Bundesverband der Deutschen Volksbanken und Raiffeisenbanken e. V.
Bundesverband deutscher Banken e. V.
Bundesverband Öffentlicher Banken Deutschlands e. V.
Deutscher Sparkassen- und Giroverband e. V.
Verband deutscher Pfandbriefbanken e. V.

Die Deutsche Kreditwirtschaft

FinTS

Financial Transaction Services

Schnittstellenspezifikation

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: 3.0-FV Stand: 29.11.2018 Final Version

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

Versionsführung

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

Name	Organisation	Datum	Version	Dokumente	Anmerkungen
Stein	SIZ	15.11.2002	3.0-FV	FinTS 3.0 Security - Sicherheitsverfahren HBCI.doc	Frühere Versionen wurden im Rahmen der HBCI-Spezifikation veröffentlicht
Haubner	für SIZ	21.06.2005	3.0-FV	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2005-06-21.doc	Enthält alle bekannt gewordenen Fehler und Klarstellungen bis zum Releasedatum 21.06.2005.
Haubner	für GAD	07.05.2007	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2007-05-07 final version.doc	Enthält die Anpassungen im Zusammenhang mit der Einführung von SECCOS 6 Bankensignaturkarten
Haubner	für GAD	15.05.2008	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2008-05-15 final version.doc	Korrekturen und Klarstellungen zur SECCOS 6 Unterstützung.
Haubner	für SIZ	14.10.2011	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2011-09-23 final version.doc	Ergänzen RAH-Verfahren
Haubner	für SIZ	25.09.2012	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2012-09-25 final version.doc	Einführen DK-Padding bei RAH-Verfahren
Haubner	für SIZ	18.07.2013	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2013-07-18 FV.doc	Klarstellungen und Fehlerkorrekturen, Verweise auf DK Kryptokatalog
Haubner	für SIZ	29.11.2018	3.0	FinTS 3.0 Security - Sicherheitsverfahren HBCI Rel. 2018-11-29 FV.doc	

Änderungen gegenüber der Vorversion

Hinzufügungen und Änderungen sind im Dokument in dieser Farbe und zusätzlich durch Unterstreichung und einen Randbalken markiert. Löschungen sind aufgrund der besseren Übersichtlichkeit nur durch einen Randbalken markiert. Hypertextlinks sind in dieser Farbe markiert. Falls sich die Kapitelnummerierung geändert hat, bezieht sich die Kapitelangabe auf die neue Nummerierung. Aufgrund der umfangreichen Textumstellungen wurden nicht alle Änderungen markiert.

lfd. Nr.	Kapitel	Seitennummer	Kennung[1]	Art[2]	Beschreibung
1	Diverse	Diverse	0408	E	Ergänzen des RAH-Verfahrens und der damit verbundenen Sicherheitsprofile RAH-7, RAH-9 und RAH-10
2	B.2.2.	S. 17ff	0408, 0425	Ä	Anpassen der Abbildungen im Zuge der Einführung des RAH-Verfahrens. Ergänzen des DK-Paddings. Ersetzen des Terminus „HBCI-Nachricht“ durch „FinTS-Nachricht“
3	B.1.1, S. 3			Ä	Anpassen des Passus zu verpflichtenden Sicherheitsprofilen
4	B.2.2.1			Ä	ZKA-Padding, einfügen der AES-Blocklänge=16 Byte für den Wert „L“ Fehlerbehebungen und Klarstellungen in den Abbildungen 1, 2 und 3
5	B.3.1.3.1			Ä	Löschen von Step 5, da nicht mehr relevant.
6	Diverse			Ä	Ersetzen der konkret angegebenen Schlüssellängen durch Referenz auf die Empfehlungen des DK Kryptokatalogs [DK Krypto]
7	C.1.3.2.4.1	S. 99		Ä	Wegfall der Prüfung, ob Ausgangswert = Verschlüsselungsergebnis ist.

