hashwerte im Signatur-Segment gespeichert und mitsigniert.

Bei Verwendung als Botensignatur wird der gesamte Inhalt der Nachricht einschließlich aller im Signatursegment enthaltenen Hashwerte, Referenzen und Zusatzinformationen signiert, ausgenommen sind lediglich der Signaturwert selbst sowie der Bezeichner des Signaturschlüssels.

Bei Verwendung als Signatur des Auftragsteils werden ein- oder mehrere Aufträge eines Auftragsteils sowie die Hashwerte, Referenzen und Signatur-Zusatzinformationen außer dem Signaturwert selbst sowie dem Bezeichner des Signaturschlüssels signiert.

Wenn Referenzen auf Nachrichtenteile gespeichert werden, ist zu gewährleisten, dass diese bei Signaturen eines Auftragsteils unabhängig vom Kontext der Gesamtnachricht bleiben.

Allgemeine Informationen:

- Rolle des Signierenden
- Sicherheitsdatum und -uhrzeit

Für HBCI-Sicherheit spezifische Informationen:

- die verwendeten Algorithmen (siehe *II.1.1 Sicherheitsprofile*)
- Kennzeichen für Sicherheitsprofil (siehe *II.1.1 Sicherheitsprofile*)
- Signatur-ID
- Schlüsselname bestehend aus:
 - Kreditinstitutskennung
 - Benutzerkennung (bei DDV generell und bei RDH-Benutzerschlüsseln)
 - Nummer
 - Version
 - Typ
- Zertifikat

Für DDV spezifische Informationen:

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung	Stand: 09.07.2004	Seite: 35

- Kartenidentifizierung (CID) der DES-Chipkarte

Für RDH spezifische Informationen:

- Kundensystemkennung (bei softwarebasiertem RDH)
- Kartenidentifizierung (CID) der RSA-Chipkarte (bei kartenbasiertem RDH)
- Nummer und Version des dem Benutzer vorliegenden Bankschlüssels (bei Benutzersignaturen)

♦ Belegungsrichtlinien

Rolle des Signierenden

Die folgenden Codes können für die folgenden Signaturarten verwendet werden, sofern dies zwischen Benutzer und Kreditinstitut zuvor vereinbart wurde:

1. Botensignatur:

Es ist genau eine Botensignatur zu erstellen. Der Lieferant einer Botensignatur ist implizit immer Bote der Gesamtnachricht. Er kann darüberhinaus aber auch Herausgeber oder Zeuge eines darin enthaltenen Auftragsteils sein, welchen er in jedem Fall mitsigniert. Die folgenden Codes können mit den folgenden Bedeutungen verwendet werden:

- MSG: Der Sicherheitslieferant ist nur Bote der Gesamtnachricht.
- ISS: Der Sicherheitslieferant ist sowohl Bote der Gesamtnachricht als auch Herausgeber des darin enthaltenen Auftragsteils. Das Kreditinstitut signiert grundsätzlich in dieser Rolle.
- WIT: Der Sicherheitslieferant ist sowohl Bote der Gesamtnachricht als auch Zeuge des darin enthaltenen Auftragsteils.

2. Auftragssignatur:

In jedem Auftragsteil sind beliebig viele Signaturen über alle Aufträge des Auftragsteils oder eine Teilmenge davon möglich. Der Lieferant einer Auftragssignatur kann Herausgeber oder Zeuge der Aufträge sein, die er signiert hat. Die folgenden Codes können mit den folgenden Bedeutungen verwendet werden:

- ISS: Der Sicherheitslieferant ist Herausgeber der von ihm signierten Aufträge. Das Kreditinstitut signiert grundsätzlich in dieser Rolle.
- WIT: Der Sicherheitslieferant ist Zeuge der von ihm signierten Aufträge.



Kennzeichen für Sicherheitsprofil

Da das aktuelle Sicherheitsprofil aufgrund der Algorithmusbezeichner ggf. nicht eindeutig bestimmt werden kann, hat der Sicherheitslieferant hier zusätzlich ein Kennzeichen für das aktuell von ihm verwendete Sicherheitsprofil einzustellen.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 36	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung

Signatur-ID

Wert, der zur Doppeleinreichungskontrolle in die Signatur einzustellen ist.

 	<p>Wenn eine Synchronisierung der Kundensystemkennung durchgeführt wird, wird die eingestellte Signatur-ID nicht überprüft, da eine Doppeleinreichungskontrolle für eine Synchronisierungsnachricht dieses Typs nicht notwendig ist. Der Benutzer kann hier somit einen beliebigen Wert einstellen (siehe auch [Formals], Abschnitt <i>III.2 Statusprotokoll</i>)</p>
--	---



In Kreditinstitutsnachrichten kann entweder der Wert aus der Auftragssignatur verwendet werden, oder das Institut führt einen eigenen ID-Zähler.

Kartenidentifikation

Bei DDV oder kartenbasiertem RDH ist hier die CID der verwendeten Chip-Karte einzustellen.

Kundensystemkennung

Bei softwarebasiertem RDH ist hier eine zuvor vom Kreditinstitut angeforderte Kennung für das eigene Kundensystem einzustellen.

 	<p>Wenn eine Synchronisierung der Kundensystemkennung durchgeführt wird, wird die eingestellte Kundensystemkennung nicht überprüft. Der Benutzer kann hier somit einen beliebigen Wert einstellen, wenn ihm keine echte Kundensystemkennung vorliegt (siehe auch [Formals], Abschnitt <i>III.2 Statusprotokoll</i>)</p>
--	---

Nummer und Version des dem Kunden vorliegenden Bankschlüssels

Im Falle einer Signatur durch einen FinTS-Benutzer hat dieser in seiner Signatur auch die Schlüsselnummer und -version des ihm vorliegenden Bankschlüssels anzugeben. So ist für eine spätere Signatur durch das Kreditinstitut sichergestellt, mit welchen Schlüsseln diese Signatur zu erfolgen hat, damit der Benutzer sie verifizieren kann.

Im Falle von Kreditinstitutssignaturen sind die Felder leer zu lassen.

Zertifikat

Bei RDH-3 und RDH-4 in Verbindung mit mindestens einem zu signierenden Geschäftsvorfall, der Sicherheitsklasse 3 oder 4 ist ein Zertifikat Pflicht.

Bei RDH-1 und RDH-2 oder bei RDH-3 und RDH-4 in Verbindung mit zu signierenden Geschäftsvorfällen der Sicherheitsklassen 1 bis 2 ist ein Zertifikat optional.

Bei DDV-1 und DDV-2 ist ein Zertifikat nicht zulässig.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung	Stand: 09.07.2004	Seite: 37

Details der syntaktischen Umsetzung finden sich in [Syntax].

II.5.2 Verschlüsselungsdaten

♦ Beschreibung

Für die Speicherung der Sicherheitsinformationen für die Verschlüsselung werden Verschlüsselungsdaten-Segmente in die bestehende Nachricht eingefügt. Sie ersetzen jeweils die Nachrichtenteile, die sie in verschlüsselter Form enthalten. Darüberhinaus enthalten sie Zusatzinformationen, die teilweise für das jeweilige Sicherheitsverfahren spezifisch sind sowie gegebenenfalls syntaxspezifische Zusatzinformationen (z. B. Algorithmusbezeichner).

Für HBCI-Sicherheit spezifische Informationen:

- Schlüsselname bestehend aus:
 - Kreditinstitutskennung
 - Benutzerkennung (bei DDV generell und bei RDH-Benutzerschlüsseln)
 - Nummer
 - Version
 - Typ
- die verwendeten Algorithmen (siehe *II.1.1 Sicherheitsprofile*)
- Kennzeichen für Sicherheitsprofil (siehe *II.1.1 Sicherheitsprofile*)

Für DDV spezifische Informationen:

- Kartenidentifizierung (CID) der DES-Chipkarte

♦ Belegungsrichtlinien

Kennzeichen für Sicherheitsprofil

Da das aktuelle Sicherheitsprofil aufgrund der Algorithmusbezeichner ggf. nicht eindeutig bestimmt werden kann, hat der Sicherheitslieferant hier zusätzlich ein Kennzeichen für das aktuell von ihm verwendete Sicherheitsprofil einzustellen.

Details der syntaktischen Umsetzung finden sich in [Syntax].

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 38	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Formate für Signatur und Verschlüsselung

II.5.3 Komprimierungsdaten

♦ Beschreibung

Die Komprimierung wird formal als Verschlüsselung betrachtet (siehe II.5.2 *Verschlüsselungsdaten*). Als Algorithmen sind ausschließlich die vom Kreditinstitut laut BPD unterstützten Komprimierungsalgorithmen zulässig.

Details der syntaktischen Umsetzung mittels finden sich in [Syntax].

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 39

II.6 Key-Management

II.6.1 Key-Management-Nachrichten

Aufträge des Key-Managements dürfen nur in speziellen Nachrichten übertragen werden, siehe [Syntax].

Die Nachrichten für das Key-Management müssen zum Teil kryptographisch geschützt werden. Alternativ können auch Offline-Sicherungsverfahren (z. B. Brief) zum Einsatz kommen (vgl. *II.3.1.3 Asymmetrische Schlüssel für RDH*). In den Beschreibungen ist jeweils angegeben, welche Verfahren anzuwenden sind.

Für alle kryptographisch zu signierenden Nachrichten gilt: es muss mindestens eine Signatur des Schlüsseleigentümers als Auftraggebersignatur (Signatur mit Rolle ISS) vorhanden sein. Bei direkter Einreichung der Nachricht (ohne Intermediär) bringt der Auftraggeber hierfür eine Botensignatur mit dem Rollenkennzeichen ISS an, eine zusätzliche Signatur im Auftragsteil ist optional. Bei Einreichung über einen Intermediär wird die Botensignatur durch den Intermediär geleistet, so dass die Auftraggebersignatur als Signatur im Auftragsteil angebracht werden muss.

Es sind folgende Key-Management-Nachrichten vorgesehen:

- Änderung eines öffentlichen Schlüssels des Benutzers
- Erstmalige Anforderung der Schlüssel des Kreditinstituts
- Erstmalige Übermittlung der Schlüssel des Benutzers
- Schlüsselsperrung durch den Benutzer

Mit Ausnahme der Sperrnachricht sind alle Key-Management-Nachrichten nur bei Einsatz des RDH-Verfahrens möglich.

Alle Key-Management-Nachrichten müssen eine Initialisierung enthalten und stehen somit am Anfang einer FinTS-Kommunikation. Die durch sie begonnenen Kommunikationen werden stets mit der Kreditinstitutsantwort wieder beendet. Keymanagement-Nachrichten sind damit formal FinTS-Datagramme. Außer dem KeyManagement-Auftrag können in der Nachricht keine weiteren Aufträge versendet werden.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 40	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management

II.6.1.1 Änderung eines öffentlichen Schlüssels des Benutzers

Realisierung Kreditinstitut: verpflichtend

Realisierung Kundenprodukt: verpflichtend

a) Benutzernachricht

♦ Beschreibung

Diese Nachricht ist nur bei Verwendung des RDH-Verfahrens möglich. Der Änderungsauftrag muss mit dem alten Signierschlüssel signiert werden.

Es können bis zu zwei Schlüsseländerungen gleichzeitig durchgeführt werden. Der bei RDH-3 und RDH-4 vorhandene dritte Schlüssel (Schlüssel für digitale Signatur) kann nicht geändert werden. Es muss unterschieden werden, ob die Schlüsseländerung auch das Sicherheitsprofil wechselt oder nicht.

1. ohne Wechsel des Sicherheitsprofils:

Nach der erfolgreichen Durchführung der Schlüsseländerung wird der vorher aktuelle Schlüssel automatisch gesperrt und die Kreditinstitutsnachricht wird mit dem neuen Chiffrierschlüssel abgesichert. Es ist darauf zu achten, dass die Version des neuen Schlüssels höher ist als die des alten Schlüssels.

2. mit Wechsel des Sicherheitsprofils:

Bei einem Sicherheitsprofilwechsel muss der Kunde sowohl den neuen Signier- als auch den neuen Chiffrierschlüssel einstellen. Nach der erfolgreichen Durchführung der Schlüsseländerung wird durch das Kreditinstitut mitgeteilt, ob die vorher aktuellen Schlüssel automatisch gesperrt wurden. Die Kreditinstitutsnachricht wird mit den alten Schlüsseln (bzw. im alten Sicherheitsprofil) abgesichert.



Falls der Benutzer eine Schlüsseländerungsnachricht sendet, diese aber aus kreditinstitutsinternen Verarbeitungsgründen nicht beantwortet wird, sollte das Kundensystem zunächst eine neue Kommunikation auf Basis eines der Schlüsselpaare aufbauen. Falls diese Nachricht abgelehnt wird ist ein erneuter Versuch auf Basis des anderen Schlüsselpaares vorzunehmen. Aus der Reaktion des Kreditinstituts ist für das Kundensystem ersichtlich, ob die Schlüsseländerung erfolgreich war oder wiederholt werden muss.

Zum Verfahren siehe *II.3.1.3.3 Schlüsseländerungen*

♦ Belegungsrichtlinien

Verfahrensbezeichner für das Sicherheitsverfahren RDH

Es ist anzugeben, für welches Sicherheitsverfahren der neue Schlüssel verwendet werden soll.

Schlüsselname

Es ist der Name des neuen öffentlichen Schlüssels des Benutzers wie unter *II.5.2 Verschlüsselungsdaten* beschrieben einzustellen.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 41

Modulus, Exponent

Es ist der öffentliche Schlüssel des Benutzers einzustellen.

Zertifikat

Falls für den neuen öffentlichen Schlüssel ein Zertifikat verfügbar ist, kann es dem Kreditinstitut auf diese Weise eingereicht werden, sofern es für das gewählte Sicherheitsverfahren verwendet werden darf.

b) Kreditinstitutsnachricht

♦ Erläuterungen

Es werden keine Datensegmente zurückgemeldet. Die Kommunikation wird durch die Kreditinstitutsnachricht explizit beendet.

♦ Ausgewählte Beispiele für RückmeldungsCodes

Code	Beispiel
0020	Öffentlicher Schlüssel wurde geändert
3260	Schlüssel weiterhin gültig. Schlüsselsperre wird empfohlen
9010	Schlüsseländerung zur Zeit nicht möglich
9010	Sicherheitsverfahren unterstützt keine öffentlichen Schlüssel
9210	Eingereichter Schlüssel ist mit dem aktuellen Schlüssel identisch

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 42	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management

II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts

Realisierung Kreditinstitut: optional
Realisierung Kundenprodukt: verpflichtend

Diese Nachricht ist nur bei Verwendung des RDH-Verfahrens möglich. Mit Hilfe dieser Nachricht fordert der Benutzer erstmalig die öffentlichen Schlüssel des Kreditinstituts an. Da die Anforderung innerhalb eines eigenen Dialoges mit Initialisierung liegt, erhält der Benutzer die aktuellen Bankparameterdaten, die er benötigt, um die unterstützten Verschlüsselungsverfahren des Kreditinstituts in Erfahrung zu bringen. Mit Hilfe dieser Informationen wird der Benutzer in die Lage versetzt, beliebige Nachrichten zu verschlüsseln.

a) Benutzernachricht

◆ Beschreibung

Die Nachricht wird anonym gesendet. Sie wird weder signiert noch verschlüsselt.

◆ Belegungsrichtlinien

Verfahrensbezeichner für das Sicherheitsverfahren RDH

Es ist anzugeben, für welches Sicherheitsverfahren der Schlüssel angefordert werden soll.

Schlüsselname

In den Schlüsselnamen ist die Nummer und Version des Schlüssels einzustellen, den das Kundensystem als aktuellen öffentlichen Schlüssel des Kreditinstituts kennt. Falls dieser noch nicht vorliegt, sind beide Felder wegzulassen.



Da bei der Erstinitialisierung noch keine BPD vorliegt, ist es für das Kundensystem evtl. problematisch, zu ermitteln welche Sicherheitsprofile das Kreditinstitut anbietet und - wenn mehrere möglich sind - welches Profil für den Benutzer gilt. Falls dem Benutzer diese Information nicht von seinem Kreditinstitut mitgeteilt wurde, sollte das Kundensystem versuchen, das Sicherheitsmedium zu lesen und daraus das richtige Sicherheitsprofil zu erschließen.



Falls die angegebene Nummer und Version gültig sind, aber nicht mehr aktuelle Schlüssel bezeichnen, sendet das Kreditinstitut diese abgelaufenen Schlüssel, sofern es noch darüber verfügt.

In allen anderen Fällen sendet das Kreditinstitut seine aktuellen Schlüssel. Falls Nummer und Version in der Anforderung abgelaufene Schlüssel bezeichnen und das Kreditinstitut diese Schlüssel nicht senden kann, oder falls Nummer und Version ungültig sind, sollte die Antwortnachricht zusätz-

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 43

[lich zu den aktuellen Schlüsseln eine Warnmeldung enthalten.](#)

Typ

[Als Schlüsseltyp wird hier immer der Signierschlüssel angegeben.](#)



Generell sollte das Kreditinstitut abgelaufene Bankschlüssel für einen Übergangszeitraum parallel zu den neuen Bankschlüsseln verwenden können. So können auch solche Benutzernachrichten bearbeitet werden, die noch mit den abgelaufenen Schlüssel verschlüsselt wurden. In diesem Fall sollte das Kreditinstitut dem Benutzer die neuen öffentlichen Schlüssel auch ohne dessen expliziter Anforderung innerhalb der Kreditinstitutsantwort zusenden.

b) Kreditinstitutsnachricht

♦ **Erläuterungen**

Die Kommunikation wird durch die Kreditinstitutsnachricht explizit beendet.

In dieser Nachricht sind die öffentlichen Schlüssel des Kreditinstituts explizit als Auftragsantwort enthalten. Außer mit dieser Nachricht können diese Schlüssel vom Kreditinstitut bei Änderungen auch implizit im Rahmen einer normalen Initialisierungsantwort versandt werden.

Die Nachricht ist nicht verschlüsselt. Falls das Kreditinstitut einen Signierschlüssel besitzt, d. h. seine Nachrichten grundsätzlich signiert, hat es auch diese Nachricht zu signieren, um die Authentizität des Chiffrierschlüssels zu sichern (s. u.).

Falls das Kreditinstitut seine Nachrichten nicht signiert, erhält der Benutzer nur den öffentlichen Chiffrierschlüssel zurückgemeldet. Auf die Anforderung des Signierschlüssels erhält er einen entsprechenden Rückmeldungscode der Kategorie „Warnungen und Hinweise“, der ihm anzeigt, dass das Kreditinstitut seine Nachrichten nicht signiert. Da die Authentizität des Chiffrierschlüssels nicht gesichert ist, muss diese Nachricht durch einen Ini-Brief an den Benutzer mit dem Hash-Wert des Chiffrierschlüssels begleitet werden (siehe *II.3.1.3.2 Initiale Schlüsselverteilung*).

Falls das Kreditinstitut seine Nachrichten signiert, erhält der Benutzer sowohl den öffentlichen Chiffrier- als auch den Signierschlüssel zurückgemeldet. Die Authentizität des Chiffrierschlüssels ist dabei durch die Signatur gesichert. Die Authentizität des Signierschlüssels ist jedoch nicht gesichert, da das Kundensystem die Echtheit der Signatur nicht prüfen kann. Daher muss in diesem Fall die Nachricht durch einen Ini-Brief mit dem Hash-Wert des Signierschlüssels begleitet werden.

Da die Nachricht unverschlüsselt ist, werden grundsätzlich keine UPD zurück gemeldet.

♦ **Belegungsrichtlinien**

Verfahrensbezeichner für das Sicherheitsverfahren RDH

Es ist anzugeben, für welches Sicherheitsverfahren der Schlüssel übermittelt wird. Es sind nur RDH-Verfahren zulässig.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 44	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management

Schlüsselname

Der zurückgemeldete Schlüsselname enthält insbesondere die zugehörige Schlüssel- und Versionsnummer, die das Kundensystem für die Referenzierung des übertragenen neuen öffentlichen Schlüssels verwendet.

Modulus, Exponent

Der neue öffentliche Schlüssel des Kreditinstituts.

Zertifikat

Falls für den neuen öffentlichen Schlüssel ein Zertifikat verfügbar ist und für das gewählte Sicherheitsverfahren verwendet werden darf, wird es in diesem Feld übermittelt.

♦ Ausgewählte Beispiele für RückmeldungsCodes

Code	Beispiel
0020	Auftrag ausgeführt
3310	Kein Schlüssel verfügbar, da Kreditinstitutsnachrichten nicht signiert werden

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 45

II.6.1.3 Erstmalige Übermittlung der Schlüssel des Benutzers

Realisierung Kreditinstitut: verpflichtend

Realisierung Kundenprodukt: verpflichtend

Diese Nachricht ist nur bei Verwendung des RDH-Verfahrens möglich. Mit Hilfe dieser Nachricht übermittelt der Benutzer erstmalig seine öffentlichen Schlüssel an das Kreditinstitut („Erstinitialisierungsnachricht“).

Da der Absender des öffentlichen Schlüssels den Beweis erbringen muss, dass er auch im Besitz des zugehörigen privaten Schlüssels ist, muss die Nachricht des Benutzers signiert sein.



Das Kreditinstitut darf eine Nachricht nicht ablehnen, nur weil für den Benutzer noch kein öffentlicher Schlüssel in der Schlüsselverwaltung existiert. Falls die normale Signaturprüfung aus diesem Grund negativ verläuft, muss zunächst geprüft werden, ob es sich um eine Erstinitialisierung handelt. In diesem Fall ist der öffentliche Schlüssel aus der Erstinitialisierungsnachricht zu extrahieren und die Signaturprüfung auf der Basis dieses Schlüssels erneut vorzunehmen.

Die Erstinitialisierungsnachricht des Benutzers ist zu verschlüsseln, da die darin enthaltenen benutzerbezogenen Daten (Kunden-ID, Benutzerkennung) als vertraulich einzustufen sind. Dies erfordert, dass sich der öffentliche Chiffrierschlüssel des Kreditinstituts schon vor dem Senden der Erstinitialisierung im Besitz des Benutzers befinden muss. Ferner muss dem Benutzer das Verschlüsselungsverfahren bekannt sein, das ihm in den Bankparameterdaten mitgeteilt wird. Um dem Benutzer diese Daten vorab zukommen zu lassen, bieten sich folgende Lösungen an:

- Das Kreditinstitut sendet dem Benutzer eine Diskette zu, die die Schlüssel und die aktuelle BPD enthält, wie in *II.3.1.3.2 Initiale Schlüsselverteilung* beschrieben.
- Der Benutzer sendet die Key-Management-Nachricht „Erstmalige Anforderung der Schlüssel des Kreditinstituts“ (siehe *II.6.1.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts*). Diese Nachricht wird begleitet von einem Ini-Brief.



Um die wiederholte Ausführung unberechtigter Initialisierungsversuche zu verhindern, sind kreditinstitutsseitig folgende Vorkehrungen zu treffen:

- Die Benutzerkennung sollte bei Verwendung des RDH-Verfahrens nicht durch benutzerindividuelle Merkmale (z. B. Kontonummer) hergeleitet werden können.
- Eine erneute Erstinitialisierung ist nur zulässig, wenn zuvor eine Sperrung der Schlüssel des Benutzers erfolgt ist. In allen anderen Fällen ist eine erneute Erstinitialisierungsnachricht abzulehnen.