Releasedatum 29.11.2018

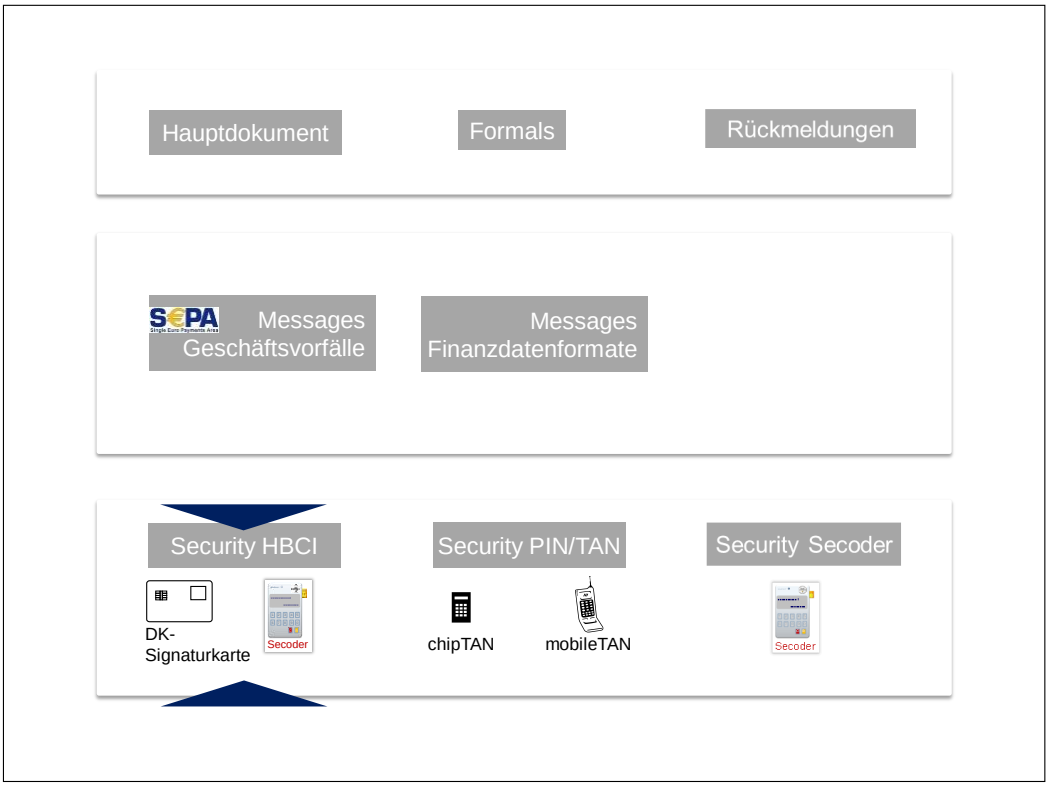
lfd. Nr.	Kapitel	Kapitelnummer	Kennung	Art	Beschreibung
1	Verfahrensbeschreibung	B	0490	Ä/K	Berücksichtigung von PSD2-Anforderungen bei einer Neustrukturierung der unterstützten Sicherheits-Verfahren und –Mechanismen. Entfernen von nicht mehr unterstützten Sicherheitsprofilen.
2	Verfahrensbeschreibung	B	0490	Ä	Festlegen der maximalen RSA-Schlüssellängen auf bis zu 2048 Bit.
3	Verfahrensbeschreibung	B	0456	E	Ergänzen der Zertifikatsverarbeitung nach HICERS zum verpflichtenden Senden von Zertifikaten

[1] nur zur internen Zuordnung
[2] F = Fehler; Ä = Änderung; K = Klarstellung; E = Erweiterung

lfd. Nr.	Kapitel	Kapitel-nummer	Ken-nung	Art	Beschreibung
4	Chipapplikationen	C	0490	Ä	Löschen der Chipkartenapplikation „DF_BANKING“
5	Chipapplikationen	C.1.2.1	0490	Ä	Löschen der EF-NOTPAD Version 001.