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 46	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management



Auf der Chipkarte können Kommunikationszugänge abgelegt werden (siehe *III CHIPAPPLIKATIONEN*). Da pro Kreditinstitut jedoch mehrere Kommunikationszugänge gespeichert sein können (z. B. HTTP und SMTP), muss ein Kundensystem zunächst prüfen, ob für dieses Kreditinstitut bereits die Schlüssel eingereicht wurden, bevor eine erstmalige Übermittlung der Schlüssel des Benutzers durchgeführt wird. Für den Fall, dass das Kundensystem die Schlüssel dennoch sendet, sollte das Kreditinstitut die Warnung 3330 „Schlüssel liegen bereits vor“ zurückmelden.

a) Benutzernachricht

◆ Beschreibung

Die Nachricht muss signiert und verschlüsselt werden.

Der Benutzer stellt seine öffentlichen Schlüssel ein. Dies können Signier-, Chiffrier- oder Authentifikationsschlüssel sein.

Die Authentizität des Chiffrierschlüssels ist dabei durch die Signatur gesichert. Die Authentizität des Signierschlüssels ist jedoch nicht gesichert, da das Kreditinstitut die Echtheit der Signatur nicht prüfen kann. Daher muss die Nachricht durch einen Ini-Brief an das Kreditinstitut mit dem Hash-Wert des Signierschlüssels begleitet werden (siehe *II.3.1.3.2 Initiale Schlüsselverteilung*).

◆ Belegungsrichtlinien

Sicherheitsverfahren

Es ist anzugeben, für welches Sicherheitsverfahren der Schlüssel übermittelt wird.

Schlüsselname

Der zurückgemeldete Schlüsselname enthält insbesondere die zugehörige Schlüssel- und Versionsnummer, die das Kundensystem für die Referenzierung des übertragenen neuen öffentlichen Schlüssels verwendet.

Modulus, Exponent

Diese Datenelementgruppe enthält den neuen öffentlichen Schlüssel des Kreditinstitutes.

Zertifikat

Falls für den neuen öffentlichen Schlüssel ein Zertifikat verfügbar ist und für das gewählte Sicherheitsverfahren verwendet werden darf, wird es in diesem Feld übermittelt.

b) Kreditinstitutsnachricht

Die Nachricht ist bei erfolgreicher Ausführung signiert. Sie ist stets unverschlüsselt, da der Chiffrierschlüssel des Benutzers erst nach erfolgreicher Verifikation des Ini-Briefs gültig und damit verwendbar ist.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 47

Da die Nachricht unverschlüsselt ist, werden grundsätzlich keine UPD zurück gemeldet.

◆ Beschreibung



Die Ablehnung der Erstinitialisierungsnachricht darf aus sicherheitstechnischen Aspekten im Rahmen der Rückmeldungs-codes nicht inhaltlich begründet werden. Fehlermeldungen, die sich auf den syntaktischen Aufbau der Nachricht bzw. der Segmente beziehen, sind hiervon unberührt.

◆ Erläuterungen

Es werden keine Datensegmente zurückgemeldet. Die Kommunikation wird durch die Kreditinstitutsnachricht explizit beendet.

◆ Ausgewählte Beispiele für Rückmeldungs-codes

Code	Beispiel
0010	Öffentlicher Schlüssel wurde entgegengenommen
0020	Öffentlicher Schlüssel wurde freigeschaltet
0020	Benutzer wurde freigeschaltet
3330	Schlüssel liegen bereits vor
9010	Auftrag abgelehnt

Kapitel: II	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 48	Stand: 09.07.2004	Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management

II.6.1.4 Schlüsselsperrung durch den Benutzer

Realisierung Kreditinstitut: verpflichtend
Realisierung Kundenprodukt: verpflichtend

Diese Nachricht ist sowohl bei Verwendung des RDH- wie auch des DDV-Verfahrens möglich. Sie beschreibt die Anforderung zum Sperren der Schlüssel durch den Benutzer und die Bestätigung der Schlüsselsperrung durch das Kreditinstitut (vgl. II.3.2 *Schlüsselsperrung*).

a) Benutzernachricht

◆ Beschreibung

Es werden immer alle Schlüssel gesperrt. Eine selektive Schlüsselsperrung (z. B. nur Chiffrierschlüssel) ist gegenwärtig nicht zulässig.

Die Nachricht muss signiert sein. Nicht-signierte (anonyme) Schlüsselsperrungen sind nicht vorgesehen.

Bei Verlust des Sicherheitsmediums liegen dem Benutzer u. U. die zur Durchführung der Sperrung erforderlichen Daten (Schlüsselnummer und -version) nicht vor. In diesem Fall ist die Sperre über einen anderen Weg (z. B. Callcenter) durchzuführen.

Beim RDH-Verfahren muss der Benutzer nach einer Schlüsselsperrung zur Entsperung eine erneute Erstinitialisierung durchführen.

◆ Belegungsrichtlinien

Verfahrensbezeichner für das Sicherheitsverfahren RDH

Es ist anzugeben, für welches Sicherheitsverfahren die Schlüssel gesperrt werden sollen.

Benutzerschlüsselname

Es sind die Identifikationsmerkmale des zu sperrenden Signierschlüssels einzustellen, unabhängig davon, dass grundsätzlich immer sowohl der/die Signier- als auch der Chiffrierschlüssel gesperrt werden (siehe II.3.2 *Schlüsselsperrung*).

Sperrgrund

Es ist der Grund für die Sperre anzugeben. Dies kann z. B. die Kompromittierung der Benutzerschlüssel oder der Verdacht darauf sein.

Annulierungszeitpunkt

Enthält optional Datum und Uhrzeit, ab welcher der Schlüssel nicht mehr gültig ist.



Es ist zu beachten, dass eine terminierte Sperre nicht von allen Kreditinstituten unterstützt wird. Das Kundensystem sollte den Benutzer auf diesen Sachverhalt hinweisen.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: II
Kapitel: VERFAHRENSBESCHREIBUNG Abschnitt: Key-Management	Stand: 09.07.2004	Seite: 49

b) Kreditinstitutsnachricht

♦ Erläuterungen

Beim DDV-Verfahren wird die Kommunikation im Anschluss an die Sperrnachricht ungesichert beendet, d. h. die Kreditinstitutsantwortnachricht wird weder signiert noch verschlüsselt.

Beim RDH-Verfahren wird im Anschluss an die Sperrnachricht die Antwortnachricht des Kreditinstituts nicht chiffriert, aber signiert (sofern das Kreditinstitut grundsätzlich signiert).

Diese Verfahren gelten nur bei einer erfolgreichen Sperrung. Bei einer fehlgeschlagenen Sperrung ist die Kommunikation gesichert zu Ende zu führen, da die Schlüssel des Benutzers weiterhin aktiv sind.

♦ Ausgewählte Beispiele für Rückmeldungs_codes

Code	Beispiel
0020	Schlüssel wurde erfolgreich gesperrt
9010	Schlüssel ist bereits gesperrt
9010	Terminierte Sperren werden nicht unterstützt
9210	Unbekanntes Sperrenkennzeichen
9210	Sperrdatum liegt zu weit in der Zukunft

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 51

III. CHIPAPPLIKATIONEN

III.1 Chipapplikation für RDH

Kapitel *III.1.1 Applikation Notepad* spezifiziert die Datenstrukturen und Zugriffsregeln der Chipapplikation "DF_NOTEPAD" für SECCOS-Chipkarten [SECCOS]. Kapitel *III.1.2 Terminalabläufe* spezifiziert die Terminalabläufe im Umgang mit dem RDH-Verfahren auf SECCOS-Chipkarten [SECCOS].

Im Verlauf dieses Kapitels ist mit "Bankensignaturkarte" eine Chipkarte mit SECCOS-Betriebssystem und Signaturanwendung gemeint, die u. U. auch die Notepad-Applikation aus *III.1.1 Applikation Notepad* enthält. Weitere Applikationen, wie z. B. die elektronische Geldbörse, sind nicht notwendigerweise auf der Chipkarte enthalten. Ebenso kann die Karte kontobezogen oder kontoungebunden sein.

III.1.1 Applikation Notepad

Die Anwendung „Notepad“ dient als „Notizbuch“ zur Aufnahme von Daten anderer Anwendungen. Durch das Notizbuch wird somit ein mobiler Datenspeicher geschaffen, in dem bestimmte anwendungs- bzw. benutzerspezifische Parameter abgelegt werden können, z. B. für die Bankverbindungsdaten in FinTS.

Wenn eine Anwendung auf die Karte zugreift, wird geprüft, ob auf der Chipkarte das Notizbuch DF_NOTEPAD vorhanden ist. Falls ja, werden die Daten ausgelesen, falls nein, muss der Benutzer die Zugangsdaten selbst eingeben bzw. die Zugangsdaten werden im Kundensystem selber verwaltet.

Im Datenspeicher EF_NOTEPAD kann jeder Record durch eine Anwendung belegt werden. Die Unterscheidung der Zugehörigkeit bestimmter Dateninhalte erfolgt an Hand der Tags eines Records:

- '00' bedeutet, dass der Record nicht belegt ist
- 'F0' bedeutet, dass der Record FinTS-Bankverbindungsdaten (FinTS-Parameterblock) enthält.

Weitere Kennungen sind für den späteren Gebrauch durch andere Anwendungen vorgesehen (Tag 'F1' bis 'FE').

Somit können mehrere FinTS-Bankverbindungsdaten (im Sinne der Multibankfähigkeit) in unterschiedlichen Records, jeweils mit Kennung/Tag 'F0' abgelegt werden. Jede FinTS-Bankverbindung belegt dabei einen Record.

Es werden bestimmte Zugriffs- und Authentifikationsmechanismen auf der Chipkarte genutzt, und zwar das CSA-Passwort im MF und die Möglichkeit einer Komponentenauthentifikation zwischen Terminal und Chipkarte. Dies ist erfüllt, wenn die Erweiterungen im MF aus [ZKASIG] auf der Chipkarte vorhanden sind. SECCOS-Karten mit Signatur-Anwendung (DF_SIG) erfüllen diese Voraussetzungen somit implizit.

Lesender und schreibender Zugriff auf das DF_NOTEPAD ist somit an eine vorangegangene Karteninhaber-Authentifikation mit dem Client-Server-Authentifikations-Passwort gebunden.

Das ADF der Applikation Notepad wird mit DF_NOTEPAD bezeichnet. Die Positionierung im Dateibaum der Chipkarte kann frei gewählt werden, da nur Sicherheitsmechanismen, die im MF verankert sind, genutzt werden. Eine Personalisierung

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 52	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

entweder direkt unterhalb des MF oder direkt unterhalb des ZA_MF (Memory Organizer, sofern vorhanden) wird empfohlen. Die für die Applikation relevanten DF-spezifischen Schlüssel sind im EF_KEY abgelegt, das direkt im DF_NOTEPAD enthalten ist.

In der vorliegenden Spezifikation werden zwei Security Environments verwendet:

1. Das Security Environment mit der Nummer 1 (SE #1) als Standard-SE legt die Zugriffsregeln für die Dateien der Applikation Notepad für den Zugriff an einem privaten (Signatur-)Terminal fest. Dies ist insbesondere der Standard-Anwendungsfall des „Home“-Banking. Die Kommunikation mit der Chipkarte erfolgt ohne Secure Messaging. Nach vorhergehender Benutzerauthentifikation mit dem CSA-Passwort kann auf das Notepad lesend und schreibend zugegriffen werden.
2. Das Security Environment mit der Nummer 2 (SE #2) legt die Zugriffsregeln für die Dateien und das Applikationsverzeichnis der Applikation Notepad für den Einsatz
 - an einem öffentlichen Signatur-, Geschäfts- oder Administrationsterminal
 - an einem Terminal zur Administration der Notepad-Anwendung

fest.

Ein Signatur-Geschäftsterminal führt mit der Chipkarte eine Komponenten-authentifikation durch, die Kartenkommandos und -antworten werden mit einem so ausgehandelten Session-Key (SK2) mit Secure Messaging abgesichert [ZKASIG].

Zur Administration der Notepad-spezifischen Daten der Karte ist die Kenntnis eines (symmetrischen) Administrationsschlüssels $K_{\text{Notepad_Admin}}$ nötig.

Die Selektion von SEs erfolgt, wie in [SECCOS] beschrieben, mit dem Kommando `MANAGE SECURITY ENVIRONMENT`. Für den Standard-Anwendungsfall, d. h. an privaten Signatur-Terminals, ist eine Selektion des SE nicht notwendig, da bereits mit der Selektion einer Applikation implizit das SE #1 aktiviert wird.

Das DF_NOTEPAD ist optional.

Financial Transaction Services (FinTS)	Version:	Kapitel:
Dokument: Security - Sicherheitsverfahren HBCI	4.0	III
Kapitel: CHIPAPPLIKATIONEN	Stand:	Seite:
Abschnitt: Chipapplikation für RDH	09.07.2004	53

III.1.1.1 Daten der Applikation Notepad

Die folgende Abbildung gibt eine Übersicht über die Dateien einer Bankensignaturkarte mit der Applikation Notepad (beispielhaft direkt unterhalb des MF angelegt).

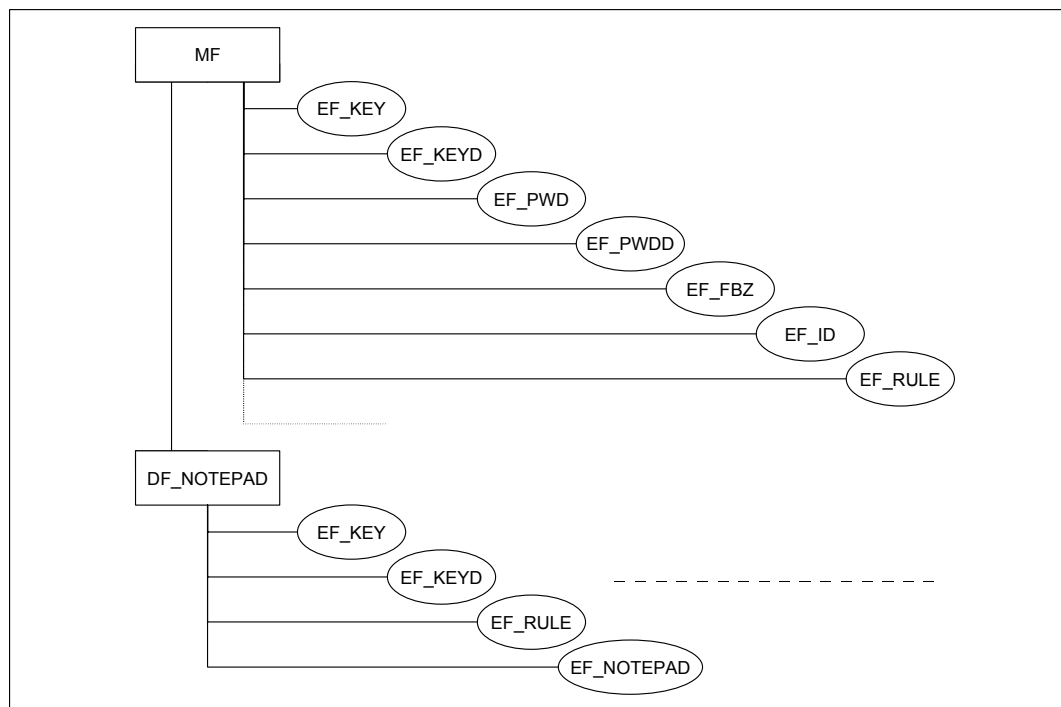


Abbildung 10: Datenelemente der Applikation „Notepad“

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 54	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.1.1 ADF der Applikation Notepad

Für das ADF der Applikation Notepad (DF_NOTEPAD) sind beim Anlegen die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1A'		Tag und Länge für FCP
'82'	'01'	'38'	Datei-Deskriptor für DF
'83'	'02'	'A6 10'	Datei-ID des DF_NOTEPAD
'84'	'09'	'D2 76 00 00 25 4E 50 01 00'	DF-Name (AID) des DF_NOTEPAD
'A1'	'06'	'8B 04 00 30 02 01'	Zugriffsregel-Referenzen

Der DF-Name (die AID) des DF_NOTEPAD bestehend aus der nationalen RID des ZKA ('D2 76 00 00 25'), der ASCII-kodierten Kennung "NP" (Note-Pad '4E 50') sowie der Version der Applikation 1.0 ('01 00').

Die Zugriffsregeln für das DF_NOTEPAD stehen in der zugeordneten Regeldatei EF_RULE. Durch die Zugriffsregeln werden für die DF-spezifischen Kommandos die folgenden Festlegungen getroffen:

Wenn das DF_NOTEPAD selektiert ist, darf ein DELETE FILE (self), INCLUDE, EXCLUDE oder CREATE FILE (EF) nur ausgeführt werden, wenn die Kommandonachricht mit Secure Messaging ausgeführt wird und mit einem korrekten MAC versehen ist, der unter Verwendung des Schlüssels $K_{\text{Notepad_Admin}}$ aus dem EF_KEY des DF_NOTEPAD gebildet ist. Der Returncode wird für jedes dieser Kommandos durch die Karte mit einem MAC mit dem Schlüssel $K_{\text{Notepad_Admin}}$ versehen.

Die Kommandos INCLUDE, EXCLUDE, CREATE FILE (DF/EF) und DELETE FILE (child DF) dürfen nie ausgeführt werden. Alle zulässigen Administrationskommandos dürfen nur im SE #2 ausgeführt werden (Zugriffsregeln im Record 1 des EF_RULE).

Der Applikation Notepad sind 5 Dateien als AEF zuzuordnen:

SFI '01': EF_RULE im DF_NOTEPAD
SFI '02': EF_KEY im DF_NOTEPAD,
SFI '19': EF_ID im MF,
SFI '1A': EF_NOTEPAD im DF_NOTEPAD,
SFI '1E': EF_KEYD im DF_NOTEPAD.

Wenn das DF_NOTEPAD mittels SELECT FILE selektiert wird und die entsprechende Option im Parameterbyte P2 des Kommandos gesetzt ist, wird die folgende FCI ausgegeben:

Tag	Länge	Wert	Erläuterung
'6F'	'0D'		Tag und Länge für FCI
'84'	'09'	'D2 76 00 00 25 4E 50 01 00'	DF-Name (AID) des DF_NOTEPAD
'A5'	'00'		keine proprietären Informationen

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH		Stand: 09.07.2004	Seite: 55

Wird das DF_NOTEPAD mittels SELECT FILE selektiert und die entsprechende Option im Parameterbyte P2 des Kommandos gesetzt, werden die folgenden FMD mit den Pfaden der AEFs ausgegeben (hier beispielhaft: das DF_NOTEPAD befindet sich direkt unterhalb des MF):

Tag	Länge	Wert	Erläuterung
'64'	'21'		Tag und Länge für FMD
'85'	'03'	'C8 00 03'	Pfad für AEF mit SFI '19' (EF_ID im MF)
'85'	'05'	'08 <u>A6 10</u> 00 30'	Pfad für AEF mit SFI '01' (EF_RULE im DF_NOTEPAD)
'85'	'05'	'10 <u>A6 10</u> 00 10'	Pfad für AEF mit SFI '02' (EF_KEY im DF_NOTEPAD)
'85'	'05'	'D0 <u>A6 10 A6 11</u> '	Pfad für AEF mit SFI '1A' (EF_NOTEPAD im DF_NOTEPAD)
'85'	'05'	'F0 <u>A6 10</u> 00 13'	Pfad für AEF mit SFI '1E' (EF_KEYD im DF_NOTEPAD)

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 56	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.1.2 EF_RULE

Die Datei EF_RULE enthält die Zugriffsregeln für die Applikation DF_NOTEPAD. In den FCP von Dateien und Verzeichnissen wird auf diese Zugriffsregeln referenziert.

◆ FCP

Für das EF_RULE des DF_NOTEPAD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		Tag und Länge für FCP
'82'	'05'	'14 41 00 2A 07'	Datei-Deskriptor für lineares EF mit variabler Recordlänge (max. 42 Byte), 7 Records
'83'	'02'	'00 30'	Datei-ID des EF_RULE
'85'	'02'	'00 95'	für Nutzdaten allozierter Speicherplatz in Byte (149 Byte)
'88'	'01'	'08'	SFI '01' für das EF_RULE
'A1'	'08'	'8B 06 00 30 01 02 02 03'	Zugriffsregel-Referenzen

In SE #1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 dürfen die Kommandos READ / SEARCH RECORD, SELECT FILE (EF) ohne Einhaltung von Zugriffsbedingungen oder mit Secure Messaging durchgeführt werden. Die MAC-Bildung im Secure Messaging erfolgt für Kommando- und Antwortnachricht mit dem Session-Key SK2 (Zugriffsregel im Record 3 des EF_RULE).

◆ Daten

Das EF_RULE im DF_NOTEPAD enthält 7 Records mit den Zugriffsregeln für das Verzeichnis und die Datenfelder des Verzeichnisses.

Die folgende Tabelle gibt einen Überblick über die Referenzierung der einzelnen Records in den ZRR-DO der FCP der einzelnen Dateien und die Bytelängen der Records.

Record	SE #1	SE #2	Länge
1		DF_NOTEPAD	10
2	EF_RULE EF_KEYD		5
3		EF_RULE	29
4	EF_NOTEPAD		12
5		EF_NOTEPAD	42
6		EF_KEY	22
7		EF_KEYD	29
Summe			149

Im folgenden werden die einzelnen Records des EF_RULE näher erläutert.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 57

Record 1 wird referenziert als Zugriffsregel von DF_NOTEPAD in SE #2.

DELETE FILE (self) , INCLUDE, EXCLUDE, CREATE FILE (EF): MAC-SM-AC für Kommando- und Antwortnachricht mit $K_{\text{Notepad_Admin}}$:

Tag	Länge	Wert	Erläuterung
'80'	'01'	'DA'	Zugriffsart für DELETE FILE (self), INCLUDE, EXCLUDE, CREATE FILE (EF)
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$

Record 2 wird referenziert als Zugriffsregel von EF_RULE und EF_KEYD in SE #1.

READ / SEARCH RECORD: ALW

Tag	Länge	Wert	Erläuterung
'80'	'01'	'81'	Zugriffsart für Read / Search Record
'90'	'00'		Zugriffsbedingung ALW

Record 3 wird referenziert als Zugriffsregel von EF_RULE in SE #2.

READ / SEARCH RECORD, SELECT FILE (EF): ALW oder MAC mit SK2.

APPEND RECORD, UPDATE RECORD: MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel $K_{\text{Notepad_Admin}}$.

Tag	Länge	Wert	Erläuterung
'80'	'01'	'81'	Zugriffsart für Read / Search Record
'90'	'00'		ALW
'80'	'01'	'81'	Zugriffsart für Read / Search Record
'8C'	'02'	'0C A4'	Zugriffsart für Select File mit Secure Messaging (CLA, INS angegeben)
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'00 0D FF'	Schlüsselreferenz für SK2
'80'	'01'	'86'	Zugriffsart für Append Record, Update Record
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$

Record 4 wird referenziert als Zugriffsregel von EF_NOTEPAD in SE #1.

READ, SEARCH und UPDATE RECORD: Karteninhaberauthentifikation (PWD) mit globalem CSA-Passwort (Nummer 3).

Tag	Länge	Wert	Erläuterung
'80'	'01'	'83'	Zugriffsart für Read / Search / Update Record
'A4'	'07'		AT - Tag und Länge
'95'	'01'	'08'	Usage Qualifier für Karteninhaber-Authentifikation
'83'	'02'	'00 03'	Passwort-Referenz, globales Passwort mit Nr. 3

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 58	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

Record 5 wird referenziert als Zugriffsregel von EF_NOTEPAD in SE #2.

READ, SEARCH und UPDATE RECORD: Karteninhaberauthentifikation (PWD) mit globalem CSA-Passwort (Nummer 3) und MAC mit SK2; **oder** MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel $K_{\text{Notepad_Admin}}$.

APPEND RECORD: MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel $K_{\text{Notepad_Admin}}$.

SELECT FILE (EF): ALW oder MAC mit SK2.

Tag	Länge	Wert	Erläuterung
'80'	'01'	'83'	Zugriffsart für Read / Search / Update Record
'AF'	'10'		AND-Template
'A4'	'07'		AT – Tag und Länge
'95'	'01'	'08'	Usage Qualifier für Karteninhaber-Authentifikation
'83'	'02'	'00 03'	Passwort-Referenz, globales Passwort mit Nr. 3
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'00 0D FF'	Schlüsselreferenz für SK2
'80'	'01'	'87'	Zugriffsart für Read / Search / Update / Append Record
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$
'8C'	'02'	'0C A4'	Zugriffsart für Select File mit Secure Messaging (CLA, INS angegeben)
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'00 0D FF'	Schlüsselreferenz für SK2

Record 6 wird referenziert als Zugriffsregel von EF_KEY in SE #2.

READ / SEARCH RECORD: NEV

APPEND RECORD, UPDATE RECORD: MAC-ENC-SM-AC für Kommandonachricht und MAC-SM-AC für Antwortnachricht mit $K_{\text{Notepad_Admin}}$.

Tag	Länge	Wert	Erläuterung
'80'	'01'	'86'	Zugriffsart für Append Record, Update Record
'AF'	'11'		AND- Template, Tag und Länge
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$
'B8'	'08'		CT – Tag und Länge
'95'	'01'	'10'	Usage Qualifier: Nur für Kommandonachricht
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 59

Record 7 wird referenziert als Zugriffsregel von EF_KEYD in SE #2.

READ / SEARCH RECORD, SELECT FILE (EF): ALW oder MAC mit SK2

APPEND RECORD, UPDATE RECORD: MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel $K_{\text{Notepad_Admin}}$.

Tag	Länge	Wert	Erläuterung
'80'	'01'	'81'	Zugriffsart für Read / Search Record
'90'	'00'		ALW
'80'	'01'	'81'	Zugriffsart für Read / Search Record
'8C'	'02'	'0C A4'	Zugriffsart für Select File mit Secure Messaging (CLA, INS angegeben)
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'00 0D FF'	Schlüsselreferenz für SK2
'80'	'01'	'86'	Zugriffsart für Append Record, Update Record
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für $K_{\text{Notepad_Admin}}$

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 60	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.1.3 EF_KEY

Die applikationsspezifischen Schlüssel der Applikation Notepad sind im EF_KEY des Applikationsverzeichnisses DF_NOTEPAD gespeichert. Dies ist

- ein 16 Byte langer kartenindividueller symmetrischer Schlüssel $K_{\text{Notepad_Admin}}$ mit der Schlüsselnummer '01' zur Administration der Applikation DF_NOTEPAD.

Der Schlüssel $K_{\text{Notepad_Admin}}$ ist nur der Chipkarte und dem für sie zuständigen Hintergrundsystem bekannt. Sie werden jeweils aus einem KGK (Key Generating Key) unter Verwendung der Kartenidentifikationsdaten im EF_ID des MF abgeleitet (vgl. Kapitel 8.4.1 von [SECCOS]). Das zuständige Hintergrundsystem kennt die jeweiligen KGK und leitet die kartenindividuellen Schlüssel bei Bedarf ab.

Es können pro logischer Schlüsselnummer verschiedene KGK verwendet werden. Ein KGK wird wie alle daraus abgeleiteten Schlüssel anhand der Schlüsselversion identifiziert. Die Schlüsselversion zur jeweiligen logischen Schlüsselnummer im zugehörigen EF_KEYD zeigt an, aus welchem KGK der jeweilige kartenindividuelle Schlüssel abgeleitet ist.

◆ FCP

Für das EF_KEY des DF_NOTEPAD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'16'		Tag und Länge für FCP
'82'	'05'	'12 41 00 12 01'	Datei-Deskriptor für lineares EF mit fester Recordlänge (18 Byte), 1 Record
'83'	'02'	'00 10'	Datei-ID des EF_KEY
'88'	'01'	'10'	SFI '02' für das EF_KEY
'A1'	'06'	'8B 04 00 30 02 06'	Zugriffsregel-Referenzen

Auf das EF_KEY darf nur im SE #2 zugegriffen werden.

Die Kommandos APPEND RECORD und UPDATE RECORD dürfen nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden, der Recordinhalt verschlüsselt (ENC) ist und die Kommandonachricht mit einem MAC abgesichert ist. Verschlüsselung und MAC-Bildung erfolgen mit dem $K_{\text{Notepad_Admin}}$. Der Returncode eines APPEND RECORD oder UPDATE RECORD wird mit dem $K_{\text{Notepad_Admin}}$ MAC-gesichert. Das Kommando READ RECORD darf nie ausgeführt werden. (Zugriffsregel im Record 6 des EF_RULE)

◆ Daten

Das EF_KEY im DF_NOTEPAD enthält 1 Record mit dem DF-spezifischen Schlüssel des DF_NOTEPAD.

Logische Schlüsselnummer	Schlüssel-Version	Schlüssel
'01'	'XX'	16 Byte langer $K_{\text{Notepad_Admin}}$

Die konkret zu verwendende Schlüsselversion wird in diesem Dokument nicht festgelegt.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 61

III.1.1.1.4 EF_KEYD

Das EF_KEYD im DF_NOTEPAD enthält die Zusatzinformationen zu den DF-spezifischen Schlüsseln des DF_NOTEPAD.

◆ FCP

Für das EF_KEYD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		Tag und Länge für FCP
'82'	'05'	'14 41 00 1F 01'	Datei-Deskriptor für lineares EF mit variabler Recordlänge (max. 31 Byte) und 1 Record
'83'	'02'	'00 13'	Datei-ID des EF_KEYD
'85'	'02'	'00 1F'	für Nutzdaten allozierter Speicherplatz in Byte
'88'	'01'	'F0'	SFI '1E' für das EF_KEYD
'A1'	'08'	'8B 06 00 30 01 02 02 07'	Zugriffsregel-Referenzen

Im SE #1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 dürfen die Kommandos READ / SEARCH RECORD, SELECT FILE (EF) ohne Einhaltung von Zugriffsbedingungen oder mit Secure Messaging durchgeführt werden. Die MAC-Bildung im Secure Messaging erfolgt für Kommando- und Antwortnachricht mit dem Session-Key SK2.

Im SE #2 dürfen die Kommandos APPEND RECORD und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem $K_{\text{Notepad_Admin}}$ (Zugriffsregel im Record 7 des EF_RULE).

◆ Daten

Das EF_KEYD enthält 1 Record mit den Zusatzinformation zu dem DF-spezifischen Schlüssel des DF_NOTEPAD.

Das Datenobjekt mit Tag '93' enthält im Wertfeld als zweites Byte die Version des entsprechenden Schlüssels.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 62	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

Im folgenden wird der Aufbau der Schlüsselzusatzinformation zu $K_{\text{Notepad_Admin}}$ dargestellt:

Tag	Länge	Wert	Erläuterung
'93'	'02'	'01 XX'	Schlüsselnummer und Schlüssel-Version
'C0'	'02'	'81 10'	Symmetrischer Schlüssel der Länge 16 Byte
'90'	'01'	'FF'	Fehlbedienungs-zähler
'7B'	'12'		SE-Datenobjekt
'80'	'01'	'02'	Festlegung für SE #2
'B4'	'07'		CCT - Tag und Länge (Usage Qualifier '30' ist Defaultwert)
'89'	'02'	'12 22'	Algorithmus-ID: Schlüssel darf zur Bildung eines Retail-MAC im CFB-Mode verwendet werden
'87'	'01'	'02'	ICV-Indikator: Script-ICV für SM der Kommando- und Antwortnachricht
'B8'	'04'		CT - Tag und Länge (Usage Qualifier '10' ist Defaultwert)
'89'	'02'	'11 23'	Algorithmus-ID: Schlüssel darf zur Verschlüsselung als Triple-DES Schlüssel im CBC-Mode mit $ICV \neq 0$ und ICV-Variante verwendet werden

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 63

III.1.1.1.5 EF_NOTEPAD

Bei dem EF_NOTEPAD handelt es sich um ein lineares EF mit einer variablen Recordlänge, die aus technischen Gründen auf maximal 239¹ Byte begrenzt ist, in dem Informationen abgelegt sind. Damit können in der konkreten Umsetzung nicht alle Felder bzw. nicht alle Felder in der maximalen Länge personalisiert werden. Andernfalls würde maximale Recordlänge überschritten. Da verschiedene Anwendungen das Notepad nutzen können, werden in den ersten beiden Bytes Tag und Länge der Daten abgelegt, die eine Zuordnung erlauben.

So kann beispielsweise ein Homebanking-Kundenprodukt in einem Record die vom Aufbau her festgelegte FinTS-Benutzerkennung ablegen, in einem anderen Record aber auch (produktspezifische) Informationen zu Benutzerpräferenzen und -einstellungen (z.B. Sprache, Anzeigeparameter etc.).

Der Inhalt des Notepad kann im wesentlichen nur nach vorhergehender CSA-Passwort-Eingabe gelesen und verändert werden. Somit ist der Inhalt insbesondere vor unberechtigtem Auslesen geschützt (z.B. wenn die Kontonummer als Bestandteil der Benutzerkennung gespeichert ist).

Nach vorhergehender Benutzerauthentikation mit dem CSA-Passwort kann auf das Notepad lesend und schreibend zugegriffen werden. Das Auslesen der FinTS-Parameterblöcke erfolgt über ein *Read Record* auf alle vorhandenen Records und anschließendem Vergleich, ob der Tag des Inhalts 'F0' ist.

Alternativ können über ein SEARCH RECORD mit dem Suchmuster 'F0' für das erste Byte des Recordinhalts genau die relevanten (und anschließend auszulesenden) Recordnummern ermittelt werden.

◆ FCP

Für das EF_NOTEPAD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		Tag und Länge für FCP
'82'	'05'	'14 41 00 EF XX'	Datei-Deskriptor für lineares EF mit variabler Recordlänge bis zu 2390 Byte und XX Records
'83'	'02'	<u>'A6 11'</u>	Datei-ID des EF_NOTEPAD
'85'	'02'	'YY YY'	für Nutzdaten allozierter Speicherplatz in Byte (XX Records mal 239 Byte)
'88'	'01'	'D0'	SFI '1A' für das EF_NOTEPAD
'A1'	'08'	'8B 06 00 30 01 04 02 05'	Zugriffsregel-Referenzen

Die Anzahl der Records wird durch die Personalisierung festgelegt.

Beispiel: für XX = '05' Records ist YY YY = '04 AB'.

Im SE #1 dürfen READ, SEARCH und UPDATE RECORD nur ausgeführt werden, wenn zuvor eine Karteninhaberauthentikation mit dem globalen Passwort 3 (CSA-Passwort) erfolgt ist. Der Returncode wird nicht MAC-gesichert (Zugriffsregeln im Record 4 des EF_RULE).

¹ Nach ISO 7816-4 ist eine APDU maximal 255 Bytes lang. Nach Abzug der Protokolldaten steht eine netto Datenlänge von maximal 239 Byte zur Verfügung.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 64	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

Im SE #2 dürfen die Kommandos READ, SEARCH und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden. **Entweder** ist zuvor eine Karteninhaberauthentikation mit dem globalen Passwort 3 (CSA-Passwort) erfolgt und die MAC-Bildung im Secure Messaging erfolgt für Kommando- und Antwortnachricht mit dem Sessionkey SK2; **oder** (ohne vorherige Karteninhaberauthentikation) die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem $K_{\text{Notepad_Admin}}$ (Zugriffsregel im Record 5 des EF_RULE).

Im SE #2 darf das Kommando APPEND RECORD nur mit Secure Messaging durchgeführt werden. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem $K_{\text{Notepad_Admin}}$.

Im SE #2 darf das Kommando SELECT FILE (EF) ohne Einhaltung von Zugriffsbedingungen oder mit Secure Messaging durchgeführt werden. Die MAC-Bildung im Secure Messaging erfolgt für Kommando- und Antwortnachricht mit dem Sessionkey SK2.

◆ Daten

Byte	Länge	Wert	Erläuterung
1-2	2	'XX L'	Tag und Länge
3-(L+2)	L	'XX..XX'	Nutzdaten

Als Tags werden festgelegt:

Byte 1	Bedeutung
'00'	freier Record
'F0'	Belegung mit FinTS-Parameterblock
'F1'-'FE'	RFU

Durch den Tag 'F0' wird ein Recordeintrag für die FinTS-Anwendung gekennzeichnet. Für Belegungen der EF_NOTEPAD-Records durch andere Anwendungen stehen die Tags 'F1' bis 'FE' zur Verfügung. Die Kennungen werden durch den ZKA vergeben.

Initial werden alle Records mit '00..00' belegt.

◆ Beispiel

In der folgenden Tabelle ist die beispielhafte Belegung eines EF_NOTEPAD mit 7 Records angegeben.

Record	Eintrag	Erläuterung
1	'F0 XX...XX'	Erste FinTS-Bankverbindung
2	'F0 XX...XX'	Zweite FinTS-Bankverbindung
3	'F0 XX...XX'	Dritte FinTS-Bankverbindung
4	'00..00'	frei
5	'F1 XX..XX'	belegt durch Anwendung mit Kennung 'F1'
6	'00..00'	frei
7	'F0 XX...XX'	Vierte FinTS-Bankverbindung

◆ Umgang mit variablen Recordlängen

Durch die Definition des EF_NOTEPAD als lineares EF mit variabler Recordlänge werden beim Lesen eines Records nur die tatsächlich vorhandenen Daten ausgegeben.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH		Stand: 09.07.2004	Seite: 65

Command APDU eines READ RECORD:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'0X'	P1, Recordnummer X
4	'D4'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die Chipkarte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Länge	Wert	Erläuterung
1-L	L	'XX ...XX'	Recordeintrag
(L+1)-(L+2)	2	'SW1 SW2'	Positiver Returncode SW1 SW2

Ein FinTS-Recordeintrag beginnt in diesem Fall mit dem Tag 'F0' und dem Längenbyte (L-2).

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 66	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.2 Recordbelegung des EF_NOTEPAD mit einem FinTS-Parameterblock

Ein FinTS-Recordseintrag hat folgenden prinzipiellen Aufbau:

Tag	Länge (Byte)	Wert	For- mat	Sta- tus	Erläuterung
'F0'	Var.				FinTS-Parameterblock
'E1'	Var. max. '5B'			M	FinTS-Institutparameterblock
'C1'	'01'-'14'	Kreditinstituts- bezeichnung	..20an	O	
'C2'	'03'	Länderkenn- zeichen	3an	M	ISO 3166 numerisch in 3 ASCII-Zeichen codiert
'C3'	'01'-'1E'	Kreditinstitutscode	..30an	M	in jeweils national bekannter Notation
'C4'	'1B'	Hashwert Instituts- schlüssel	27bin	O	Hashwert des Signierschlüs- sels, falls das Institut signiert, anderenfalls Hashwert des Chiffrierschlüssels
'C5'	'01'	Schlüsselstatus	1bin	M	8 Statusflags
'E2'	Var. max. '37'			M	FinTS-Kommunikations- parameterblock
'C6'	'01'	Kommunikations- dienst	1n	M	„32“ = TCP/IP „33“ = HTTPS „34“ = SMTP „35“ = HTTP „36“ = SOAPHTTP
'C7'	'01'-'32'	Kommunikations- adresse	..50an	M	
			°		
'E2'	Var. max. '37'			O	2. FinTS-Kommunikations- parameterblock
'C6'	'01'	Kommunikations- dienst	1n	M	„32“ = TCP/IP „33“ = HTTPS „34“ = SMTP „35“ = HTTP „36“ = SOAPHTTP
'C7'	'01'-'32'	Kommunikations- adresse	..50an	M	
'E3'	Var. max. '4E'			O	FinTS-Benutzerparameterblock
'C8'	'01'-'1E'	Benutzerkennung	..30an	M	
'C9'	'01'-'1E'	Kunden-ID	..30an	O	
'CA'	'0C'	Info Inhaber- schlüssel	12n	M	Schlüsselnummer und Schlüs- selversion des Signierschlüs- sels des Karteninhabers Schlüsselnummer und Schlüs- selversion des Chiffrierschlüs- sels des Karteninhabers

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 67

Die Längen der einzelnen Records werden wie folgt nach ASN.1 BER (Basic Encoding Rules) kodiert:

Längen 'XX', wobei 'XX' die hexadezimale Darstellung eines Wertes zwischen 0 und 127 ist, werden als 'XX' in ein Byte kodiert werden.

Längen 'XX', wobei 'XX' die hexadezimale Darstellung eines Wertes zwischen 128 und 255 ist, müssen als '81 XX' in zwei Byte kodiert werden

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 68	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.2.1 Tag 'F0': FinTS-Parameterblock

Durch den Tag 'F0' wird ein Recordeintrag für die FinTS-Anwendung gekennzeichnet. Für Belegungen der EF_NOTEPAD-Records durch andere Anwendungen stehen die Tags 'F1' bis 'FE' zur Verfügung.

Ein FinTS-Parameterblock enthält in der angegebenen Reihenfolge:

- genau einen FinTS-Institutsparameterblock mit **Tag 'E1'**
- einen oder zwei FinTS-Kommunikationsparameterblöcke mit **Tag 'E2'** (der erste FinTS-Kommunikationsparameterblock ist also verpflichtend, der zweite ist optional)
- einen optionalen FinTS-Benutzerparameterblock mit **Tag 'E3'**

Die Länge des FinTS-Parameterblocks wird beschränkt durch die maximale Recordlänge.

III.1.1.2.2 Tag 'E1': FinTS-Institutsparameterblock

Durch den Tag 'E1' wird der Block der institutsspezifischen Parameter gekennzeichnet. Ein FinTS-Institutsparameterblock enthält in der angegebenen Reihenfolge:

- **optional** eine Kreditinstitutsbezeichnung mit **Tag 'C1'**, alphanumerisch mit bis zu 20 Zeichen
- genau ein Länderkennzeichen des kontoführenden Instituts mit **Tag 'C2'**. Verwendet wird der numerische ISO 3166-Code als 3-stellige alphanumerische Zeichenkette (z.B. Deutschland = "280")
- genau eine Kreditinstitutskennung mit **Tag 'C3'**, in einer jeweils national bekannten Notation mit bis zu 30 Stellen. Für deutsche Kreditinstitute wird hier die 8-stellige Bankleitzahl verwendet.
- **optional** einen Hashwert des öffentlichen Signierschlüssels des Instituts mit **Tag 'C4'**, binär mit genau 23 Byte. Der Eintrag besteht aus

[3 Byte Schlüsselnummer | 3 Byte Schlüsselversion | 1 Byte Kennzeichen Hashverfahren | 20 Byte Hashwert].

Als Kennzeichen für das Hashverfahren werden festgelegt:

- '01' = SHA-1² (impliziert Sicherheitsverfahren RDH-4)
- '02' = RIPEMD-160 (impliziert Sicherheitsverfahren RDH-3)

Die Parameter Schlüsselnummer und Schlüsselversion des Institutsschlüssels werden in je 3 Byte rechtsbündig mit führenden Nullen codiert (z.B. Schlüsselnummer 1 → die Bytefolge '30' '30' '31').

- genau ein Schlüsselstatus mit **Tag 'C5'**, binär von genau 1 Byte Länge. Der Schlüsselstatus enthält acht Flags mit folgender Bedeutung:

Bit1	Erstmalige Übermittlung der Benutzerschlüssel notwendig	'1'b - Ja '0'b - Nein
------	---	--------------------------

² Informativ: Die Festlegung '01' = SHA-1 entspricht dem Hash Algorithm Indicator in [EMV 2000, Book 2, Annex B3.1].

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH		Stand: 09.07.2004	Seite: 69

Bit2	Institutsrechner erwartet Signaturen nach ISO9796 mit AnnexA	'1'b - Ja '0'b - Nein
Bit3	Hashwert für Institutsschlüssel vorhanden	'1'b - Ja '0'b - Nein
Bit4	Ausstehende Übermittlung des neuen öffentlichen Chiffrierschlüssels des Instituts	'1'b - Ja '0'b - Nein
Bit5	Ausstehende Übermittlung des neuen öffentlichen Signierschlüssels des Instituts	'1'b - Ja '0'b - Nein
Bit6	Schlüsselsperre mit Erfolg durchgeführt (Info, da terminierte Sperrung erst in der Zukunft wirksam werden kann)	'1'b - Ja '0'b - Nein
Bit7	Leitungsprobleme bei Übermittlung neuer Schlüssel	'1'b - Ja '0'b - Nein
Bit8	Reserviert	'0'b

Bei der Personalisierung muss als Initialisierungswert '01' aufgebracht werden.

Ein FinTS-Institutsparemeterblock belegt inklusive der Tag- und Längenbytes somit maximal 93 Byte.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 70	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.1.2.3 Tag 'E2':FinTS -Kommunikationsparameterblock

Durch den Tag 'E2' wird der Block der generellen Kommunikations-Parameter gekennzeichnet. Ein FinTS -Kommunikationsparameterblock enthält in der angegebenen Reihenfolge:

- genau einen Kommunikationsdienst mit **Tag 'C6'**, 1 Stelle numerisch ([gültige Codierungen siehe III.1.1.2](#)).
- genau eine Kommunikationsadresse mit **Tag 'C7'**, alphanumerisch mit bis zu 50 Zeichen

Ein FinTS-Kommunikationsparameterblock belegt inklusive der Tag- und Längenbytes somit maximal 57 Byte.

III.1.1.2.4 Tag 'E3':FinTS-Benutzerparameterblock

Durch den optionalen Tag 'E3' wird der Block der benutzerspezifischen Parameter gekennzeichnet. Ist der Tag nicht vorhanden, so handelt es sich um eine im Rahmen der FinTS-Anwendung unpersonalisierte Karte. Ein FinTS-Benutzerparameterblock – sofern vorhanden – enthält in der angegebenen Reihenfolge:

- genau eine Benutzerkennung mit **Tag 'C8'**, alphanumerisch mit bis zu 30 Zeichen
- **optional** eine Kunden-ID mit **Tag 'C9'**, alphanumerisch mit bis zu 30 Zeichen
- genau ein Info Inhaberschlüssel mit **Tag 'CA'**, von genau 12 Byte Länge:

[Schlüsselnummer Signierschlüssel | Schlüsselversion Signierschlüssel | Schlüsselnummer Chiffrierschlüssel | Schlüsselversion Chiffrierschlüssel]

Die Parameter Schlüsselnummer und Schlüsselversion für jeden der beiden Schlüssel werden in je 3 Byte rechtsbündig mit führenden Nullen codiert (z.B. Schlüsselnummer 1 → die Bytefolge '30' '30' '31').

Ein FinTS-Benutzerparameterblock belegt inklusive der Tag- und Längenbytes somit maximal 80 Byte.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 71

III.1.2 Terminalabläufe

Dieses Kapitel spezifiziert die Terminalabläufe im Umgang mit dem RDH-Verfahren auf SECCOS-Chipkarten [SECCOS]. Ein Homebanking-Kundensystem nutzt

- zur Verschlüsselung und Signierung von FinTS-Nachrichten die auf der Chipkarte zur Verfügung stehende Signatur-Anwendung (DF_SIG, [ZKASIG]) und die durch das Betriebssystem bereitgestellten Signatur-Funktionen,
- als Sequenzzähler (Signatur-ID) interne Bedienungszähler der Signatur-Anwendung (siehe *III.1.2.1 Verfahren zur Ermittlung der Sequenzzähler*),
- als Datenspeicher für die Zugangsdaten ein auf der Chipkarte optional vorhandenes DF_NOTEPAD ([NOTEPAD], siehe *III.1.1 Applikation Notepad*).