Dokumentenstruktur

Das vorliegende Dokument steht in folgendem Bezug zu den anderen Bänden der FinTS V3.0 Spezifikation:



Inhaltsverzeichnis

Versionsführung **1**

Änderungen gegenüber der Vorversion **2**

Dokumentenstruktur **1**

Inhaltsverzeichnis **1**

Abbildungsverzeichnis **3**

A. Einleitung **4**

B. Verfahrensbeschreibung **5**

B.1 Allgemeines **5**

B.1.1 Dynamic Linking und Transparenz der zu signierenden Daten 5

B.1.2 Sicherheitsprofile 5

B.1.3 Kartenbasierte Sicherheitsprofile im Secoder-Applikationsmodus 7

Kartenbasierte 8

B.1.4 Sicherheitsprofile ohne Secoder-Applikationsmodus 8

B.1.5 SW-basiertes Sicherheitsprofil 10

B.1.6 Sicherheitsklassen 10

B.2 Mechanismen **12**

B.2.1 Elektronische Signatur 12

B.2.2 Verschlüsselung 13

B.2.3 Sicherheitsmedien beim Kundenprodukt 15

B.3 Abläufe **17**

B.3.1 Schlüsselverwaltung 17

B.3.2 Schlüsselsperrung 27

B.4 Bankfachliche Anforderungen **29**

B.5 Formate für Signatur und Verschlüsselung **30**

B.5.1 Signaturkopf 31

B.5.2 Signaturabschluss 34

B.5.3 Verschlüsselungskopf 35

B.5.4 Verschlüsselte Daten 36

B.6 Key-Management 37

B.6.1 Formate für Key-Management 37

B.6.2 Key-Management-Nachrichten 45

C. Chipapplikationen 59

C.1 Chipapplikation für RAH 59

C.1.1 Applikation Notepad 59

C.1.2 EF_NOTEPAD 59

C.1.3 Terminalabläufe 70

D. Data Dictionary 83

E. Anlagen 109

E.1 Übersicht der Segmente 109

Abbildungsverzeichnis

Abbildung 1: Nachrichtenverschlüsselung mit AES im CBC-Mode für RAH-Verfahren 14

Abbildung 2: Verschlüsselung bei RAH-7 und RAH-9 15

Abbildung 3: Verschlüsselung bei RAH-10 15

Abbildung 4: Ablauf der Erstinitialisierung bei RAH 24

Abbildung 5: Beispiel für die Gestaltung des Ini-Briefs bei RAH-9 bzw. RAH-10 25

Abbildung 6: Unterstützte Sicherheitsprofilwechsel RDH-1, RDH-2 und RDH-5 auf RAH-9 und RAH-10 46

Abbildung 7: Unterstützte Sicherheitsprofilwechsel beim Übergang von RDH- auf RAH-Verfahren 48

A. EINLEITUNG

In diesem Dokument wird das Sicherheitsverfahren HBCI („Homebanking Computer-Interface“) beschrieben. Dieses Verfahren beruht auf zeitgemäßen kryptographischen Methoden und Algorithmen, wie sie auch durch [PSD2] gefordert sind (z. B. der Digitalen Signatur mit Kryptoverfahren nach Stand der Technik und Chipkartentechnologie).

HBCI kann in multibankfähigen Onlinebanking-Verfahren der Deutschen Kreditwirtschaft eingesetzt werden.

Diese Spezifikation enthält die Formate für Signatur, Verschlüsselung und Keymanagement. Informationen bzgl. FinTS-Nachrichtenaufbau und Dialogablauf sind dem Dokument [Formals] zu entnehmen.

B. VERFAHRENSBESCHREIBUNG

B.1 Allgemeines

Die Verfahrensbeschreibung ist in folgende sechs Abschnitte unterteilt:

1. Allgemeines, Sicherheitsprofile und Sicherheitsklassen
2. Verwendete Sicherheitsmechanismen
3. Abläufe
4. Bankfachliche Anforderungen
5. Segmentformate für Signatur und Verschlüsselung
6. Key-Management

Die Ausführungen lehnen sich an bestehende deutsche Kreditinstitutsstandards (z. B. DFÜ-Abkommen, ec-Chipkarte), sowie an europäische und internationale Standards (z. B. ISO, UN/EDIFACT) an.

Grundsätzlich kommen im Rahmen von HBCI Sicherheitslösungen zum Einsatz, die auf dem asymmetrischen RSA-Verfahren basieren.