III.1.2.1 Verfahren zur Ermittlung der Sequenzzähler

Auf der Bankensignaturkarte wird für FinTS kein eigenständiger Sequenzzähler (wie das EF_SEQ im DDV-Verfahren) verwaltet, sondern es werden jeweils chipkarteninterne Bedienungszähler der beiden zur Signatur von FinTS-Nachrichten verwendeten Schlüssel $S_{K.CH.DS}$ und $S_{K.CH.AUT_{C/S}}$ herangezogen.

Für jedes Signaturschlüsselpaar wird ein separater Sequenzzähler verwaltet.

Da die Bedienungszähler auf der Chipkarte dekrementiert werden, als Sequenzzähler (Signatur-ID) aber ein streng monoton aufsteigender Zähler gefordert ist, wird der konkrete Sequenzzähler nach folgendem Algorithmus ermittelt:

1. Auslesen der 2 Byte langen Bedienungszähler BZ_{DS} des Schlüssels $S_{K.CH.DS}$ bzw. BZ_{AUT} des Schlüssels $S_{K.CH.AUT_{C/S}}$.

2. Sei $neg(BZ)$ die bitweise logische Negation von BZ . Dann sind die Sequenzzähler:

$$\underline{SZ_{DS} = neg(BZ_{DS})}$$

$$\underline{SZ_{AUT} = neg(BZ_{AUT})}$$

Da ein einzelner Bedienungszähler einen Wertebereich von 0 bis 65535 (2 Byte) hat, hat ein Sequenzzähler SZ auch einen Wertebereich von 0 bis 65535 und benötigt zur Darstellung mindestens 2 Byte. Ein Wrap-around bei Erreichen des Maximalwerts findet nicht statt, da das Erreichen eines Bedienungszählers 0 den Schlüssel der Chipkarte für die weitere Verwendung sperrt.

Beispiel:

$$\underline{BZ_{DS} = '00\ 0A' \text{ (dezimal 10)} \Rightarrow SZ_{DS} = neg(BZ_{DS}) = 'FF\ F5' \text{ (dezimal 65525)}}$$

$$\underline{BZ_{AUT} = 'FA\ 1D' \text{ (dezimal 64029)} \Rightarrow SZ_{AUT} = neg(BZ_{AUT}) = '05\ E2' \text{ (dezimal 1506)}}$$

Dieser Algorithmus ist in der jeweiligen Anwendungs-Software zu realisieren.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 72	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.2.2 Beschreibung der Terminalabläufe

Nachfolgend werden die Anwendungsabläufe aus Endgerätesicht an einem privaten Signaturterminal [KT-KONZEPT] spezifiziert. Hierbei werden ausschließlich die chipkartenbezogenen Aspekte berücksichtigt. Anwendungsbezogene Details sind nicht Bestandteil dieser Spezifikation.

Um die Abläufe möglichst einfach beschreiben zu können, werden in der nachfolgenden Beschreibung Befehle der ZKA-SIG-API [KT-SIG] verwendet. Hiermit ist jedoch die Verwendung der ZKA-SIG-API für technische Implementierungen nicht zwingend vorgeschrieben. Wird die ZKA-SIG-API nicht verwendet, so sind die in [KT-SIG] angegebenen Abläufe zum Aufruf der KT-Kommandos zu berücksichtigen.

Die Anwendungsabläufe lassen sich auch auf öffentliche Signaturterminals (Geschäftsterminals) erweitern. Zu beachten ist dabei insbesondere, dass in diesem Fall zusätzlich eine

- Komponenten-Authentifikation zwischen Chipkarte und Geschäftsterminal mit Aushandlung eines Session-Key-Paares (SK1, SK2) stattfindet;
- alle Befehle an die Chipkarte im Secure Messaging mit einem SK2-MAC durchgeführt werden müssen.

Falls bei der Ausführung der Kommandos ein Fehler auftritt, bricht das Terminal den Vorgang ab, es sei denn, es ist ein abweichendes Verhalten spezifiziert.



In den hier beschriebenen Abläufen ist das Kundenterminal durch ein *zka_sig_open* (zu Beginn des Ablaufs „Signatur einleiten“) und ein *zka_sig_close* (Am Ende des Ablaufs „Signatur beenden“) für die gesamte Zeitdauer exklusiv für die FinTS-Kundenanwendung reserviert.

Um zwischenzeitlich anderen Anwendungen die Möglichkeit zu geben, die Signaturdienste der Karte zu nutzen (z. B. für die Zeitdauer der Nachrichtengenerierung), können die im Folgenden beschriebenen Teilabläufe jeweils auch durch ein *zka_sig_open* und ein *zka_sig_close* gekapselt werden. Dadurch wird die exklusive Reservierung des Kundenterminals aufgehoben, die internen Zwischenwerte der ZKA-SIG-API (insbes. der Chipdaten) bleiben jedoch erhalten. Erst durch Aufruf des *zka_sig_fini_signature_application* im Ablauf „Signatur beenden“ werden die internen Zwischenwerte der ZKA-SIG-API gelöscht.



Zur Administration der Signaturkarten (z. B. Freischalten eines Zertifikates, Rücksetzen des Fehlbedienungs Zählers) werden von den Kreditinstituten bzw. den Kartenemittenten Softwarekomponenten zur Verfügung gestellt werden, die in der privaten Kundenumgebung zum Einsatz kommen sollen. In Homebanking-Kundensystemen, die nicht von den Kartenemittenten herausgegeben werden, sollen diese Administrationsfunktionen nicht realisiert werden.



Für die kreditinstitutsseitige Realisierung dieser Softwarekomponenten hat der ZKA Anforderungen und Festlegungen formuliert, die bei Bedarf über die FinTS-Leitstelle erhältlich sind.

III.1.2.2.1 Signatur einleiten

Chipkarte		Endgerät	
		M1	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_open</i>
		←	
		M2	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_init_signature_application</i>
		→	
R2	OK	←	
		M3	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_verify_CSA_password</i>
		→	
R3	OK	←	
		C4	SELECT FILE DF_NOTEPAD
		→	
R4	OK / „File not found“	←	
		C5	ggf. READ RECORD EF_NOTEPAD
		→	
R5	Bankverbindung	A5	Daten prüfen und speichern

♦ Erläuterung

1. Die ZKA-SIG-API-Funktion *zka_sig_open* wird ausgeführt. Diese Funktion stellt eine exklusive Verbindung zum Kundenterminal her.
2. Die ZKA-SIG-API-Funktion *zka_sig_init_signature_application* wird ausgeführt. Diese sorgt insbesondere für ein Reset der Karte und das Auslesen der relevanten Basisinformationen der Karte.
3. Die ZKA-SIG-API-Funktion *zka_sig_verify_CSA_password* wird ausgeführt. Diese Funktion liest das CSA-Passwort ein und führt eine Verifikation gegenüber der Chipkarte durch.
4. Die Applikation „Notepad“ wird geöffnet, indem das ADF der Applikation, DF_NOTEPAD durch das Terminal mittels des Kommandos SELECT FILE ausgewählt wird.

♦ Command APDU

Byte	Wert	Erläuterung
1-2	'00 A4'	CLA, INS
3	'04'	P1, Selektion mit DF-Name
4	'0C'	P2, Keine Antwortdaten
5	'09'	L _C
6-14	'D2 76 00 00 25 4E 50 01 00'	AID der Notepad-Applikation

Wenn die Notepad-Applikation auf der Karte nicht vorhanden ist, wird der folgende Schritt übersprungen. In diesem Fall müssen die Zugangsdaten von einer anderen Stelle gelesen oder vom Benutzer eingegeben werden.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 74	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

4. Das Terminal liest mittels READ RECORD sukzessive die Bankverbindungsdaten in den Records des EF_NOTEPAD (SFI '1A'), bis der oder die "passenden" Einträge gefunden wurden. Das Lesen von Einträgen ist erst nach erfolgreicher CSA-Passwort-Verifikation (Schritt 2) möglich.

♦ Command APDU

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'0X'	P1, Recordnummer X
4	'D4'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die Chipkarte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Länge	Wert	Erläuterung
1-2	2	'XX LL'	Kennung und Länge
3-LL	LL	'XX..XX'	Nutzdaten
(LL+1)- (LL+2)	2	'XX XX'	Positiver Returncode SW1 SW2

Ist die Kennung 'F0', so sind FinTS-Zugangsdaten gemäß *III.1.1 Applikation Notepad* enthalten. Es werden alle weiteren Records gelesen, bis die Chipkarte das Ende der Datei (keine weiteren Records) signalisiert.

Anstatt alle Records auszulesen und auf Übereinstimmung mit der Kennung zu überprüfen, kann alternativ auch das Kommando SEARCH RECORD verwendet werden, um mittels eines übergebenen Suchmusters vorab die "passenden" Recordnummern in einem Schritt zu finden. Anschließend müssen dann nur diese Recordnummern mittels READ RECORD ausgelesen werden.

♦ Command APDU

Byte	Wert	Erläuterung
1-2	'00 A2'	CLA, INS für SEARCH RECORD
3	'01'	P1, Start mit Recordnummer 1
4	'D7'	P2, spezifische Suche im SFI '1A'
5	'04'	L _c
6	'04'	CTRLB
7	'00'	Offset Indicator Byte
8	'02'	Konfigurationsbyte
9	'F0'	Suchmuster
10	'00'	L _e

Wenn das SEARCH RECORD erfolgreich ausgeführt wird, gibt die Chipkarte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Länge	Wert	Erläuterung
1-n	n	'XX XX'	Recordnummer(n)
n+1	1	'XX'	Statusbyte SW1
n+2	1	'XX'	Statusbyte SW2

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 75

Es können nun gezielt nur die in der Antwortnachricht angegebenen Records ausgelesen werden.

III.1.2.2.2 Nachrichten generieren

Dieser Teil des Gesamtablaufs ist nur insofern chipkartenrelevant, als (optional) Bankverbindungsdaten, die für die Auftragsgenerierung benötigt werden, aus der Chipkarte entnommen werden. Dies ist bereits im Schritt „Signatur einleiten“ (*III.1.2.2.1 Signatur einleiten*) geschehen. Für die folgende Ablaufbeschreibung wird angenommen, dass die Anwendung bereits FinTS-Nachrichten generiert hat. Diese Nachrichten müssen jetzt ggf. noch kryptographisch gesichert werden, d. h. es werden Segmente für die elektronische(n) Signatur(en) und für die Verschlüsselung entsprechend den FinTS-Spezifikationen eingefügt.

III.1.2.2.3 Nachrichten signieren

Die folgenden Abläufe können offline, d. h. außerhalb eines FinTS-Dialoges vollzogen werden. Dies gilt nicht für die Erstellung von Botensignaturen. Der Grund besteht darin, dass für die Absicherung aller Kreditinstitutsnachrichten der Schlüssel des Boten erforderlich ist. Daher muss während eines gesamten Dialoges die Chipkarte des Boten im Endgerät stecken.

Die Abläufe für die Botensignatur sind grundsätzlich identisch mit den im Folgenden beschriebenen Abläufen für die Erstellung von Auftragssignaturen. Da aber ggf. für die Botensignatur anwendungsseitig noch weitere Chipkartendaten (Benutzerkennung, Benutzerreferenz, Kommunikationszugang etc.) benötigt werden, wird der komplette Ablauf in *III.1.2.2.5 FinTS-Dialog führen* noch einmal beschrieben.

Chipkarte		Endgerät	
R1	BZ	→	M1 Sequenzzähler (Signatur-ID) ermitteln durch Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_read_key_usage_counter</i> und anschließende Invertierung des Rückgabewerts gemäß Abschnitt <i>III.1.2.1 Verfahren zur Ermittlung der Sequenzzähler</i>)
		←	A2 Signatur aufbauen und in FinTS-Nachricht einfügen
			A3 Daten (Signatur-Segment, FinTS-Nutzdaten) für Signaturerstellung bereitstellen
		→	M4 Signaturerstellung (siehe <i>III.1.2.3.1 Signatur-Berechnung</i>)
		←	A5 ggf. M1 bis M4 für weitere Nachrichten wiederholen
			A6 signierte FinTS-Nachrichten zur Weiterverarbeitung speichern

◆ Erläuterung

- Der Sequenzzähler (Signatur-ID) wird durch Auslesen der Bedienungszähler der Signaturanwendung und anschließende Berechnung ermittelt. Das Auslesen erfolgt durch Aufruf der ZKA-SIG-API-Funktion *zka_sig_read_key_usage_counter* mit der Parameterbelegung

- counter_type = '00' bei Verwendung des S_K.CH.DS, bzw.
- counter_type = '02' bei Verwendung des S_K.CH.AUT_{C/S}

Das Ergebnis BZ wird gemäß *III.1.2.1 Verfahren zur Ermittlung der Sequenzzähler* zu SZ = **neg**(BZ) invertiert und als Sequenzzähler gespeichert.

- Das Signatur-Segment wird aufgebaut und in die FinTS-Nachricht eingefügt.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 76	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

- Die Daten (Signatur-Segment, FinTS-Nutzdaten) für die Signaturerstellung werden bereitgestellt.
- Die Signatur wird berechnet (siehe hierzu *III.1.2.3.1 Signatur-Berechnung*).
- Ggf. können die Schritte 1 bis 4 für weitere Nachrichten wiederholt werden.
- Die signierten FinTS-Nachrichten können zur Weiterverarbeitung gespeichert werden.

Anmerkung: Für Mehrfachsignaturen wird jeweils die Abfolge „Signatur einleiten“ – „Nachrichten signieren“ – „Signatur beenden“ wiederholt. Dies kann auch zu einem späteren Zeitpunkt geschehen. Mehrfachsignaturen müssen jedoch abgeschlossen sein, bevor die Verschlüsselung der Nachricht (*III.1.2.2.4 Nachrichten verschlüsseln*) durchgeführt wird.

III.1.2.2.4 Nachrichten verschlüsseln

Die Chipkarte ist bei der eigentlichen Nachrichtenverschlüsselung nicht involviert. Die Software berechnet einen Einmalschlüssel, verschlüsselt das Dokument und verschlüsselt den Einmalschlüssel zur Übertragung mit dem öffentlichen Key-Encryption-Schlüssel $P_{K.RECV_{INST}.KE}$ des empfangenden Kreditinstituts, welches dem entsprechenden Zertifikat des Empfängers entnommen wurde³.

Allerdings wird die Chipkarte zur Berechnung von Zufallszahlen herangezogen, welche den Einmalschlüssel bilden.

Chipkarte		Endgerät	
		A1	Daten (FinTS-Nutzdaten und ggf. Signatur) für die Verschlüsselung bereitstellen
R2	RND	← C2	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_get_challenge</i>
		→ A2	RND als Einmalschlüssel-Hälfte KS_L speichern
R3	RND	← C3	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_get_challenge</i>
		→ A3	RND als Einmalschlüssel-Hälfte KS_R speichern
		A4	KS_L mit KS_R zu KS konkatenieren und speichern
		A5	KS auf Eigenschaft „(halb-)schwacher Schlüssel“ überprüfen und ggfs. Schritte 2-4 wiederholen.
		A6	Herstellung der Parität für KS (Parity Adjustment)
		A7	Daten mit KS (symmetrisch) verschlüsseln
		A8	KS mit $P_{K.RECV_{INST}.KE}$ (asymmetrisch) verschlüsseln
		A9	Verschlüsselungsdaten aufbauen und in FinTS-Nachricht einfügen
		A10	Verschlüsselte Daten als Binärdaten in Verschlüsselungsdaten einfügen
		A11	ggf. A1 bis A10 für weitere Nachrichten wiederholen
		A12	Verschlüsselte und signierte FinTS-Nachrichten zur weiteren Bearbeitung speichern

³ [DIN-SIG4, Kapitel 6.10.1]: „If an enciphered document is sent, the card is not involved: the software computes the content encryption key, enciphers the document and finally enciphers the content encryption key by applying the receiver's public key taken from the receiver's KE certificate.“

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 77

♦ Erläuterung

1. Die Daten (FinTS-Nutzdaten und ggf. Signatur) für die Verschlüsselung werden bereitgestellt.
2. Mit dem Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge* lässt sich das Terminal eine Zufallszahl von der FinTS-Karte geben.

Wenn das Kommando erfolgreich ausgeführt wurde, gibt die FinTS-Karte eine 8 Byte lange Zufallszahl als Antwortdatum aus, die als Einmalschlüssel-Hälfte KS_L gespeichert wird.

3. Mit dem Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge* lässt sich das Terminal eine weitere Zufallszahl von der FinTS-Karte geben, die als Einmalschlüssel-Hälfte KS_R gespeichert wird.
4. KS_L wird mit KS_R zu KS konkateniert und gespeichert.
5. KS wird auf die Eigenschaft „(halb-)schwacher Schlüssel“ überprüft. Liegt ein (halb-)schwacher Schlüssel vor, so wird Schritt 2-4 wiederholt.

Schwache 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

Halbschwache 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

6. Für KS wird ein Parity Adjustment durchgeführt. Das Resultat ist der zu verwendende Einmalschlüssel.
7. Die zu übertragenden Daten werden mit KS symmetrisch verschlüsselt (Triple-DES CBC-Mode, IV=0, ANSI X9.23 Padding).
8. Der Einmalschlüssel KS wird linksbündig mit Nullbits auf 768 Bit (RDH-1) bzw. 1024-2048 Bit (RDH-2 bis RDH-4) aufgefüllt und anschließend mit dem öffentlichen Key-Encryption-Schlüssel $P_{K.RECV_{INST}.KE}$ des empfangenden Instituts, welches dem entsprechenden Zertifikat des Empfängers entnommen wurde, verschlüsselt. Stimmt das Verschlüsselungsergebnis mit dem Ausgangswert überein, werden die Schritte 2 bis 8 wiederholt (Generierung eines neuen Schlüssels); ansonsten wird das Ergebnis mit führenden Nullbits auf 1024 Bit erweitert und es wird mit dem folgenden Schritt 9 fortgefahren.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 78	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

9. Die Verschlüsselungsdaten werden aufgebaut und in die FinTS-Nachricht eingefügt.
10. Die verschlüsselten Daten als Binärdaten in die Verschlüsselungsdaten eingefügt.
11. Ggf. werden die Schritte 1 bis 10 für weitere Nachrichten wiederholt.
12. Die verschlüsselten und signierten FinTS-Nachrichten werden zur weiteren Bearbeitung gespeichert.

III.1.2.2.5 FinTS-Dialog führen

Chipkarte		Endgerät		Kreditinstitut	
		A1	Benutzererkennung aus der bereits gelesenen Bankverbindung extrahieren		
		→	M2	Nachricht signieren (siehe III.1.2.2.3 Nachrichten signieren)	
		←	A3	Kommunikationszugang aus Bankverbindung herstellen	
			C4	Nachricht (beginnend mit Initialisierungsnachricht) senden	→
				←	R4 Antwortnachricht
			A5	falls Antwortnachricht verschlüsselt: Daten (Binärdaten in Verschlüsselungsdaten) und verschlüsselten Einmalschlüssel enc(KS) aus den Verschlüsselungsdaten für die Entschlüsselung bereitstellen	
		→	M6	Ausführung der ZKA-SIG-API-Funktion <i>zka_sig_decrypt</i> zur Einmalschlüssel-Entschlüsselung, Resultat ist der Einmalschlüssel KS	
		←	A7	Daten mit Einmalschlüssel KS entschlüsseln.	
			A8	falls Kreditinstitutsnachricht signiert: Daten (Signatur, Nutzdaten) für Signatur-Prüfung bereitstellen	
		→	M9	Signatur-Prüfung (siehe III.1.2.3.2 Signatur-Prüfung)	
		←	A10	C4 bis M9 für alle weiteren FinTS-Nachrichten wiederholen	

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH		Stand: 09.07.2004	Seite: 79

III.1.2.2.6 Signatur beenden

Chipkarte	

Endgerät	
M1	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_fini_signature_application</i>
M2	Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_close</i>

♦ Erläuterung

1. Die ZKA-SIG-API-Funktion *zka_sig_fini_signature_application* wird ausgeführt. Diese Funktion setzt die ZKA-SIG-API in den Zustand „passiv“ und löscht die darin gespeicherten Werte.
2. Die ZKA-SIG-API-Funktion *zka_sig_close* gibt die Verbindung zum Kundenterminal wieder frei.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 80	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

III.1.2.3 Makros

III.1.2.3.1 Signatur-Berechnung

Signaturen mit der Chipkarte werden im Rahmen der beiden Sicherheitsdienste „Authentication“ und „Non-Repudiation“ erzeugt.

- Sicherheitsdienst Authentication: Signatur mit Schlüssel $S_{K.CH.AUT_{C/S}}$ (Client-Server-Authentifikations-Schlüssel)
- Sicherheitsdienst Non-Repudiation: Signatur mit Schlüssel $S_{K.CH.DS}$ (Digitaler Signatur-Schlüssel)

Die tatsächliche Durchführung der Signatur durch die Chipkarte ist insbesondere an die Erfüllung von Zugriffsbedingungen geknüpft, hier sind dies insbesondere eine vorhergehende Benutzer-Authentifikation in Form der Verifikation

- des CSA-Passworts für die Erlaubnis zur Signatur mit dem Schlüssel $S_{K.CH.AUT_{C/S}}$
- der Signatur-PIN für die Erlaubnis zur Signatur mit dem Schlüssel $S_{K.CH.DS}$

Durch einen in der Chipkarte personalisierten Parameter der Signatur-Anwendung [ZKASIG] wird dabei festgelegt, nach wie vielen elektronischen Signaturen spätestens die Benutzer-Authentifikation zu wiederholen ist. Eine Benutzer-Authentifikation wird bei Bedarf innerhalb der ZKA-SIG-API-Funktionen *zka_sig_digital_signature* bzw. *zka_sig_cs_authentication* durchgeführt.

Chipkarte			Endgerät	
R1	evtl. Hash-Wert	←	M1	Hash-Wert HASH berechnen, optional unter Verwendung der ZKA-SIG-API-Funktion <i>zka_sig_hash</i>
		→		
R2a	Signatur	←	M2a	Sicherheitsdienst Non-Repudiation: Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_digital_signature</i>
		→		oder:
		←	M2b	Sicherheitsdienst Authentication: Aufruf der ZKA-SIG-API-Funktion <i>zka_sig_cs_authentication</i>
R2b	Signatur	→		

♦ Erläuterung

1. Die Berechnung des Hash-Wertes erfolgt in der Regel außerhalb der Chipkarte (Hash-Algorithmus gemäß Vorgabe für den Sicherheitsdienst bzw. vom Kreditinstitut übermittelter BPD). Optional ist es auch möglich, den letzten Schritt oder alle Schritte der Hash-Wert-Berechnung durch die Chipkarte durchführen zu lassen. Diese Berechnung ist dann Bestandteil des Ablaufs der ZKA-SIG-API-Funktion *zka_sig_hash*. Der zu verwendende Hash-Algorithmus wird dabei in Form der zugehörigen OID übergeben:
 - OID = 1.3.14.3.2.26 für SHA-1
 - OID = 1.3.36.3.2.1 für RIPEMD-160
- 2a. Bei Verwendung des Schlüssels $S_{K.CH.DS}$ (Sicherheitsdienst Non-Repudiation) wird die Signatur durch Aufruf der ZKA-SIG-API-Funktion *zka_sig_digital_signature* erzeugt. Die Auswahl des Signaturalgorithmus und

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH	Stand: 09.07.2004	Seite: 81

Paddingverfahrens erfolgt gemäß Vorgabe für den Sicherheitsdienst bzw. vom Kreditinstitut übermittelter BPD. Die Signaturanwendung der Chipkarte bietet die Verfahren „sha-1WithRSAEncryption“ (PKCS#1-Signaturverfahren, Standard-RSA, SHA-1) und „sigS_ISO9796-2rndWithripemd160“ (DIN-Signaturverfahren, Standard-RSA, RIPEMD-160) an.

Falls der Hash-Wert im vorangegangenen Schritt 1 durch die Chipkarte berechnet wurde, ist er noch in der Chipkarte gespeichert und braucht nicht erneut als Parameter des *zka_sig_digital_signature* übergeben zu werden.

- 2b. Bei Verwendung des Schlüssels $S_{K.CH.AUT_{C/S}}$ (Sicherheitsdienst Authentication) wird die Signatur durch Aufruf der ZKA-SIG-API-Funktion *zka_sig_cs_authentication* erzeugt. Die Chipkarte verwendet dabei intern ein Padding-Format gemäß PKCS#1 ([SECCOS, Kapitel 8.3.2.1]⁴), wobei die Digest-Info nicht von der Chipkarte selbst erzeugt wird, sondern als aufbereiteter „Authentication-Input“ (= zu signierendes Datenfeld) übergeben werden muss.

Der Authentication-Input ist wie folgt aufgebaut ([SECCOS, Kapitel 8.1.8.3.1]):

Tag	Länge	Wert	Erläuterung
'30'	'21'		Tag und Länge von SEQUENCE
'30'	'09'		Tag und Länge von SEQUENCE
'06'	'05'	'2B 0E 03 02 1A' bzw. '2B 24 03 02 01'	OID des SHA-1 (1 3 14 3 2 26) bzw. OID des RIPEMD-160 (1 3 36 3 2 1)
'05'	'00'	-	TLV-Kodierung von NULL
'04'	'14'	'XX..XX'	Hash-Wert

Anmerkung: Die direkte Weiterverwendung eines eventuell im Chip berechneten und dort zwischengespeicherten Hash-Wertes ist bei der Signatur im Sicherheitsdienst „Authentication“ nicht möglich. Der Hash-Wert (als Ergebnis von Schritt 1) muss daher explizit als Aufrufparameter in der oben beschriebenen Form in Schritt 2 übergeben werden.

III.1.2.3.2 Signatur-Prüfung

Die ZKA-Chipkarte selbst unterstützt zurzeit keine Signatur-Prüfung⁵. Die Prüfung einer Signatur wird vom Kundenterminal-Makro „Überprüfen der Korrektheit der elektronischen Unterschrift“ durchgeführt.

⁴ Auszug aus [SECCOS, Kapitel 8.3.2.1]: Falls der Authentication Input nicht zu lang ist, wird er zu einer Folge von N-1 Byte wie folgt formatiert:

Bezeichnung	Byte-Länge	Wert
Blocktyp	1	'01'
Paddingfeld (PS)	N-3-L	'FF...FF'
Separator	1	'00'
Datenfeld	L	Authentication Input (AI)

⁵ [ZKASIG, Kapitel 1.1]: „Die ZKA-Chipkarte unterstützt [die] Signaturprüfung zur Zeit aus dem folgenden Grund nicht: Die Prüfung digitaler Signaturen, die mit beliebigen privaten Schlüsseln und/oder Algorithmen berechnet sind, würde voraussetzen, dass die Chipkarte X.509-Zertifikate auswertet. Dies ist gemäß Kapitel 16.1 von [DINSIG] zur Zeit nicht möglich.“

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 82	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für RDH

Die (mathematische) Korrektheit einer elektronischen Unterschrift wird überprüft, in dem sie mit dem entsprechenden öffentlichen Schlüssel entschlüsselt wird und das Ergebnis mit dem Hash-Wert über die signierten Daten verglichen wird. Der für die Überprüfung der elektronischen Signatur eingesetzte öffentliche Schlüssel liegt in dem Kundenterminal authentisch vor, falls die zu ihm gehörende Zertifikatshierarchie vorher ebenfalls in dem Kundenterminal überprüft wurde [KT-KONZEPT].

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 83

III.2 Chipapplikation für DDV

Im Folgenden wird für das in II *VERFAHRENSBESCHREIBUNG* beschriebene DDV-Verfahren eine entsprechende Chipanwendung namens „Banking“ synonym „FinTS-Banking“ spezifiziert. Voraussetzung ist neben den nachfolgend beschriebenen Datenelementen zusätzlich das Vorhandensein des Datenelements EF_ID sowie des Kryptoalgorithmus Triple-DES, wie sie in der „Schnittstellenspezifikation für die ec-Karte mit Chip“ vom ZKA festgelegt wurden. Die Spezifikation bezieht sich allein auf die für FINTS erforderlichen Datenelemente.

Die Anwendung „Banking“ kann auf einer dedizierten Chipkarte („FinTS-Karte“) oder auf beliebigen multifunktionalen Chipkarten implementiert werden, sofern sie das Betriebssystem der ec-Karte mit Chip einsetzen. Für die FinTS-Anwendung ist kein ausführbarer Code über die Spezifikationen in ISO 7816-4 bzw. der ec-Karte mit Chip hinaus erforderlich.

In diesem Kapitel werden die Datenstrukturen und Zugriffsregeln der Chipapplikation „DF_BANKING“ für Chipkarten vom Typ 0 („altes ZKA-Betriebssystem“) und Typ 1 („neues ZKA-Betriebssystem“) spezifiziert. Die Kommandoabläufe im Terminal sind gemeinsam für Chipkarten vom Typ 0 und Typ 1 spezifiziert.

In *III.2.1 Daten der Applikation FinTS-Banking für Typ 0* wird explizit auf die Beschreibung für Typ 0 eingegangen. Im weiteren Verlauf dieses Dokuments ist mit „*FinTS-Chipkarte*“ eine Chipkarte mit neuem ZKA-Betriebssystem gemäß [DATKOM] und [DAT-MF] gemeint, die die FinTS-Applikation enthält. Weitere Applikationen, wie z. B. die elektronische Geldbörse, sind nicht notwendigerweise auf der Chipkarte enthalten. Ebenso kann die Karte kontobezogen oder kontoungebunden sein.

Das ADF der Applikation FinTS-Banking für das neue ZKA-Betriebssystem wird mit DF_BANKING_20 bezeichnet. In der vorliegenden Spezifikation ist es direkt im MF enthalten. Die für die Applikation relevanten DF-spezifischen Schlüssel sind im EF_KEY abgelegt, das direkt im DF_BANKING_20 enthalten ist.

In der vorliegenden Spezifikation werden im Kontext von Typ 1-Karten zwei Security-Environments verwendet:

- 1 Das Security-Environment mit der Nummer 1 (SE #1) als Standard-SE legt die Zugriffsregeln für die Dateien der Applikation FinTS-Banking für den Anwendungsfall, d. h. für den Zugriff im Feld an FinTS-fähigen Terminals fest.
- 2 Das Security-Environment mit der Nummer 2 (SE #2) als Administrations-SE legt die Zugriffsregeln für die Dateien und das Applikationsverzeichnis der Applikation FinTS-Banking für den Fall von Administrationsvorgängen, z. B. Kontrolle, Änderungen oder Erweiterungen, fest.

Die Selektion von SEs erfolgt, wie in [DATKOM] beschrieben, mit dem Kommando `MANAGE SECURITY ENVIRONMENT`. Für den Anwendungsfall, d. h. an FinTS-fähigen Terminals, ist eine Selektion des SE nicht notwendig, da mit der Selektion einer Applikation implizit das SE #1 aktiviert wird.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 84	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.1 Daten der Applikation FinTS-Banking für Typ 0

Die folgende Abbildung gibt einen Überblick über die Datenelemente (EFs) der Applikation "Banking" für die Typ 0-Karte.

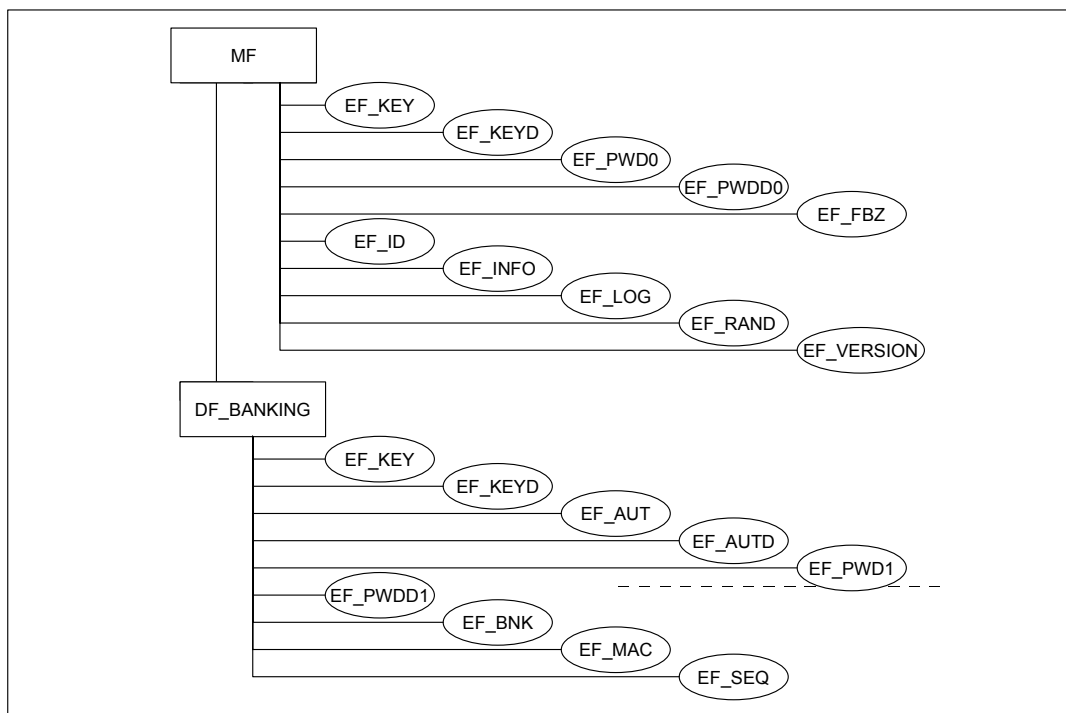


Abbildung 11: Datenelemente der Applikation „Banking“⁶

⁶ Die Elementary Files EF_PWD0, EF_PWDD0, EF_FBZ und EF_INFO sind nicht bei allen Kartentypen vorhanden.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 85

III.2.1.1 DF_Banking

◆ Beschreibung

Der Applikation DF_Banking sind 4 Datenfelder als AEFs zugeordnet, die für das FinTS-Endgerät über Lese- und Schreibzugriff zugänglich sind:

SFI '19'	EF_ID im MF
SFI '1A'	EF_BNK im DF_Banking
SFI '1B'	EF_MAC im DF_Banking
SFI '1C'	EF_SEQ im DF_Banking

Wenn das DF_Banking mittels SELECT FILE selektiert wird und eine entsprechende Option im Parameter P2 des Kommandos gesetzt ist, werden die folgenden FMD mit den Pfaden der AEFs ausgegeben (hierbei wird vorausgesetzt, dass sich das DF_Banking direkt im MF befindet).

◆ Format

Tag	Länge	Wert	Erläuterung
'64'	'1A'		Tag und Länge für FMD
'85'	'03'	'19 00 03'	Pfad für das AEF mit SFI '19' (EF_ID im MF)
'85'	'05'	'1A A6 00 03 01'	Pfad für das AEF mit SFI '1A' (EF_BNK im DF_Banking)
'85'	'05'	'1B A6 00 03 02'	Pfad für das AEF mit SFI '1B' (EF_MAC im DF_Banking)
'85'	'05'	'1C A6 00 03 03'	Pfad für das AEF mit SFI '1C' (EF_SEQ im DF_Banking)

Ebenfalls über den Parameter P2 kann mit dem SELECT FILE der folgende FCI mit den ACs der AEFs im zusammengesetzten Datenobjekt mit Tag 'A5' ausgegeben werden.

Tag	Länge	Wert	Erläuterungen
'6F'	'40'		Tag und Länge für FCI
'81'	'02'	'xxxx'	freier Speicherplatz in der ec-Karte
'82'	'01'	'38'	Datei-Deskriptor für DF
'83'	'02'	'A600'	Datei-ID der Applikation 'Banking'
'84'	'09'	'D27600002548420100'	DF-Name (AID) des DF_Banking
'86'	'02'	'0040'	AC für DF_Banking
'A5'	'24'		Tag und Länge der ACs der AEFs
'86'	'07'	'19 0040 0000 00F0'	SFI und ACs des EF_ID
'86'	'07'	'1A 0040 0000 0031'	SFI und ACs des EF_BNK
'86'	'07'	'1B 0040 3150 0031'	SFI und ACs des EF_MAC
'86'	'07'	'1C 0040 0000 0031'	SFI und ACs des EF_SEQ

◆ Erläuterungen

Tag '83':

Die Datei-ID der Applikation 'Banking' lautet A600.

Tag '84':

Application Identifier (AID) für Homebanking mit Chipkarte

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 86	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Tag '86':

Kommandogruppe ADMIN: AC '0040' (PRO_G mit Schlüsselnummer '00')

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 87

III.2.1.2 EF_KEY

◆ Beschreibung

In diesem EF wird der kartenindividuelle Signierschlüssel abgelegt.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 11'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 11'	Datei-Deskriptor für lineares EF
'83'	'02'	'00 10'	Datei-ID des EF_KEY
'86'	'06'	'0060 00F0 0060'	ACs für das EF_KEY

◆ Erläuterungen

Tag '81':

Das EF_KEY enthält maximal einen Record in der Länge 17 Byte, so dass 17 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 17 Byte (X'11').

Tag '83':

Die Datei-ID muss '00 10' lauten bei einem EF_KEY.

Tag '86':

APPEND RECORD: AC '00 60' (ENC_G mit Schlüsselnummer '00')

READ RECORD: AC '00 F0' (NEV)

UPDATE RECORD: AC '00 60' (ENC_G mit Schlüsselnummer '00')

◆ Daten

Das EF_KEY des DF_Banking enthält einen Record. Der Record enthält die logische Schlüsselnummer mit dem Signierschlüssel.

Logische Schlüsselnr.	Schlüssel
'00'	16 Byte lange K_{DSG}

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 88	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.1.3 EF_KEYD

◆ Beschreibung

Dieses EF dient der Beschreibung der Eigenschaften des Signierschlüssels.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 05'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 05'	Datei-Deskriptor für lineares EF
'83'	'02'	'00 13'	Datei-ID des EF_KEYD
'86'	'06'	'0040 0000 0040'	ACs für das EF_KEYD

◆ Erläuterungen

Tag '81':

Das EF_KEYD enthält maximal einen Record in der Länge 5 Byte, so dass 5 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 5 Byte.

Tag '83':

Die Datei-ID muss '00 13' lauten bei einem EF_KEYD.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '00 00' (ALW)

UPDATE RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

◆ Daten

Für jeden im EF_KEY gespeicherten Schlüssel muss ein Record, der die jeweilige logische Schlüsselnummer und weitere Parameter enthält, im EF_KEYD enthalten sein.

Log. Schlüsselnr.	Schlüssel- länge	Algorithmus- ID	FBZ	Schlüsselversion
'00'	'10'	'07'	'FF'	'00'

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 89

III.2.1.4 EF_AUT

◆ Beschreibung

In diesem EF wird der kartenindividuelle Chiffrierschlüssel abgelegt.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 11'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 11'	Datei-Deskriptor für lineares EF
'83'	'02'	'00 11'	Datei-ID des EF_AUT
'86'	'06'	'0060 00F0 0060'	ACs für das EF_AUT

◆ Erläuterungen

Tag '81':

Das EF_AUT enthält maximal einen Record in der Länge 17 Byte, so dass 17 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 17 Byte (X'11').

Tag '83':

Die Datei-ID muss '00 11' lauten bei einem EF_AUT.

Tag '86':

APPEND RECORD: AC '00 60' (ENC_G mit Schlüsselnummer '00')

READ RECORD: AC '00 F0' (NEV)

UPDATE RECORD: AC '00 60' (ENC_G mit Schlüsselnummer '00')

◆ Daten

Das EF_AUT der DF_Banking enthält einen Record. Der Record enthält die logische Schlüsselnummer mit dem Schlüssel.

Logische Schlüsselnr.	Schlüssel
'00'	16 Byte lange K_{ENC}

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 90	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.1.5 EF_AUTD

◆ Beschreibung

Dieses EF dient der Beschreibung der Eigenschaften des Chiffrierschlüssels.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 04'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 04'	Datei-Deskriptor für lineares EF
'83'	'02'	'00 14'	Datei-ID des EF_AUTD
'86'	'06'	'0040 0000 0040'	ACs für das EF_AUTD

◆ Erläuterungen

Tag '81':

Das EF_AUTD enthält maximal einen Record in der Länge 4 Byte, so dass 4 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 4 Byte.

Tag '83':

Die Datei-ID muss '00 14' lauten bei einem EF_AUTD.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '00 00' (ALW)

UPDATE RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

◆ Daten

Für jeden im EF_AUT gespeicherten Schlüssel muss ein Record, der die jeweilige logische Schlüsselnummer und weitere Parameter enthält, im EF_AUTD enthalten sein.

Log. Schlüsselnr.	Schlüssellänge	Algorithmus-ID	Schlüsselversion
'00'	'10'	'07'	'00'

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 92	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Jedes Feld repräsentiert ein Halbbyte:

- C: Kontrollfeld, binär kodiert hat immer den Wert '2'
- L: PIN-Länge, binär kodiert mögliche Werte von '5' bis 'C'
- P: PIN-Ziffer, BCD-kodiert
- F: Filler, Binär kodiert hat immer der Wert 'F'
- P/F: PIN/Filler abhängig von der PIN-Länge

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 93

III.2.1.7 EF_PWDD1

◆ Beschreibung

Dieses EF dient der Beschreibung der Eigenschaften der Banking-PIN.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 05'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 05'	Datei-Deskriptor für lineares EF
'83'	'02'	'00 25'	Datei-ID des EF_PWDD1
'86'	'06'	'0040 0000 0040'	ACs für das EF_PWDD1

◆ Erläuterungen

Tag '81':

Das EF_PWDD1 enthält maximal einen Record in der Länge 5 Byte, so dass 5 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 5 Byte.

Tag '83':

Die Datei-ID muss '00 25' lauten bei einem EF_PWDD1.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '00 00' (ALW)

UPDATE RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

◆ Daten

Im 5 Byte langen Record des EF_PWDD1 werden Zusatzinformationen zur Banking-PIN festgehalten:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
'01'	'21'	'05'	'03'	'03'

Byte 1: Art der Karteninhaber-Authentifikation. Zur Zeit ist nur die Paßwort-Verifikation möglich, die durch den Wert '01' kodiert ist.

Byte 2: Die PIN ist im Format 'Format 2 PIN Block' in BCD gespeichert.

Byte 3: Die PIN muss mindestens 5 Ziffern lang sein.

Byte 4: Initialisierungswert des PIN Fehlbedienungs Zählers in Byte 5.

Byte 5: PIN Fehlbedienungs Zähler.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 94	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.1.8 EF_BNK

◆ Beschreibung

Bei dem EF_BNK handelt es sich um ein lineares EF mit 5 Records in denen Bankverbindungen abgelegt sind. Die Bankverbindung kann über TCP/IP, HTTPS, SMTP oder weitere zukünftige Onlinedienste hergestellt werden.

Der Record setzt sich aus einer Bankkurzbezeichnung, der Bankleitzahl, dem Kommunikationsdienst, der Adresse und dem Adresszusatz für den Kommunikationszugang, dem Länderkennzeichen und der Benutzerkennung zusammen.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'01 B8'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 58'	Datei-Deskriptor für lineares EF
'83'	'02'	'03 01'	Datei-ID des EF_BNK
'86'	'06'	'0040 0000 0031'	ACs für das EF_BNK

◆ Erläuterungen

Tag '81':

Das EF_BNK enthält maximal 5 Records in der Länge 88 Byte, so dass 440 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 88 Byte (X'58').

Tag '83':

Als Datei-ID wurde '03 01' für das EF_BNK festgelegt.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '00 00' (ALW)

UPDATE RECORD: AC '00 31' (PWD_D)

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI		Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV		Stand: 09.07.2004	Seite: 95

◆ Daten

Das EF_BNK enthält 5 Records mit folgendem Satzaufbau:

Byte	Länge	Wert	Erläuterung
1-20	20	'aa .. aa'	Kurzbezeichner des Kreditinstituts
21-24	4	'nn nn nn nn'	Kreditinstitutscode des kontoführenden Kreditinstituts
25-25	1	'n'	Kommunikationsdienst (gültige Codierungen siehe III.1.1.2)
26-53	28	'aa .. aa'	Kommunikationsadresse
54-55	2	'aa aa'	Kommunikationsadressenzusatz
56-58	3	'aa aa aa'	Länderkennzeichen des kontoführenden Kreditinstituts
59-88	30	'aa .. aa'	Benutzerkennung

Alphanumerische Feldinhalte ('a') werden grundsätzlich ASCII-kodiert, linksbündig eingestellt und mit Leerzeichen (X'20') auf die vorgegebene Länge aufgefüllt.

Numerische Feldinhalte ('n') werden grundsätzlich BCD-kodiert.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 96	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.1.9 EF_MAC

◆ Beschreibung

Das EF_MAC wird für die MAC-Bildung über den Hash-Wert einer Nachricht benötigt.

Es besteht aus einem 12 Byte langem Record deren ACs so gesetzt werden müssen, dass beim Lesen des Records der MAC produziert wird.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 0C'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 0C'	Datei-Deskriptor für lineares EF
'83'	'02'	'03 02'	Datei-ID des EF_MAC
'86'	'06'	'0040 3150 0031'	ACs für das EF_MAC

◆ Erläuterungen

Tag '81':

Das EF_MAC enthält maximal einen Record in der Länge 12 Byte, so dass 12 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 12 Byte (X'0C').

Tag '83':

Als Datei-ID wurde '03 02' für das EF_MAC festgelegt.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '31 50' (PWD_D und PRO_D mit Schlüsselnr. '00')

UPDATE RECORD: AC '00 31' (PWD_D)

◆ Daten

Das EF_MAC besitzt einen 12 Byte langen Record der durch UPDATE RECORD modifiziert wird.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 97

III.2.1.10 EF_SEQ

◆ Beschreibung

Bei dem EF_SEQ handelt es sich um ein lineares Datenfile, dessen Record ein 2 Byte langes binär definiertes Element enthält.

Dieser binäre aufsteigende Zähler fließt als Signatur-ID zur Absicherung der Daten gegen Doppeleinreichung ein (siehe II.5.1 *Signatur-Segment*).

Der Startwert des Zählers ist 1. Ein Rücksetzen bei Überlauf findet nicht statt.

◆ Format

File Control Parameter (FCP):

Tag	Länge	Wert	Erläuterung
'62'	'15'		Tag und Länge für FCP
'81'	'02'	'00 02'	allokierter Speicherplatz in Byte
'82'	'03'	'02 41 02'	Datei-Deskriptor für lineares EF
'83'	'02'	'03 03'	Datei-ID des EF_SEQ
'86'	'06'	'0040 0000 0031'	ACs für das EF_SEQ

◆ Erläuterungen

Tag '81':

Das EF_SEQ enthält maximal einen Record in der Länge 2 Byte, so dass 2 Byte benötigt werden.

Tag '82':

Die Recordlänge beträgt 2 Byte.

Tag '83':

Als Datei-ID wurde '03 03' für das EF_SEQ festgelegt.

Tag '86':

APPEND RECORD: AC '00 40' (PRO_G mit Schlüsselnummer '00')

READ RECORD: AC '00 00' (ALW)

UPDATE RECORD: AC '00 31' (PWD_D)

◆ Daten

Das EF_SEQ besitzt einen 2 Byte langen Record der durch UPDATE RECORD modifiziert wird.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 98	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.2 Daten der Applikation FinTS-Banking für Typ 1

Die folgende Abbildung gibt eine Übersicht über die Dateien einer FinTS-Karte mit der Applikation FinTS-Banking für Typ 1.