Dabei existieren zwei Varianten, die mit RAH (RSA-AES-Hybridverfahren) und einer Sicherheitsprofilnummer gekennzeichnet werden. RAH verwendet RSA-Signaturen und chiffriert den Nachrichtenschlüssel mittels RSA. Die Daten-Verschlüsselung wird durch eine AES-Verschlüsselung mit dem Nachrichtenschlüssel erreicht. Als Träger der privaten Schlüssel dienen Bankensignaturkarten auf Basis des SECCOS-Betriebssystems der Deutschen Kreditwirtschaft oder Softwareschlüssel, die mit einem Kennwort gesichert sind.

B.1.1 Dynamic Linking und Transparenz der zu signierenden Daten

Die Verwendung von RSA-Signaturen sorgt bei HBCI für die Bildung von Authentifizierungscodes gemäß den Anforderungen aus [PSD2] und [RTS]. Die Forderung nach Transparenz relevanter Daten wie z. B. Empfänger-IBAN und Betrag bei Zahlungsverkehrstransaktionen bzgl. Integrität, Authentizität und Vertraulichkeit kann bei HBCI auf zwei Arten erreicht werden:

1. Verwenden eines Secoders im Applikationsmodus „aut“.
2. Einsatz einer Software-Komponente für einen entsprechenden Schutz bei der Anzeige der zu signierenden Daten. Hierbei kann es sich auch um eine separate Anzeige (z. B. eigenes Fenster) in einem FinTS-Kundenprodukt handeln.

Die Verwendung des Secoder-Applikationsmodus in Kombination mit HBCI ist in [Secoder] beschrieben. Für beide Anwendungsszenarien werden im Folgenden geeignete Sicherheitsprofile beschrieben.

B.1.2 Sicherheitsprofile

Die RAH-Sicherheitsverfahren können unterschiedlich parametrisiert werden, wobei Sicherheitsprofile entstehen. Um Multibankfähigkeit zu gewährleisten, ist bei Kommunikation auf Basis von FinTS 3.0 kundenproduktseitig die Unterstützung der Sicherheitsprofile R**A**H-7 und R**A**H-9 auf Basis von Bankensignaturkarten verpflichtend. Zusätzlich kann auch das Sicherheitsprofil R**A**H-10 für SW-basierte Schlüssel verwendet werden. Andere als die genannten Profile sind nicht zulässig.

Das Kreditinstitut teilt dem Kunden die bankseitig unterstützten Profile in den Bankparameterdaten mit. Der Kunde 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 Kunden gewählten Verfahren.

Hier eine Übersicht der zugelassenen Sicherheitsprofile und deren Anwendungsspektrum:

Sicherheitsprofil	Secoder-AZS-funktion	Schlüssellänge	Medium	Bemerkungen
RAH-7	811	..2048	Bankensignaturkarte ≥ SECCOS 6	mit SHA-256, PKCS#1 PSS Padding, AES-Verschlüsselung
RAH-9	811	..2048	Bankensignaturkarte ≥ SECCOS 6	wie RAH-7 ohne Zertifikate
RAH-7		..2048	Bankensignaturkarte ≥ SECCOS 6	mit SHA-256, PKCS#1 PSS Padding, AES-Verschlüsselung
RAH-9		..2048	Bankensignaturkarte ≥ SECCOS 6	wie RAH-7 ohne Zertifikate
RAH-10		..2048	Kennwortgeschützter SW-Schlüssel	mit SHA-256, PKCS#1 PSS Padding, AES-Verschlüsselung

Bei Verwendung der Secodervisualisierung hat die *AZS-Funktion* (enthalten im Datenelement *Sicherheitsfunktion, kodiert*) gemäß [Secoder] folgende Bedeutung:

811 Signatur gemäß HBCI-Sicherheitsprofil RAH-7 bzw. RAH-9 unter Verwendung einer Bankensignaturkarte und der Secoderanwendung *aut*. Die Visualisierung erfolgt im Secoderdisplay analog den Angaben im BPD-Parametersegment *HIVISS*.

Die Angaben zum SECCOS-Betriebssystem bzw. der Betriebssystemversion sind nur als beispielhaft anzusehen; es kann auch jede gleichwertige Signaturkarte verwendet werden, welche die geforderten Verfahren unterstützt.

Die Information über die Betriebssystemversion kann dem Byte 24 in EF_ID entnommen werden. Dort sind derzeit folgende Werte vorgesehen:

X'06': SECCOS 6

X'07': SECCOS 7

B.1.3 Kartenbasierte Sicherheitsprofile im Secoder-Applikationsmodus

Für den Einsatz der folgenden Sicherheitsprofle ist als Chipkartenleser ein Secoder mindestens in Version 2.1 Voraussetzung.

- **RAH-7 im Secoder-Applikationsmodus (AZS-Funktion=811)**

  Als Sicherheitsmedium für das Kundensystem ist nur die Bankensignaturkarte oder eine gleichwertige Signaturkarte zugelassen. Im Applikationsmodus des Secoders haben Signaturen z. B. bei Sicherheitsfunktion 811 folgenden Aufbau:

Parameter	Wert	Bedeutung
Signaturalgorithmus, kodiert	10	RSA
Operationsmodus bei Signatur	19	Signier- und Signaturschlüssel - RSASSA- PSS [PKCS1]
Verwendung des Signaturalgorithmus	6	Owner Signing
Hashalgorithmus, kodiert	6 3	Signierschlüssel - SHA-256 / SHA-256 [SHA-256] Signaturschlüssel - SHA-256 [SHA-256]
Verschlüsselungsalgorithmus, kodiert	14	AES-256 [AES]
Operationsmodus bei Verschlüsselung	18	RSAES-PKCS1-v1_5 [PKCS1]
Schlüsselart	S V D	Signierschlüssel Chiffrierschlüssel Schlüssel für Digitale Signaturen
Schlüssellänge	..2048	
Zertifikatstyp	3	X.509
Zertifikatsinhalt	EF_X509.CH.DS	fortgeschritten, gemäß Sicherheitsklasse

Im Rahmen des Paddingverfahrens RSASSA-PSS wird als „Mask Generation Function“ MGF1 verwendet. Beim Signierschlüssel wird ein doppeltes Hashing (Software und Bankensignaturkarte) durchgeführt. Dies wird durch eine spezielle Ausprägung des „Hashalgorithmus, kodiert“ gekennzeichnet.

Als Salt-Länge (Länge des Initialwertes) ist die Länge des Hashwertes zu verwenden. Diese Festlegung ist z. B. auch Bestandteil der SECCOS Spezifikation.

♦ **RAH-9 im Secoder-Applikationsmodus (AZS-Funktion=811)**

Als Sicherheitsmedium für das Kundensystem ist nur die Bankensignaturkarte oder eine gleichwertige Signaturkarte zugelassen. Im Applikationsmodus des Secoders haben Signaturen bei Sicherheitsfunktion 811 folgenden Aufbau:

Parameter	Wert	Bedeutung/Anmerkung
Signaturalgorithmus, kodiert	10	RSA
Operationsmodus bei Signatur	19	RSASSA-PSS [PKCS1]
Verwendung des Signaturalgorithmus	6	Owner Signing
Hashalgorithmus, kodiert	6	SHA-256 / SHA-256 [SHA-256]
Verschlüsselungsalgorithmus, kodiert	14	AES-256 [AES]
Operationsmodus bei Verschlüsselung	18	RSAES-PKCS1-v1_5 [PKCS1]
Schlüsselart	S V	Signierschlüssel Chiffrierschlüssel
Schlüssellänge	..2048	
Zertifikatstyp		ohne
Zertifikatsinhalt	nicht spezifiziert	

Im Rahmen des Paddingverfahrens RSASSA-PSS wird als „Mask Generation Function“ MGF1 verwendet. Beim Signierschlüssel wird ein doppeltes Hashing (Software und Bankensignaturkarte) durchgeführt. Dies wird durch eine spezielle Ausprägung des „Hashalgorithmus, kodiert“ gekennzeichnet.

nden. Diese Festlegung ist z. B. auch Bestandteil der SECCOS Spezifikation.