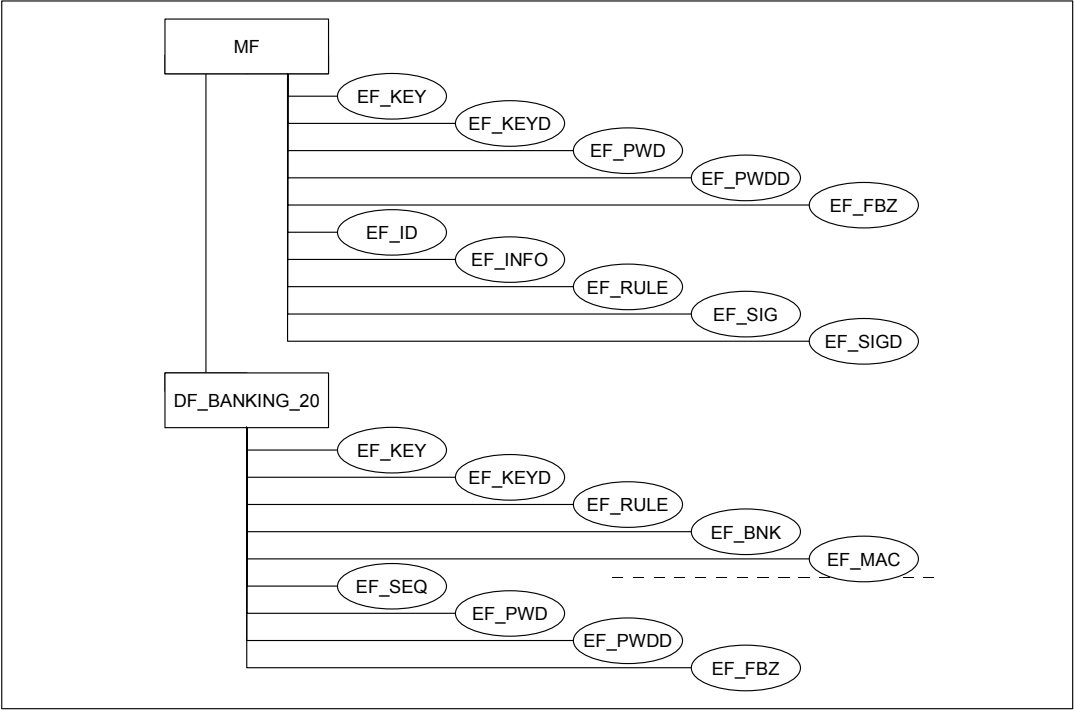


Abbildung 12: Datenelemente der Applikation "FinTS", kontobezogene Karte

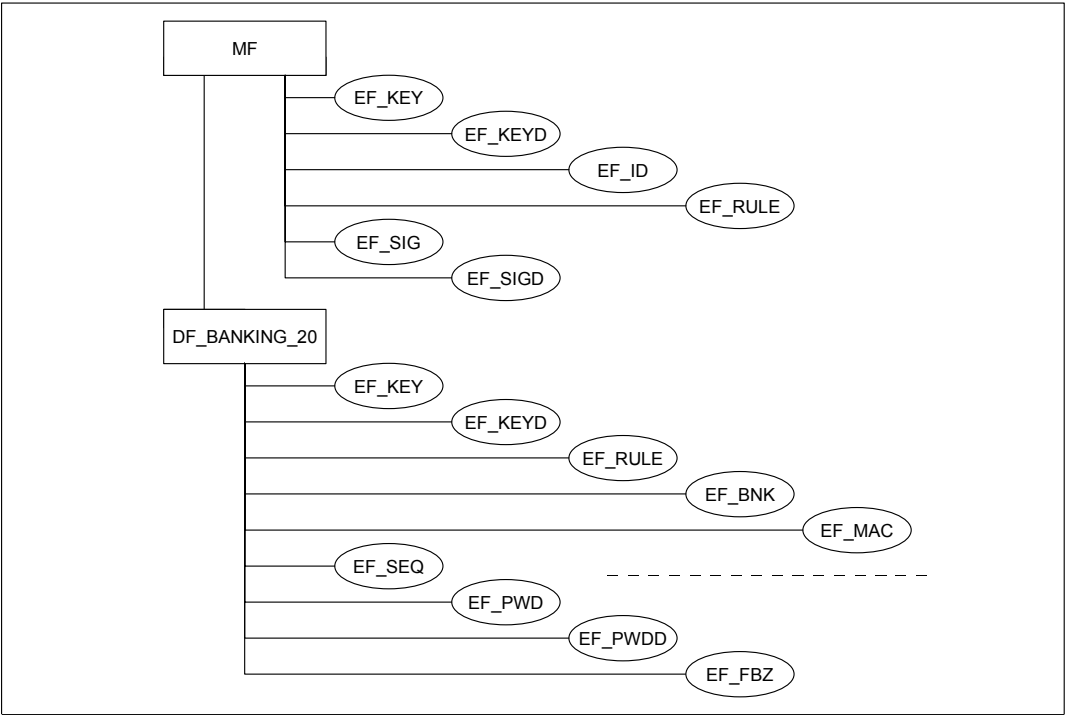


Abbildung 13: Datenelemente der Applikation "FinTS", kontoungebundene Karte

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 99

III.2.2.1 ADF der Applikation FinTS-Banking

Für das ADF der Applikation FinTS-Banking (DF_BANKING_20) sind beim Anlegen die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1A'		Tag und Länge für FCP
'82'	'01'	'38'	Datei-Deskriptor für DF
'83'	'02'	'A6 00'	Datei-ID des DF_BANKING_20
'84'	'09'	'D2 76 00 00 25 48 42 02 00'	DF-Name (AID) des DF_BANKING_20
'A1'	'06'	'8B 04 00 30 02 01'	Zugriffsregel-Referenzen

Der DF-Name (die AID) des DF_BANKING_20 bestehend aus der nationalen RID des ZKA ('D2 76 00 00 25'), der ASCII-kodierten Kennung "HB" ('48 42') sowie der Version der Applikation 2.0 ('02 00').

Die Zugriffsregeln für das DF_BANKING_20 stehen in der zugeordneten Regeldatei EF_RULE. Durch die Zugriffsregeln werden für die DF-spezifischen Kommandos die folgenden Festlegungen getroffen:

Wenn das DF_BANKING_20 selektiert ist, darf ein CREATE FILE (EF), DELETE FILE (self), INCLUDE oder EXCLUDE nur ausgeführt werden, wenn die Kommandonachricht mit Secure Messaging ausgeführt wird und mit einem korrekten MAC versehen ist, der unter Verwendung des Schlüssels K_{HBCI_Admin} aus dem EF_KEY des DF_BANKING_20 gebildet ist. Der Returncode wird für jedes dieser Kommandos durch die Karte mit einem MAC mit dem Schlüssel K_{HBCI_Admin} versehen. Die Kommandos CREATE FILE (DF) und DELETE FILE (child DF) dürfen nie ausgeführt werden. Alle zulässigen Administrationskommandos dürfen nur im SE #2 ausgeführt werden (Zugriffsregeln im Record 1 des EF_RULE).

Der Applikation FinTS-Banking sind 10 Dateien als AEF zuzuordnen:

SFI '01': EF_RULE im DF_BANKING_20
SFI '02': EF_KEY im DF_BANKING_20,
SFI '03': EF_PWD im DF_BANKING_20,
SFI '04': EF_PWDD im DF_BANKING_20,
SFI '05': EF_FBZ im DF_BANKING_20,
SFI '19': EF_ID im MF,
SFI '1A': EF_BNK im DF_BANKING_20,
SFI '1B': EF_MAC im DF_BANKING_20,
SFI '1C': EF_SEQ im DF_BANKING_20,
SFI '1E': EF_KEYD im DF_BANKING_20.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 100	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Wenn das DF_BANKING_20 mittels SELECT FILE selektiert wird und die entsprechende Option im Parameterbyte P2 des Kommandos gesetzt ist, wird die folgende FCI ausgegeben:

Tag	Länge	Wert	Erläuterung
'6F'	'0D'		Tag und Länge für FCI
'84'	'09'	'D2 76 00 00 25 48 42 02 00'	DF-Name (AID) des DF_BANKING_20
'A5'	'00'		keine proprietären Informationen

Wird das DF_BANKING_20 mittels SELECT FILE selektiert und die entsprechende Option im Parameterbyte P2 des Kommandos gesetzt, werden die folgenden FMD mit den Pfaden der AEFs ausgegeben (vorausgesetzt, das DF_BANKING_20 befindet sich direkt im MF):

Tag	Länge	Wert	Erläuterung
'64'	'44'		Tag und Länge für FMD
'85'	'03'	'C8 00 03'	Pfad für AEF mit SFI '19' (EF_ID im MF)
'85'	'05'	'08 A6 00 00 30'	Pfad für AEF mit SFI '01' (EF_RULE im DF_BANKING_20)
'85'	'05'	'10 A6 00 00 10'	Pfad für AEF mit SFI '02' (EF_KEY im DF_BANKING_20)
'85'	'05'	'18 A6 00 00 12'	Pfad für AEF mit SFI '03' (EF_PWD im DF_BANKING_20)
'85'	'05'	'20 A6 00 00 15'	Pfad für AEF mit SFI '04' (EF_PWDD im DF_BANKING_20)
'85'	'05'	'28 A6 00 00 16'	Pfad für AEF mit SFI '05' (EF_FBZ im DF_BANKING_20)
'85'	'05'	'D0 A6 00 03 01'	Pfad für AEF mit SFI '1A' (EF_BNK im DF_BANKING_20)
'85'	'05'	'D8 A6 00 03 02'	Pfad für AEF mit SFI '1B' (EF_MAC im DF_BANKING_20)
'85'	'05'	'E0 A6 00 03 03'	Pfad für AEF mit SFI '1C' (EF_SEQ im DF_BANKING_20)
'85'	'05'	'F0 A6 00 00 13'	Pfad für AEF mit SFI '1E' (EF_KEYD im DF_BANKING_20)

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 101

III.2.2.2 EF_RULE

◆ Beschreibung

Die Datei EF_RULE enthält die Zugriffsregeln für die Applikation DF_BANKING_20. In den FCP von Dateien und Verzeichnissen wird auf diese Zugriffsregeln referenziert.

◆ Format

Für das EF_RULE des DF_BANKING_20 sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		Tag und Länge für FCP
'82'	'05'	'14 41 00 24 08'	Datei-Deskriptor für lineares EF mit variabler Recordlänge (max. 36 Byte), 8 Records
'83'	'02'	'00 30'	Datei-ID des EF_RULE
'85'	'02'	'00 7D'	für Nutzdaten allozierter Speicherplatz in Byte
'88'	'01'	'08'	SFI '01' für das EF_RULE
'A1'	'08'	'8B 06 00 30 01 02 02 03'	Zugriffsregel-Referenzen

Im SE #1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando- und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 darf APPEND RECORD nur ausgeführt werden, wenn es mit Secure Messaging ausgeführt wird. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} . UPDATE RECORD darf nie ausgeführt werden (Zugriffsregel im Record 3 des EF_RULE).

◆ Daten

Das EF_RULE im DF_BANKING_20 enthält 8 Records mit den Zugriffsregeln für das Verzeichnis und die Datenfelder des Verzeichnisses.

Die folgende Tabelle zeigt die Belegung dieser Records für eine FinTS-Chipkarte:

Rec.Nr.	Record-Inhalt	Byte
1	'80 01 DA B4 05 83 03 80 01 FF'	10
2	'80 01 81 90 00'	5
3	'80 01 84 B4 05 83 03 80 01 FF'	10
4	'80 01 86 AF 11 B4 05 83 03 80 01 FF B8 08 95 01 10 83 03 80 01 FF'	22
5	'80 01 86 B4 05 83 03 80 01 FF'	10
6	'80 01 82 A4 07 95 01 08 93 02 80 01 80 01 81 90 00'	17
7	'80 01 82 A4 07 95 01 08 93 02 80 01 80 01 81 AF 13 B4 08 95 01 20 83 03 80 02 FF A4 07 95 01 08 93 02 80 01'	36
8	'80 01 83 90 00 80 01 84 B4 05 83 03 80 01 FF'	15

Die Records 1 bis 5 enthalten jeweils eine, die Records 6 bis 8 jeweils zwei Zugriffsregeln.

Im Folgenden werden die einzelnen Records des EF_RULE näher erläutert.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 102	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Record 1 wird referenziert als Zugriffsregel von DF_BANKING_20 in SE #2.

CREATE FILE (EF), DELETE FILE (self), INCLUDE, EXCLUDE: MAC-SM-AC für Kommando- und Antwortnachricht mit K_{HBCI_Admin} :

Tag	Länge	Wert	Erläuterung
'80'	'01'	'DA'	Zugriffsart für CREATE FILE (EF), DELETE FILE (self), INCLUDE, EXCLUDE
'B4'	'05'		CCT - Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K_{HBCI_Admin}

Record 2 wird referenziert als Zugriffsregel von EF_RULE, EF_KEYD, EF_PWDD und EF_FBZ in SE #1.

READ / SEARCH RECORD: ALW

Tag	Länge	Wert	Erläuterung
'80'	'01'	'81'	Zugriffsart für READ / SEARCH RECORD
'90'	'00'		Zugriffsbedingung ALW

Record 3 wird referenziert als Zugriffsregel von EF_RULE, EF_BNK und EF_MAC in SE #2.

APPEND RECORD: MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel K_{HBCI_Admin} .

Tag	Länge	Wert	Erläuterung
'80'	'01'	'84'	Zugriffsart für APPEND RECORD
'B4'	'05'		CCT - Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K_{HBCI_Admin}

Record 4 wird referenziert als Zugriffsregel von EF_KEY und EF_PWD in SE #2.

APPEND RECORD, UPDATE RECORD: MAC-ENC-SM-AC für Kommandonachricht und MAC-SM-AC für Antwortnachricht mit K_{HBCI_Admin} .

Tag	Länge	Wert	Erläuterung
'80'	'01'	'86'	Zugriffsart für APPEND RECORD, UPDATE RECORD
'AF'	'11'		AND- Template, Tag und Länge
'B4'	'05'		CCT - Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K_{HBCI_Admin}
'B8'	'08'		CT - Tag und Länge
'95'	'01'	'10'	Usage Qualifier: Nur für Kommandonachricht
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K_{HBCI_Admin}

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 103

Record 5 wird referenziert als Zugriffsregel von EF_KEYD, EF_SEQ, EF_PWDD und EF_FBZ in SE #2.

APPEND RECORD, UPDATE RECORD: MAC-SM-AC für Kommando- und Antwortnachricht mit dem Schlüssel K_{HBCI_Admin} .

Tag	Länge	Wert	Erläuterung
'80'	'01'	'86'	Zugriffsart für APPEND RECORD, UPDATE RECORD
'B4'	'05'		CCT - Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K_{HBCI_Admin}

Record 6 wird referenziert als Zugriffsregel von EF_BNK und EF_SEQ in SE #1.

UPDATE RECORD: Karteninhaber-Authentifikation (PWD) mit lokalem Passwort 1.

READ / SEARCH RECORD: ALW

Tag	Länge	Wert	Erläuterung
'80'	'01'	'82'	Zugriffsart für UPDATE RECORD
'A4'	'07'		AT - Tag und Länge
'95'	'01'	'08'	Usage Qualifier für Karteninhaber-Authentifikation
'83'	'02'	'80 01'	Passwort-Referenz, lokales Passwort mit der Nummer 1
'80'	'01'	'81'	Zugriffsart für READ / SEARCH RECORD
'90'	'00'		ALW

Record 7 wird referenziert als Zugriffsregel von EF_MAC in SE #1.

UPDATE RECORD: Karteninhaber-Authentifikation (PWD) mit lokalem Passwort 1.

READ / SEARCH RECORD: Karteninhaber-Authentifikation (PWD) mit lokalem Passwort 1 und MAC-SM-AC für die Antwortnachricht mit dem Schlüssel K_{DAK} .

Tag	Länge	Wert	Erläuterung
'80'	'01'	'82'	Zugriffsart für UPDATE RECORD
'A4'	'07'		AT - Tag und Länge
'95'	'01'	'08'	Usage Qualifier für Karteninhaber-Authentifikation
'83'	'02'	'80 01'	Passwort-Referenz, lokales Passwort mit der Nummer 1
'80'	'01'	'81'	Zugriffsart für READ / SEARCH RECORD
'AF'	'13'		AND – Template, Tag und Länge
'B4'	'08'		CCT – Tag und Länge
'95'	'01'	'20'	Usage Qualifier: Nur Antwortnachricht
'83'	'03'	'80 02 FF'	Schlüsselreferenz für K_{DAK}
'A4'	'07'		AT - Tag und Länge
'95'	'01'	'08'	Usage Qualifier für Karteninhaber-Authentifikation
'83'	'02'	'80 01'	Passwort-Referenz, lokales Passwort mit der Nummer 1

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 104	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Record 8 wird referenziert als Zugriffsregel im EF_PWDD.

VERIFY, CHANGE REFERENCE DATA: ALW

RESET RETRY COUNTER: MAC-SM-AC für Kommando- und Antwortnachricht mit
K_{HBCI_Admin}

Tag	Länge	Wert	Beschreibung
'80'	'01'	'83'	Zugriffsart für VERIFY, CHANGE REFERENCE DATA
'90'	'00'		ALW
'80'	'01'	'84'	Zugriffsart für Kommando: RESET RETRY COUNTER
'B4'	'05'		CCT – Tag und Länge
'83'	'03'	'80 01 FF'	Schlüsselreferenz für K _{HBCI_Admin}

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 105

III.2.2.3 EF_KEY

◆ Beschreibung

Die applikationsspezifischen Schlüssel der Applikation FinTS-Banking sind im EF_KEY des Applikationsverzeichnisses DF_BANKING_20 gespeichert. Dies sind

- ein 16 Byte langer kartenindividueller Schlüssel $K_{\text{HBCI_Admin}}$ mit der Schlüsselnummer '01' zur Administration der Applikation DF_BANKING_20,
- ein 16 Byte langer kartenindividueller Schlüssel K_{DAK} mit der Schlüsselnummer '02' als kartenindividueller Daten-Authentifikationsschlüssel (DAK = Data Authentication Key)⁷, sowie
- ein 16 Byte langer kartenindividueller Schlüssel K_{ENC} mit der Schlüsselnummer '03' als kartenindividueller Chiffrierschlüssel.

Die Schlüssel $K_{\text{HBCI_Admin}}$, K_{DAK} und K_{ENC} sind nur der FinTS-Chipkarte und dem für sie zuständigen Hintergrundsystem bekannt. Sie werden jeweils aus einem KGK (Key Generating Key) unter Verwendung der Kartenidentifikationsdaten im EF_ID des MF abgeleitet (vgl. Kapitel 8.4.1 von [DATKOM]). Das zuständige Hintergrundsystem kennt die jeweiligen KGK und leitet die kartenindividuellen Schlüssel bei Bedarf ab.

Es können pro logischer Schlüsselnummer verschiedene KGK verwendet werden. Ein KGK wird wie alle daraus abgeleiteten Schlüssel anhand der Schlüsselversion identifiziert. Die Schlüsselversion zur jeweiligen logischen Schlüsselnummer im zugehörigen EF_KEYD zeigt an, aus welchem KGK der jeweilige kartenindividuelle Schlüssel abgeleitet ist.

◆ Format

Für das EF_KEY des DF_BANKING_20 sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'16'		Tag und Länge für FCP
'82'	'05'	'12 41 00 12 03'	Datei-Deskriptor für lineares EF mit fester Recordlänge (18 Byte), 3 Records
'83'	'02'	'00 10'	Datei-ID des EF_KEY
'88'	'01'	'10'	SFI '02' für das EF_KEY
'A1'	'06'	'8B 04 00 30 02 04'	Zugriffsregel-Referenzen

Auf das EF_KEY darf nur im SE #2 zugegriffen werden.

Die Kommandos APPEND RECORD und UPDATE RECORD dürfen nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden, der Record-Inhalt verschlüsselt (ENC) ist und die Kommandonachricht mit einem MAC abgesichert ist. Verschlüsselung und MAC-Bildung erfolgen mit dem $K_{\text{HBCI_Admin}}$. Der Returncode eines APPEND RECORD oder UPDATE RECORD wird mit dem $K_{\text{HBCI_Admin}}$ MAC-gesichert. Das Kommando READ RECORD darf nie ausgeführt werden. (Zugriffsregel im Record 4 des EF_RULE)

⁷ Um den Begriff „Signierschlüssel“ für Anwendungen nach SigG bzw. EU-Richtlinie freizuhalten, wurde hier der Begriff „Daten-Authentifikationsschlüssel“ gewählt. Im weiteren Text wird jedoch zur besseren Lesbarkeit weiterhin davon gesprochen, dass eine Nachricht mit diesem Schlüssel signiert wird.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 106	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

◆ Daten

Das EF_KEY im DF_BANKING_20 enthält 3 Records mit den DF-spezifischen Schlüsseln des DF_BANKING_20.

Logische Schlüsselnummer	Schlüssel-Version	Schlüssel
'01'	'XX'	16 Byte langer K _{HBCI_Admin}
'02'	'XX'	16 Byte langer K _{DAK}
'03'	'XX'	16 Byte langer K _{ENC}

Es werden die Schlüsselversionen 1 bis 127 verwendet.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 107

III.2.2.4 EF_KEYD

◆ Beschreibung

Das EF_KEYD im DF_BANKING_20 enthält die Zusatzinformationen zu den DF-spezifischen Schlüsseln des DF_BANKING_20.

◆ Format

Für das EF_KEYD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		Tag und Länge für FCP
'82'	'05'	'14 41 00 1C 03'	Datei-Deskriptor für lineares EF mit variabler Recordlänge (max. 28 Byte) und 3 Records
'83'	'02'	'00 13'	Datei-ID des EF_KEYD
'85'	'02'	'00 48'	für Nutzdaten allozierter Speicherplatz in Byte
'88'	'01'	'F0'	SFI '1E' für das EF_KEYD
'A1'	'08'	'8B 06 00 30 01 02 02 05'	Zugriffsregel-Referenzen

Im SE #1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando- und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 dürfen die Kommandos APPEND RECORD und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} (Zugriffsregel im Record 5 des EF_RULE).

◆ Daten

Das EF_KEYD enthält 3 Records, die die Zusatzinformation zu den DF-spezifischen Schlüsseln des DF_BANKING_20 enthalten.

Das Datenobjekt mit Tag '93' enthält im Wertfeld als zweites Byte die Version des entsprechenden Schlüssels.

Im folgenden wird der Aufbau der Schlüsselzusatzinformation dargestellt:

Eintrag 1 (K_{HBCI_Admin}):

Tag	Länge	Wert	Erläuterung
'93'	'02'	'01 XX'	Schlüsselnummer und Schlüssel-Version
'C0'	'02'	'81 10'	Symmetrischer Schlüssel der Länge 16 Byte
'90'	'01'	'FF'	Fehlbedienungszähler
'7B'	'0F'		SE-Datenobjekt
'80'	'01'	'02'	Festlegung für SE #2
'B4'	'04'		CCT - Tag und Länge (Usage Qualifier '30' ist Defaultwert)
'89'	'02'	'12 22'	Algorithmus-ID: Schlüssel darf zur Bildung eines Retail-MAC im CFB-Mode verwendet werden
'B8'	'04'		CT - Tag und Länge (Usage Qualifier '10' ist Defaultwert)
'89'	'02'	'11 23'	Algorithmus-ID: Schlüssel darf zur Verschlüsselung als Triple-DES Schlüssel im CBC-Mode mit ICV \neq 0 und

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 108	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

			ICV-Variante verwendet werden
--	--	--	-------------------------------

Eintrag 2 (K_{DAK}):

Tag	Länge	Wert	Erläuterung
'93'	'02'	'02 XX'	Schlüsselnummer und Schlüssel-Version
'C0'	'02'	'81 10'	Symmetrischer Schlüssel der Länge 16 Byte
'7B'	'0C'		SE-Datenobjekt
'80'	'01'	'01'	Festlegung für SE #1
'B4'	'07'		CCT - Tag und Länge
'95'	'01'	'20'	Usage Qualifier: Nur SM-Antwortnachricht
'89'	'02'	'12 22'	Algorithmus-ID: Schlüssel darf zur Bildung eines Re-tail-MAC im CFB-Mode verwendet werden

Eintrag 3 (K_{ENC}):

Tag	Länge	Wert	Erläuterung
'93'	'02'	'03 XX'	Schlüsselnummer und Schlüssel-Version
'C0'	'02'	'81 10'	Symmetrischer Schlüssel der Länge 16 Byte
'7B'	'0C'		SE-Datenobjekt
'80'	'01'	'01'	Festlegung für SE #1
'A4'	'07'		AT - Tag und Länge
'95'	'01'	'40'	Usage Qualifier: Nur interne Authentifikation
'89'	'02'	'21 12'	Algorithmus-ID: Schlüssel darf zur Authentifikation der Chipkarte mit Triple-DES verwendet werden

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 109

III.2.2.5 EF_PWD

◆ Beschreibung

Das lokale EF_PWD im DF_BANKING_20 enthält in dem 9 Byte langen Record '01' die Länge der FinTS-PIN und einen Referenzwert der FinTS-PIN der ZKA-Chipkarte. Die FinTS-PIN hat eine Mindestlänge von 5 Ziffern und darf maximal 12 Ziffern lang sein.

◆ Format

Für das EF_PWD des DF_BANKING_20 sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'16'		
'82'	'05'	'12 41 00 09 01'	Datei-Deskriptor für lineares EF mit fester Recordlänge von 9 Byte
'83'	'02'	'00 12'	Datei-ID des EF_PWD
'88'	'01'	'18'	SFI '03' für das EF_PWD
'A1'	'06'	'8B 04 00 30 02 04'	Zugriffsregel-Referenz

Auf das EF_PWD darf nur im SE #2 zugegriffen werden: Die Kommandos APPEND RECORD und UPDATE RECORD dürfen nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden, der Record-Inhalt verschlüsselt (ENC) ist und die Kommandonachricht mit einem MAC abgesichert ist. Verschlüsselung und MAC-Bildung erfolgen dabei mit dem K_{HBCI_Admin} . Der Returncode eines APPEND RECORD oder UPDATE RECORD wird MAC-gesichert. Die MAC-Bildung erfolgt für die Antwortnachricht mit dem K_{HBCI_Admin} . Das Kommando READ RECORD darf nie ausgeführt werden (Zugriffsregel im Record 4 des EF_RULE).

◆ Daten

Der Record '01' des EF_PWD enthält einen Referenzwert der FinTS-PIN.

Byte	Inhalt	Beschreibung
1	'05'	Länge der PIN
2 - 9	'XX..XX'	Referenzwert der PIN

Zur Erzeugung des Referenzwertes wird aus der FinTS-PIN zunächst der 8 Byte lange 'Format 2 PIN Block' gemäß [ISO PIN1] wie folgt gebildet:

C	L	P	P	P	P	P	P/F	P/F	P/F	P/F	P/F	P/F	P/F	F	F
---	---	---	---	---	---	---	-----	-----	-----	-----	-----	-----	-----	---	---

Erläuterung:

Jedes Feld repräsentiert ein Halbbyte.

C:	Kontroll-Feld, binär kodiert	hat immer den Wert '2'
L:	PIN-Länge, binär kodiert	mögliche Werte von '5' bis 'C'
P:	PIN-Ziffer, BCD-kodiert	
F:	Filler, binär kodiert	hat immer den Wert 'F'
P/F:	PIN-Ziffer/Filler	Belegung abhängig von der PIN-Länge

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 110	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Der erzeugte Format 2 PIN Block wird mit PB bezeichnet. Aus diesem PIN Block wird der zu speichernde Referenzwert durch DES-Verschlüsselung mit sich selbst erzeugt:

PIN-Referenzwert: ePB(PB)

Falls erforderlich, wird vor der Verwendung von PB als DES-Schlüssel ein Parity Adjustment vorgenommen. PB wird als Klartext unverändert verwendet.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 111

III.2.2.6 EF_PWDD

◆ Beschreibung

Das EF_PWDD im DF_BANKING_20 enthält in Record '01' die Zusatzinformationen zu der im EF_PWD des DF_BANKING_20 abgelegten FinTS-PIN.

◆ Format

Für das EF_PWDD sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'1C'		
'82'	'05'	'14 41 00 15 01'	Datei-Deskriptor für lineares EF mit variabler Recordlänge (max. 21 Byte) und 1 Record
'83'	'02'	'00 15'	Datei-ID des EF_PWDD
'85'	'02'	'00 15'	Für Nutzdaten allozierter Speicherplatz in Byte
'88'	'01'	'20'	SFI '04' für das EF_PWDD
'A1'	'08'	'8B 06 00 30 01 02 02 05'	Zugriffsregel-Referenzen

Im SE #1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando- und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 dürfen APPEND RECORD und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden und die Kommandonachricht mit einem MAC abgesichert ist. Der Returncode wird MAC-gesichert. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} (Zugriffsregel im Record 5 des EF_RULE).

◆ Daten

Das lokale EF_PWDD enthält in Record '01' einen 21 Byte langen Record, der die Zusatzinformationen zu der FinTS-PIN enthält.

Tag	Länge	Wert	Beschreibung
'93'	'02'	'01 01'	Passwortreferenz: Passwort '01' im Record '01' des EF_PWD
'89'	'02'	'11 50'	Speicherformat des Passwortes (minimal 5 Ziffern)
'7B'	'0B'		SE-DO, Tag und Länge
'80'	'01'	'00'	SE Referenz-DO: Für alle SEs
'A1'	'03'	'8B 01 08'	Zugriffsregel-Referenz
'89'	'01'	'12'	Übertragungsformat der Authentifikationsdaten: PIN Format 2 Block

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 112	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.2.7 EF_FBZ

◆ Beschreibung

EF_FBZ bezeichnet das lineare EF, das in Record '01' den Fehlbedienungszähler und den zugehörigen Initialwert für die im DF-spezifischen EF_PWD abgelegte FinTS-PIN enthält.

◆ Format

Für das EF_FBZ im DF_BANKING_20 sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'18'		Tag und Länge für FCP
'82'	'05'	'12 41 00 02 01'	Datei-Deskriptor für lineares EF fester Recordlänge
'83'	'02'	'00 16'	Datei-ID des EF_FBZ
'88'	'01'	'28'	SFI '05' für das EF_FBZ
'A1'	'08'	'8B 06 00 30 01 02 02 05'	Zugriffsregel-Referenzen

Im SE # 1 dürfen nur die Kommandos READ / SEARCH RECORD mit ungesicherter Kommando- und Antwortnachricht ausgeführt werden (Zugriffsregel im Record 2 des EF_RULE).

Im SE #2 dürfen die Kommandos APPEND RECORD und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden und die Kommandonachricht mit einem MAC abgesichert ist. Der Returncode wird MAC-gesichert. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} (Zugriffsregel im Record 5 des EF_RULE).

◆ Daten

Das EF_FBZ enthält in Record '01' einen 2 Byte langen Record, der den Fehlbedienungszähler und den zugehörigen Initialwert '03' für die FinTS-PIN enthält.

Initialwert des FBZ	FBZ
'03'	'03'

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 113

III.2.2.8 EF_BNK

◆ Beschreibung

Bei dem EF_BNK handelt es sich um ein lineares EF mit 5 Records in dem Bankverbindungen abgelegt sind.

◆ Format

Für das EF_BNK in einer FinTS-Chipkarte sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'18'		Tag und Länge für FCP
'82'	'05'	'12 41 00 58 05'	Datei-Deskriptor für lineares EF mit fester Recordlänge 88 Byte und 5 Records
'83'	'02'	'03 01'	Datei-ID des EF_BNK
'88'	'01'	'D0'	SFI '1A' für das EF_BNK
'A1'	'08'	'8B 06 00 30 01 06 02 03'	Zugriffsregel-Referenzen

Im SE #1 dürfen READ / SEARCH RECORD immer ausgeführt werden, die Antwortnachricht wird nicht abgesichert. UPDATE RECORD darf nur ausgeführt werden, wenn zuvor eine Karteninhaber-Authentifikation mit dem lokalen Passwort 1 (FinTS-PIN) erfolgt ist. Der Returncode wird nicht MAC-gesichert (Zugriffsregeln im Record 6 des EF_RULE).

Im SE #2 darf das Kommando APPEND RECORD nur ausgeführt werden, wenn es mit Secure Messaging durchgeführt wird. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} (Zugriffsregel im Record 3 des EF_RULE).

◆ Daten

Die Records setzen sich aus einer Bankkurzbezeichnung, der Bankleitzahl, dem Kommunikationsdienst, der Adresse und dem Adresszusatz für den Kommunikationszugang, dem Länderkennzeichen und der Benutzerkennung zusammen.

Byte	Länge	Wert	Erläuterung
1-20	20	'aa .. aa'	Kurzbezeichner des Kreditinstituts
21-24	4	'nn nn nn nn'	Kreditinstitutscode des kontoführenden Kreditinstituts
25-25	1	'n'	Kommunikationsdienst (gültige Codierungen siehe III.1.1.2)
26-53	28	'aa .. aa'	Kommunikationsadresse
54-55	2	'aa aa'	Kommunikationsadressenzusatz
56-58	3	'aa aa aa'	Länderkennzeichen des kontoführenden Kreditinstituts
59-88	30	'aa .. aa'	Benutzerkennung

Alphanumerische Feldinhalte ('a') werden ASCII-kodiert, linksbündig eingestellt und mit Leerzeichen (X'20') auf die vorgegebene Länge aufgefüllt. Numerische Feldinhalte ('n') werden BCD-kodiert.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 114	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.2.9 EF_MAC

◆ Beschreibung

Das EF_MAC wird für die MAC-Bildung über den Hash-Wert einer Nachricht benötigt. Es besteht aus einem 12 Byte langem Record deren Zugriffsregeln so gesetzt werden müssen, dass beim Lesen des Records der MAC produziert wird.

◆ Format

Für das EF_MAC sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'18'		Tag und Länge für FCP
'82'	'05'	'12 41 00 0C 01'	Datei-Deskriptor für lineares EF mit einem Record der Länge 12 Byte
'83'	'02'	'03 02'	Datei-ID des EF_MAC
'88'	'01'	'D8'	SFI '1B' für das EF_MAC
'A1'	'08'	'8B 06 00 30 01 07 02 03'	Zugriffsregel-Referenzen

Im SE #1 dürfen READ / SEARCH RECORD nach Karteninhaber-Authentifikation ausgeführt werden, die Antwortnachricht wird mit einem K_{DAK} -MAC versehen. UPDATE RECORD darf nur ausgeführt werden, wenn zuvor eine Karteninhaber-Authentifikation mit dem lokalen Passwort 1 (FinTS-PIN) erfolgt ist. Der Returncode eines UPDATE RECORD wird nicht MAC-gesichert (Zugriffsregeln im Record 7 des EF_RULE).

Im SE #2 darf das Kommando APPEND RECORD nur ausgeführt werden, wenn es mit Secure Messaging durchgeführt wird. Die MAC-Bildung erfolgt für Kommando- und Antwortnachricht mit dem K_{HBCI_Admin} (Zugriffsregel im Record 3 des EF_RULE).

◆ Daten

Das EF_MAC enthält einen Record, der den folgenden Aufbau hat:

Byte	Wert	Erläuterung
1-12	'XX..XX'	Hash-Wert

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 115

III.2.2.10 EF_SEQ

◆ Beschreibung

Bei dem EF_SEQ handelt es sich um ein lineares EF, dessen Record ein 2 Byte langes binär definiertes Element enthält. Dieser binäre aufsteigende Zähler fließt als Signatur-ID zur Absicherung der Daten gegen Doppeleinreichung ein. Der Startwert des Zählers ist 1. Ein Rücksetzen bei Überlauf findet nicht statt.

◆ Format

Für das EF_SEQ sind die folgenden FCP festzulegen:

Tag	Länge	Wert	Erläuterung
'62'	'18'		Tag und Länge für FCP
'82'	'05'	'12 41 00 02 01'	Datei-Deskriptor für lineares EF mit 1 Record der Länge 2 Byte
'83'	'02'	'03 03'	Datei-ID des EF_SEQ
'88'	'01'	'E0'	SFI '1C' für das EF_SEQ
'A1'	'08'	'8B 06 00 30 01 06 02 05'	Zugriffsregel-Referenzen

Im SE #1 dürfen READ / SEARCH RECORD immer ausgeführt werden, die Antwortnachricht wird nicht abgesichert. UPDATE RECORD darf nur ausgeführt werden, wenn zuvor eine Karteninhaber-Authentifikation mit dem lokalen Passwort 1 (FinTS-PIN) erfolgt ist. Der Returncode wird nicht MAC-gesichert (Zugriffsregeln im Record 6 des EF_RULE).

Im SE #2 dürfen die Kommandos APPEND RECORD und UPDATE RECORD nur ausgeführt werden, wenn sie mit Secure Messaging durchgeführt werden. Die MAC-Bildung erfolgt für Kommando- und Antwortnachrichten jeweils mit dem K_{HBCI_Admin} (Zugriffsregel im Record 5 des EF_RULE).

◆ Daten

Das EF_SEQ enthält 1 Record, der den folgenden Aufbau hat:

Byte	Wert	Erläuterung
1-2	'XX XX'	Sequenznummer

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 116	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.3 Platzbedarf der Applikation im Chip

Die Platzbedarfsberechnung ist sehr stark von der Stärke der ROM-Maske abhängig. Der notwendige Platz für die EF-Verwaltung z. B. Record-Nummern- bzw. Adressverwaltung steht im direkten Zusammenhang mit der Verwaltung des E²-PROM. Diese Verwaltung ist Bestandteil der ROM-Maske. Der tatsächliche exakte Platzbedarf kann nur von den ROM-Maskenentwicklern ermittelt werden. Er ist von Chip zu Chip und ROM-Maske zu ROM-Maske unterschiedlich.

♦ Typ 0

Die nachfolgende Tabelle enthält daher nur die Nettodatengröße (in Byte) der "Banking"-Applikation.

Dateiname	Headergröße ⁸	Datengröße
DF_Banking	28	26
EF_KEY	23	17
EF_KEYD	23	5
EF_AUT	23	17
EF_AUTD	23	4
EF_PWD1	25	8
EF_PWDD1	23	5
EF_BNK	23	440
EF_MAC	23	12
EF_SEQ	23	2
	237	536

Demnach hat die Applikation "Banking" einen Mindestplatzbedarf von **773 Byte**.

♦ Typ 1

Die nachfolgende Tabelle enthält daher nur eine grobe Abschätzung der Nettodatengrößen (in Byte) der Applikation. Dabei wurde als Overhead die Größe des jeweiligen FCP zugrundegelegt. Zusätzlich wurde das FMD des DF_BANKING_20 (enthält die vergebenen SFIs sowie deren Pfade) als "Nutzdaten" des DF interpretiert.

Dateiname	Overhead	Nutzdaten
DF_BANKING_20	28	68
EF_KEY	24	54
EF_KEYD	30	72
EF_PWD	24	9
EF_PWDD	30	21
EF_FBZ	26	2
EF_RULE	30	125
EF_BNK	26	440
EF_MAC	26	12
EF_SEQ	26	2
	270	805

⁸ Größenangaben in Byte

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 117

Demnach hat die FinTS-Applikation einen Platzbedarf von ca. **1075 Byte**.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 118	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.4 Terminalabläufe (Typ 0 und Typ 1)

Nachfolgend werden die Anwendungsabläufe aus Endgerätesicht spezifiziert. Hierbei werden ausschließlich die chipkartenbezogenen Aspekte berücksichtigt. Anwendungsbezogene Details sind nicht Bestandteil dieser Spezifikation.

Falls bei der Ausführung der Kommandos ein Fehler auftritt, bricht das Terminal den Vorgang ab, es sei denn, es ist ein abweichendes Verhalten spezifiziert.

III.2.4.1 Startdialog

FinTS-Chipkarte			Endgerät/Gateway	
R1	ATR der FinTS-Chipkarte	←	A1	Anzeige: 'Bitte Karte einstecken'
		→	C1	Reset FinTS-Chipkarte
R2	OK	←	C2	SELECT FILE DF_BANKING(_20)
R3	Kartenidentifikationsdaten (CID)	←	C3	READ RECORD EF_ID
		→	A3	CID prüfen und speichern
R4	OK	←	A4	FinTS-PIN-Eingabe und Formatierung
		→	C4	VERIFY FinTS-PIN
R4	Sequenznummer (SEQ)	←	C5	READ RECORD EF_SEQ
		→	A5	SEQ speichern
R5	Bankverbindung	←	C6	READ RECORD EF_BNK
		→	A6	Daten prüfen und speichern

♦ Erläuterung

1. Nachdem die FinTS-Chipkarte eingesteckt ist, wird ein Reset der Karte durchgeführt (Kommunikationsprotokoll T = 1). Der korrekte ATR und seine Behandlung sind z. B. in [LT] spezifiziert.
2. Die Applikation FinTS-Banking wird geöffnet, indem das ADF der Applikation, DF_BANKING_20 für FinTS-Karten von Typ 1 oder DF_BANKING für FinTS-Karten von Typ 0, durch das Terminal mittels des Kommandos SELECT FILE ausgewählt wird. Dabei wird zunächst versucht, die neue Applikation DF_BANKING_20 zu selektieren. Bei einem Returncode '6A 82' ist die Applikation nicht vorhanden. Es wird dann die "alte" Applikation DF_BANKING selektiert.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 A4'	CLA, INS
3	'04'	P1, Selektion mit DF-Name
4	'0C'	P2, Keine Antwortdaten
5	'09'	L _C
6-14	'D2 76 00 00 25 48 42 0X 00'	AID der FinTS-Applikation (X=1,2)

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 119

Nachdem der Applikationskontext geöffnet ist, können die AEFs der Applikation mittels SFI referenziert werden. Das Terminal hält die Information vor, um welchen Kartentyp es sich handelt

3. Das Terminal liest mittels READ RECORD die Kartenidentifikationsdaten im Record '01' des EF_ID im MF der FinTS-Karte (SFI '19').

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'01'	P1, Recordnummer
4	'CC'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück.

Byte	Wert	Erläuterung
1	'67'	Branchenhauptschlüssel
2-4	'2n nn nn'	Kurz-BLZ kartenausgebendes Kreditinstitut
5-9	'nn..nn'	individuelle Kartennummer
10	'nD'	Prüfziffer für Byte 1 – 9
11-12	'JJ MM'	Verfalldatum der Karte
13-15	'JJ MM TT'	Aktivierungsdatum der Karte
16-17	'0280'	Ländercode
18-20	'45 55 52'	Währungskennzeichen "EUR"
21	'01'	Wertigkeit der Währung
22	'XX'	Chiptyp
23	'00'	Filler
24	'XX'	Betriebssystem-Version
23-24 oder 25-26	'XX XX'	Positiver Returncode SW1 SW2

Die Antwortdaten sind mindestens 22 Byte lang und können für Karten von Typ 1 länger als 24 Byte sein.

Die Kodierung der empfangenen Daten wird geprüft:

Wenn eine Karte von Typ 0 mehr als 22 Byte Antwortdaten ausgibt, oder wenn eine Karte von Typ 1 weniger als 24 Byte Antwortdaten ausgibt, oder wenn Währungskennzeichen in Byte 18-20 oder Wertigkeit der Währung in Byte 21 nicht korrekt kodiert sind, oder wenn eine Karte von Typ 0 das Währungskennzeichen "EUR", oder wenn Byte 24 einer Karte von Typ 1 den Wert '00' hat sowie bei jedem anderen Fehlerfall wird mit einer Fehlermeldung abgebrochen.

4. Das Terminal fordert den Karteninhaber auf, die PIN einzugeben und formatiert dann die eingegebene PIN zum Format 2 PIN-Block FPIN2. Das Terminal baut eine Kommandonachricht für das Kommando VERIFY auf.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 120	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Command APDU:

Byte	Wert	Erläuterung
1	'00 20'	CLA, INS
3	'00'	P1, fester Wert
4	'81'	P2, PIN im EF_PWD1 des DF suchen (bzw. hat PWDID '01')
5	'08'	L _c
6-13	'XX..XX'	FPIN2

Die Chipkarte führt die PIN-Prüfung durch und setzt das Flag des entsprechenden Sicherheitszustands, wenn die PIN-Prüfung erfolgreich war. Andernfalls wird der PIN-Fehlbedienungs-zähler dekrementiert.

Durch den Returncode des Kommandos VERIFY teilt die Chipkarte dem Terminal mit, ob die Prüfung erfolgreich war, bzw. wie viele Versuche noch möglich sind.

5. Das Terminal liest mittels READ RECORD die Sequenznummer im Record '01' des EF_SEQ (SFI '1C').

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'01'	P1, Recordnummer
4	'E4'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück.

Byte	Wert	Erläuterung
1-2	'XX XX'	Sequenz-zähler
3-4	'XX XX'	Positiver Returncode SW1 SW2

Das Terminal speichert den Wert des Sequenz-zählers.

6. Das Terminal liest mittels READ RECORD sukzessive die Bankverbindungsdaten in den Records des EF_BNK (SFI '1A'), bis der "passende" Eintrag gefunden wird.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'0X'	P1, Recordnummer X
4	'D4'	P2, Reference Control Byte
5	'00'	L _e

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 121

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Länge	Wert	Erläuterung
1-20	20	'aa .. aa'	Kurzbezeichner des Kreditinstituts
21-24	4	'nn nn nn nn'	Bankleitzahl des kontoführenden Kreditinstituts
25-25	1	'n'	Kommunikationsdienst
26-53	28	'aa .. aa'	Kommunikationsadresse
54-55	2	'aa aa'	Kommunikationsadressenzusatz
56-58	3	'aa aa aa'	Länderkennzeichen des kontoführenden Kreditinstituts
59-88	30	'aa .. aa'	Benutzerkennung
89-90	2	'XX XX'	Positiver Returncode SW1 SW2

Alternativ kann für Chipkarten vom Typ 1 das Kommando SEARCH RECORD verwendet werden, um mittels eines mit übergebenen Suchmusters den "passenden" Eintrag in einem Schritt zu finden.

Beispiel: Es soll der erste Eintrag zu einer vorgegebenen Bankleitzahl des kontoführenden Instituts (an Byteposition 21-24) gefunden werden:

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 A2'	CLA, INS
3	'01'	P1, Recordnummer an der die Suche startet
4	'D7'	P2, Reference Control Byte (SFI + spezifische Suche)
5	'07'	L _C
6	'04'	Control Byte
7	'14'	Offset 20 = Byte 21
8	'0E'	Konfigurationsbyte: Suche an dieser Position bis zum ersten erfolgreichen Record mit Rückgabe des Inhalts
9-12	'nn nn nn nn'	Bankleitzahl-Suchmuster
13	'00'	L _e

Das Kommando SEARCH RECORD gibt bei erfolgreicher Kommandoausführung die folgende Antwortnachricht aus:

Byte	Wert	Erläuterung
1	'XX'	Recordnummer
2-89	'XX..XX'	Recordinhalt
90-91	'XX XX'	Statusbytes

Es sind auch weitere, umfangreichere Suchoptionen möglich (z. B. alle passenden Einträge ermitteln oder Intervallsuche), siehe hierzu [LIT 1].

III.2.4.2 Nachricht generieren

Dieser Teil des Gesamt Ablaufs ist nur insofern chipkartenrelevant, als Bankverbindungsdaten, die für die Auftragsgenerierung benötigt werden, aus der Chipkarte

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 122	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

entnommen werden. Für die folgende Ablaufbeschreibung wird angenommen, dass die Anwendung bereits FinTS-Nachrichten generiert hat. Diese Nachrichten müssen jetzt ggf. noch kryptographisch gesichert werden, d. h. es werden Segmente für die elektronische(n) Signatur(en) und für die Verschlüsselung entsprechend den FinTS-Spezifikationen eingefügt.

III.2.4.3 Nachricht signieren

Die folgenden Abläufe können offline, d. h. außerhalb eines FinTS-Dialoges vollzogen werden. Dies gilt nicht für die Erstellung von Botensignaturen. Der Grund besteht darin, dass für die Absicherung aller Kreditinstitutsnachrichten der Schlüssel des Boten erforderlich ist. Daher muss während eines gesamten Dialoges die Chipkarte des Boten im Endgerät stecken.

Die Abläufe für die Botensignatur sind grundsätzlich identisch mit den im Folgenden beschriebenen Abläufen für die Erstellung von Auftragssignaturen. Da aber ggf. für die Botensignatur anwendungsseitig noch weitere Chipkartendaten (Benutzerkennung, Benutzerreferenz, Kommunikationszugang etc.) benötigt werden, wird der komplette Ablauf im *III.2.4.5 FinTS-Dialog führen* noch einmal beschrieben.

FinTS-Chipkarte		Endgerät/Gateway	
R1a	KV	← C1a	GET KEYINFO (nur Typ 1)
		→ A1a	Schlüsselversion KV speichern
R1b	OK	← C1b	SELECT EF_KEYD (nur Typ 0)
		→ C1c	READ RECORD EF_KEYD (nur Typ 0)
R1c	Datensatz	→ A1c	Schlüsselversion KV speichern
		A2	Sequenzzähler (Signatur-ID) SEQ inkrementieren
		A3	Signatur-Segment aufbauen und in FinTS-Nachricht einfügen
		A4	Daten (Signatur-Segment, FinTS-Nutzdaten) für MAC-Berechnung bereitstellen
		← M5	MAC über Daten berechnen (siehe <i>III.2.5.1 MAC-Berechnung / Prüfung</i>)
		→ C6	UPDATE RECORD EF_SEQ mit SEQ
R6	OK	← A7	ggf. A2 bis C6 für weitere Nachrichten wiederholen
		A8	signierte FinTS-Nachrichten zur Weiterverarbeitung speichern
		A9	ggf. Startdialog und A1 bis A8 für Mehrfachsignaturen wiederholen

◆ Erläuterung

1. In diesem Schritt stellt das Terminal fest, welcher Daten-Authentifikationsschlüssel KGK_{DAK} bzw. K_{DAK} zur Signatur der Nachricht verwendet werden muss. Dabei wird Schritt 1a *nur* für Karten vom Typ 1, Schritt 1b und 1c *nur* für Karten vom Typ 0 durchgeführt.
- 1a. Falls es sich um eine FinTS-Karte von Typ 1 handelt, wird hierzu das Kommando GET KEYINFO verwendet.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 124	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Command APDU:

Byte	Wert	Erläuterung
1-2	'B0 EE'	CLA,INS
3	'80'	P1 für "DF-spezifisch"
4	'02'	P2, Schlüsselnummer
5	'00'	L _e

Bei der erfolgreichen Ausführung des GET KEYINFO gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Wert	Erläuterung
1	'XX'	1 vorhandene Schlüssel-Version KV
2-3	'XX XX'	Positiver Returncode SW1 SW2

Die Schlüsselversion wird gespeichert.

- 1b. Falls es sich um eine FinTS-Karte von Typ 0 handelt, wird hierzu das EF_KEYD im DF_BANKING mittels SELECT FILE EF_KEYD ausgewählt.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 A4'	CLA, INS
3	'02'	P1, Selektion eines EF im aktuellen DF
4	'0C'	P2, Keine Antwortdaten
5	'02'	L _c
6-7	'00 13'	Datei-ID von EF_KEYD

- 1c. Mittels READ RECORD liest das Terminal aus Record '02' die Zusatzinformationen für den Schlüssel K_{DAK}.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'02'	P1, Recordnummer für logische Schlüsselnr. '02'
4	'04'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wurde, gibt die FinTS-Karte die folgende Antwortnachricht zurück:

Byte	Wert	Erläuterung
1	'02'	Logische Schlüsselnummer
2	'10'	Schlüssellänge
3	'07'	Algorithmus-ID
4	'XX'	Fehlbedienungs-zähler
5	'XX'	Schlüssel-Version
6-7	'XX XX'	Positiver Returncode SW1 SW2

Die Schlüsselversion wird gespeichert.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 125

2. Der zuvor gelesene und gespeicherte Sequenzzähler SEQ wird inkrementiert.
3. Der Signatur-Segment wird aufgebaut und in die FinTS-Nachricht eingefügt.
4. Die Daten (Signatur-Segment, FinTS-Nutzdaten) für die MAC-Berechnung werden bereitgestellt.
5. Der MAC über die Daten wird berechnet (siehe hierzu *III.2.5.1 MAC-Berechnung / Prüfung*).
6. Das Terminal überschreibt den Sequenzzähler in EF_SEQ mit dem inkrementierten Wert. Dies geschieht durch ein UPDATE RECORD EF_SEQ ohne Secure Messaging. Aufgrund der Zugriffsbedingungen für das EF_SEQ kann das Kommando nur ausgeführt werden, wenn zuvor die FinTS-PIN erfolgreich verifiziert wurde.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 DC'	CLA, INS
3	'01'	P1, Recordnummer
4	'E4'	P2, Reference Control Byte (SFI '1C')
5	'02'	Lc
6-7	'XX XX'	neuer Sequenzzähler SEQ

7. Ggf. können die Schritte 2 bis 6 für weitere Nachrichten wiederholt werden. Schritt 1 braucht nicht erneut durchgeführt zu werden, da die zu verwendende Schlüsselsession bereits gespeichert ist..
8. Die signierten FinTS-Nachrichten können zur Weiterverarbeitung gespeichert werden.
9. Ggf. werden Startdialog und die Schritte 1 bis 8 für Mehrfachsignaturen wiederholt.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 126	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.4.4 Nachricht verschlüsseln

FinTS-Chipkarte			Endgerät/Gateway	
R1a	KV	←	C1a	GET KEYINFO (nur Typ 1)
		→	A1a	Schlüsselversion KV speichern
R1b	OK	←	C1b	SELECT EF_AUTD (nur Typ 0)
R1c	Datensatz	←	C1c	READ RECORD EF_AUTD (nur Typ 0)
		→	A1c	Schlüsselversion KV speichern
			A2	Daten (FinTS-Nutzdaten und ggf. Signatur) für die Verschlüsselung bereitstellen
R3	RND	←	C3	GET CHALLENGE
		→	A3	RND als Einmalschlüssel-Hälfte KS_L speichern
R4	$e^* K_{ENC}(KS_L)$	←	C4	INTERNAL AUTHENTICATE mit KS_L
		→	A4	$e^* K_{ENC}(KS_L)$ speichern
R5	RND	←	C5	GET CHALLENGE
		→	A5	RND als Einmalschlüssel-Hälfte KS_R speichern
R6	$e^* K_{ENC}(KS_R)$	←	C6	INTERNAL AUTHENTICATE mit KS_R
		→	A6	$e^* K_{ENC}(KS_R)$ speichern
			A7	$e^* K_{ENC}(KS_L)$ mit $e^* K_{ENC}(KS_R)$ zu $e^* K_{ENC}(KS)$ konkatenieren und speichern
			A8	KS_L mit KS_R zu KS konkatenieren und Daten mit KS verschlüsseln (Triple-DES CBC-Mode, IV=0, X9.23 Padding)
			A9	Verschlüsselungsdaten aufbauen und in FinTS-Nachricht einfügen
			A10	Verschlüsselte Daten als Binärdaten in Verschlüsselungsdaten einfügen
			A11	ggf. A2 bis A10 für weitere Nachrichten wiederholen
			A12	Verschlüsselte und signierte FinTS-Nachrichten zur weiteren Bearbeitung speichern

♦ Erläuterung

- In diesem Schritt stellt das Terminal fest, welche Version des Chiffrierschlüssels KGK_{ENC} bzw. K_{ENC} zur Verschlüsselung der Nachricht verwendet werden muss. Dabei wird Schritt 1a *nur* für Karten vom Typ 1, Schritt 1b und 1c *nur* für Karten vom Typ 0 durchgeführt.
- Falls es sich um eine FinTS-Karte von Typ 1 handelt, wird hierzu das Kommando GET KEYINFO verwendet.

Command APDU:

Byte	Wert	Erläuterung
1-2	'B0 EE'	CLA,INS
3	'80'	P1 für "DF-spezifisch"
4	'03'	P2, Schlüsselnummer
5	'00'	L _e

Bei der erfolgreichen Ausführung des GET KEYINFO gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Wert	Erläuterung
1	'XX'	1 vorhandene Schlüssel-Version KV
2-3	'XX XX'	Positiver Returncode SW1 SW2

Die Schlüsselversion wird gespeichert.

- 1b. Falls es sich um eine FinTS-Karte von Typ 0 handelt, wird hierzu das EF_AUTD im DF_BANKING mittels SELECT FILE EF_AUTD ausgewählt.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 A4'	CLA, INS
3	'02'	P1, Selektion eines EF im aktuellen DF
4	'0C'	P2, Keine Antwortdaten
5	'02'	L _c
6-7	'00 14'	Datei-ID von EF_AUTD

- 1c. Mittels READ RECORD liest das Terminal die Zusatzinformationen für den Schlüssel K_{ENC}. Diese sind im Record '01' des selektierten EF_AUTD zu finden.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 B2'	CLA, INS
3	'01'	P1, Recordnummer für logische Schlüsselnr. '00'
4	'04'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wurde, gibt die FinTS-Karte die folgende Antwortnachricht zurück:

Byte	Wert	Erläuterung
1	'03'	Logische Schlüsselnummer
2	'10'	Schlüssellänge
3	'07'	Algorithmus-ID
4	'XX'	Schlüssel-Version
5-6	'XX XX'	Positiver Returncode SW1 SW2

Die Schlüsselversion wird gespeichert.

2. Die Daten (FinTS-Nutzdaten und ggf. Signatur) für die Verschlüsselung werden bereitgestellt.

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 128	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

3. Mit dem Kommando GET CHALLENGE lässt sich das Terminal eine Zufallszahl von der FinTS-Karte geben.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 84'	CLA, INS
3	'00'	P1
4	'00'	P2
5	'00'	Le

Wenn das Kommando erfolgreich ausgeführt wurde, gibt die FinTS-Karte eine 8 Byte lange Zufallszahl als Antwortdatum aus, die als Einmalschlüssel-Hälfte KS_L gespeichert wird.

4. Mit dem Kommando INTERNAL AUTHENTICATE wird der Wert KS_L von der FinTS-Karte mit dem Schlüssel K_{ENC} verschlüsselt und in der Antwortnachricht als $e^* K_{ENC}(KS_L)$ übergeben.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 88'	CLA, INS
3	'00'	P1
4	'80' oder '83'	P2, Typ 0: '80' (log. Schlüsselnummer '00'), Typ 1: '83' (log. Schlüsselnummer '03')
5	'08'	Lc
6-13	'XX .. XX'	Zufallszahl KS_L
14	'00'	Le

Das Kommando INTERNAL AUTHENTICATE gibt folgende Antwortnachricht zurück:

Byte	Wert	Erläuterung
1-8	'XX .. XX'	Verschlüsselter Wert $e^* K_{ENC}(KS_L)$
9-10	'XX XX'	Positiver Returncode SW1 SW2

5. Mit dem Kommando GET CHALLENGE lässt sich das Terminal eine weitere Zufallszahl von der FinTS-Karte geben, die als Einmalschlüssel-Hälfte KS_R gespeichert wird.
6. Analog zu Schritt 4 wird ein INTERNAL AUTHENTICATE mit KS_R durchgeführt.
7. $e^* K_{ENC}(KS_L)$ wird mit $e^* K_{ENC}(KS_R)$ zu $e^* K_{ENC}(KS)$ konkateniert und gespeichert.
8. KS_L wird mit KS_R zu KS konkateniert und die Daten werden mit KS verschlüsselt (Triple-DES CBC-Mode, IV=0, X9.23 Padding).
9. Die Verschlüsselungsdaten werden aufgebaut und in die FinTS-Nachricht eingefügt.
10. Die verschlüsselten Daten als Binärdaten in die Verschlüsselungsdaten eingefügt.
11. Ggf. werden die Schritte 2 bis 10 für weitere Nachrichten wiederholt (eine Wiederholung von Schritt 1 ist nicht nötig).
12. Die verschlüsselten und signierten FinTS-Nachrichten werden zur weiteren Bearbeitung gespeichert.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 129

III.2.4.5 FinTS-Dialog führen

FinTS-Chipkarte		Endgerät/Gateway		Kreditinstitut	
		A1	Sequenzzähler (Signatur-ID) SEQ inkrementieren		
		A2	Benutzerkennung aus der bereits gelesenen Bankverbindung (EF_BNK) ermitteln		
		A3	Initialisierungsnachricht aufbauen		
		A4	Signatur-Segment aufbauen und in FinTS-Nachricht einfügen		
		A5	Daten (Signatur-Segment, FinTS-Nutzdaten) für MAC-Berechnung bereitstellen		
		← M6 →	MAC über Daten berechnen (siehe <i>III.2.5.1 MAC-Berechnung / Prüfung</i>)		
R7	OK	← C7 →	UPDATE RECORD EF_SEQ mit SEQ		
		A8	Kommunikationszugang aus Bankverbindung herstellen		
		C9	Nachricht (beginnend mit Initialisierungsnachricht) senden	→ ← R9	Antwortnachricht senden
		A10	falls Antwortnachricht verschlüsselt: Daten (Binärdaten aus den Verschlüsselungsdaten) und $d \cdot K_{ENC}(KS)$ aus den Verschlüsselungsdaten für die Entschlüsselung bereitstellen		
		← M11 →	Daten entschlüsseln (siehe <i>III.2.5.2 Entschlüsselung</i>)		
		A12	falls Kreditinstitutsnachricht signiert: Daten (Signatur, Nutzdaten) und Referenz-MAC für MAC-Prüfung bereitstellen		
		← M13 →	MAC über Daten und Referenz-MAC prüfen (siehe <i>III.2.5.1 MAC-Berechnung / Prüfung</i>)		
		A14	C9 bis M13 für alle weiteren FinTS-Nachrichten wiederholen		

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 130	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.5 Makros

III.2.5.1 MAC-Berechnung / Prüfung

FinTS-Chipkarte		Endgerät/Gateway	
		A1	Hash-Wert HASH über Daten berechnen (RIPEMD160)
		A2	HASH zerlegen in HASH _L (die linken 8 Byte von HASH) und HASH _R (die restlichen 12 Byte)
R3	OK	← C3	UPDATE RECORD EF_MAC mit HASH _R
R4	OK	← C4	PUT DATA mit HASH _L (nur Typ 0)
R5	Daten aus EF_MAC mit CFB-64 MAC über HASH _R (identisch mit CBC-MAC über HASH)	← C5	READ RECORD EF_MAC mit Secure Messaging (für Typ 1 wird hier HASH _L mit übergeben)
		→ A5	Bei MAC-Berechnung: MAC zwischenspeichern Bei MAC-Prüfung: MAC aus Kreditinstitutsnachricht mit MAC der Chipkarte vergleichen

♦ Erläuterung

1. Der Hash-Wert HASH wird über die Daten berechnet (RIPEMD160).
2. Der Hash-Wert HASH wird zerlegt in HASH_L (die linken 8 Byte von HASH) und HASH_R (die restlichen 12 Byte).
3. HASH_R wird in den ersten Record des EF_MAC eingetragen. Die Zugriffsbedingung für das EF_MAC stellt sicher, dass das UPDATE-Kommando nur ausgeführt werden kann, wenn zuvor die FinTS-PIN verifiziert wurde.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 DC'	CLA, INS
3	'01'	P1, Recordnummer
4	'DC'	P2, Reference Control Byte (SFI '1B')
5	'0C'	L _C
6-17	'XX .. XX'	Recordinhalt HASH _R

4. Das Terminal übergibt HASH_L mittels PUT DATA an die FinTS-Karte. Dieser Schritt wird *nur* für Karten vom Typ 0 durchgeführt, da für Karten vom Typ 1 der Zufallswert als Bestandteil des Kommandos im nächsten Schritt übergeben wird.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 131

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 DA'	CLA, INS
3-4	'01 00'	P1, P2
5	'08'	L _C
6-13	'XX..XX'	HASH _L

5. Das Terminal liest mittels READ RECORD den soeben in EF_MAC eingetragenen Hash-Wert mit Secure Messaging.

Command APDU für Chipkarten vom Typ 0:

Byte	Wert	Erläuterung
1-2	'04 B2'	CLA, INS
3	'01'	P1, Recordnummer
4	'DC'	P2, Reference Control Byte
5	'00'	L _e

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Wert	Erläuterung
1-12	'XX ... XX'	Recordinhalt HASH _R
13-20	'XX ... XX'	CFB-MAC mit K _{ENC} über die 16 Byte 1-12 '00 00 00 00' mit ICV= HASH _L
21-22	'XX XX'	Positiver Returncode SW1 SW2

Command APDU für Chipkarten vom Typ 1:⁹

Byte	Wert	Erläuterung
1-2	'08 B2'	CLA, INS mit Secure Messaging
3	'01'	P1, Recordnummer
4	'DC'	P2, Reference Control Byte
5	'11'	L _C
6-7	'BA 0C'	Tag und Länge für Response Descriptor
8-9	'B4 0A'	Tag und Länge für CCT
10-11	'87 08'	Tag und Länge für Zufallszahl
12-19	'XX..XX'	Zufallszahl HASH _L
20-22	'96 01 00'	Tag, Länge und Wert des L _e -Datenobjekts
23	'00'	L _e

⁹ Bezüglich der Übergabe von ICVs über Response Descriptors siehe Kapitel 8.6.1.1 von [DATKOM].

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 132	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

Wenn das READ RECORD erfolgreich ausgeführt wird, gibt die FinTS-Karte eine Antwortnachricht mit der folgenden Struktur zurück:

Byte	Wert	Erläuterung
1-2	'81 0C'	Tag und Länge des Klartext-Datenobjekts
3-14	'XX ... XX'	Recordinhalt HASH_R
15-16	'8E 08'	Tag und Länge des MAC-Datenobjekts
17-24	'XX ... XX'	CFB-MAC mit K_{DAK} über die 16 Byte 1-14 '80 00' mit $\text{ICV} = \text{HASH}_L$
25-26	'XX XX'	Positiver Returncode SW1 SW2

Das Terminal speichert den Wert des MAC.

Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI	Version: 4.0	Kapitel: III
Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV	Stand: 09.07.2004	Seite: 133

III.2.5.2 Entschlüsselung

FinTS-Chipkarte		Endgerät/Gateway	
		A1	$d^* K_{ENC}(KS)$ in die zwei Hälften $d^* K_{ENC}(KS_L)$ und $d^* K_{ENC}(KS_R)$ zerlegen
R2	KS_L	← C2	INTERNAL AUTHENTICATE mit $d^* K_{ENC}(KS_L)$
		→ A2	KS_L zwischenspeichern
R3	KS_R	← C3	INTERNAL AUTHENTICATE mit $d^* K_{ENC}(KS_R)$
		→ A3	KS_R zwischenspeichern
		A4	KS_L mit KS_R zu KS konkatenieren und Daten mit KS entschlüsseln (Triple-DES CBC-Mode, IV=0, X9.23 Padding)

♦ Erläuterung

1. $d^* K_{ENC}(KS)$ wird in die zwei Hälften $d^* K_{ENC}(KS_L)$ und $d^* K_{ENC}(KS_R)$ zerlegt.
2. Mit dem Kommando INTERNAL AUTHENTICATE wird der Wert $d^* K_{ENC}(KS_L)$ von der FinTS-Karte mit dem Schlüssel K_{ENC} entschlüsselt und in der Antwortnachricht als KS_L übergeben.

Command APDU:

Byte	Wert	Erläuterung
1-2	'00 88'	CLA, INS
3	'00'	P1
4	'80' oder '83'	P2, Typ 0: '80' (log. Schlüsselnummer '00'), Typ 1: '83' (log. Schlüsselnummer '03')
5	'08'	L_C
6-13	'XX .. XX'	Parameterwert $d^* K_{ENC}(KS_L)$
14	'08'	L_e

Das Kommando INTERNAL AUTHENTICATE gibt folgende Antwortnachricht zurück:

Byte	Wert	Erläuterung
1-8	'XX .. XX'	Entschlüsselter Wert KS_L
9-10	'XX XX'	Positiver Returncode SW1 SW2

KS_L wird gespeichert.

3. Analog zu Schritt 2 wird ein INTERNAL AUTHENTICATE mit $d^* K_{ENC}(KS_R)$ durchgeführt. Das Ergebnis wird als KS_R gespeichert.
4. KS_L wird mit KS_R zu KS konkateniert und die Daten werden mit KS entschlüsselt (Triple-DES CBC-Mode, IV=0, X9.23 Padding).

Kapitel: III	Version: 4.0	Financial Transaction Services (FinTS) Dokument: Security - Sicherheitsverfahren HBCI
Seite: 134	Stand: 09.07.2004	Kapitel: CHIPAPPLIKATIONEN Abschnitt: Chipapplikation für DDV

III.2.6 Übersicht der Chip-Applikations-Parameter (Typ 1)

◆ Dateistruktur

Lage	Datei-ID	Name	SFI	Zugriffsregel SE #1 (Standard)	Zugriffsregel SE #2 (Admin)
MF	'00 03'	EF_ID	'19'		
	'A6 00'	DF_BANKING_20			1
DF_BANKING_20	'00 30'	EF_RULE	'01'	2	3
	'00 10'	EF_KEY	'02'	--	4
	'00 12'	EF_PWD	'03'	--	4
	'00 13'	EF_KEYD	'1E'	2	5
	'00 15'	EF_PWDD	'04'	2	5
	'00 16'	EF_FBZ	'05'	2	5
	'03 01'	EF_BNK	'1A'	6	3
	'03 02'	EF_MAC	'1B'	7	3
	'03 03'	EF_SEQ	'1C'	6	5

◆ Zugriffsregeln

#	READ / SEARCH RECORD	APPEND RECORD	UPDATE RECORD	IN-/EXCLUDE CREATE EF DELETE self	VERIFY CHANGE REF DATA	RESET RETRY COUNTER
1				K_{HBCI_Admin} -MAC		
2	ALW					
3		K_{HBCI_Admin} - MAC	NEV			
4		K_{HBCI_Admin} -ENC-MAC (K) K_{HBCI_Admin} -MAC (A)				
5		K_{HBCI_Admin} -MAC				
6	ALW		FinTS-PIN			
7	FinTS-PIN K_{DAK} -MAC (A)		FinTS-PIN			
8					ALW	K_{HBCI_Admin} - MAC

Die angegebenen Access Conditions gelten sowohl für Kommando- (K) als auch Antwortnachrichten (A), sonst in Klammern eingeschränkt.

◆ Schlüssel der Applikation

Logische Schlüsselnr.	Erlaubte SE #	Schlüssel	Wer kennt den Masterschlüssel
'01'	2	K_{HBCI_Admin}	zuständiges Hintergrundsystem
'02'	1	K_{DAK}	zuständiges Hintergrundsystem
'03'	1	K_{ENC}	zuständiges Hintergrundsystem