B.1.4 Kartenbasierte Sicherheitsprofile ohne Secoder-Applikationsmodus

X’06’: SECCOS 6

X‘07‘: SECCOS 7

- **RAH-7**

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

Parameter	Wert	Bedeutung
Signaturalgorithmus, kodiert	10	RSA
Operationsmodus bei Signatur	19	Signier- und Signaturschlüssel - RSASSA- PSS [PKCS1]
Verwendung des Signaturalgorithmus	6	Owner Signing
Hashalgorithmus, kodiert	6 3	Signierschlüssel - SHA-256 / SHA-256 [SHA-256] Signaturschlüssel - SHA-256 [SHA-256]
Verschlüsselungsalgorithmus, kodiert	14	AES-256 [AES]
Operationsmodus bei Verschlüsselung	18	RSAES-PKCS1-v1_5 [PKCS1]
Schlüsselart	S V D	Signierschlüssel Chiffrierschlüssel Schlüssel für Digitale Signaturen
Schlüssellänge	..2048	
Zertifikatstyp	3	X.509
Zertifikatsinhalt	EF_X509.CH.DS	fortgeschritten, gemäß Sicherheitsklasse

Im Rahmen des Paddingverfahrens RSASSA-PSS wird als „Mask Generation Function“ MGF1 verwendet. Beim Signierschlüssel wird ein doppeltes Hashing (Software und Bankensignaturkarte) durchgeführt. Dies wird durch eine spezielle Ausprägung des „Hashalgorithmus, kodiert“ gekennzeichnet.

Als Salt-Länge (Länge des Initialwertes) ist die Länge des Hashwertes zu verwenden. Diese Festlegung ist z. B. auch Bestandteil der SECCOS Spezifikation.

- **RAH-9**

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

Parameter	Wert	Bedeutung/Anmerkung
Signaturalgorithmus, kodiert	10	RSA
Operationsmodus bei Signatur	19	RSASSA-PSS [PKCS1]
Verwendung des Signaturalgorithmus	6	Owner Signing
Hashalgorithmus, kodiert	6	SHA-256 / SHA-256 [SHA-256]
Verschlüsselungsalgorithmus, kodiert	14	AES-256 [AES]
Operationsmodus bei Verschlüsselung	18	RSAES-PKCS1-v1_5 [PKCS1]
Schlüsselart	S V	Signierschlüssel Chiffrierschlüssel
Schlüssellänge	..2048	
Zertifikatstyp		ohne
Zertifikatsinhalt	nicht spezifiziert	

Im Rahmen des Paddingverfahrens RSASSA-PSS wird als „Mask Generation Function“ MGF1 verwendet. Beim Signierschlüssel wird ein doppeltes Hashing (Software und Bankensignaturkarte) durchgeführt. Dies wird durch eine spezielle Ausprägung des „Hashalgorithmus, kodiert“ gekennzeichnet.

Als Salt-Länge (Länge des Initialwertes) ist die Länge des Hashwertes zu verwenden. Diese Festlegung ist z. B. auch Bestandteil der SECCOS Spezifikation.

B.1.5 SW-basiertes Sicherheitsprofil

♦ **RAH-10**

Als Sicherheitsmedium für das Kundensystem ist eine RSA-Softwarelösung unter folgenden Rahmenbedingungen zugelassen.

Parameter	Wert	Bedeutung/Anmerkung
Signaturalgorithmus, kodiert	10	RSA
Operationsmodus bei Signatur	19	RSASSA-PSS [PKCS1]
Verwendung des Signaturalgorithmus	6	Owner Signing
Hashalgorithmus, kodiert	6	SHA-256 / SHA-256 [SHA-256]
Verschlüsselungsalgorithmus, kodiert	14	AES-256 [AES]
Operationsmodus bei Verschlüsselung	2	Zero-Padding
Schlüsselart	S V	Signierschlüssel Chiffrierschlüssel
Schlüssellänge	..2048	
Zertifikatstyp	3	X.509
Zertifikatsinhalt	nicht spezifiziert	

Im Rahmen des Paddingverfahrens RSASSA-PSS wird als „Mask Generation Function“ MGF1 verwendet. Beim Signierschlüssel wird ein doppeltes Hashing durchgeführt. Dies wird durch eine spezielle Ausprägung des „Hashalgorithmus, kodiert“ gekennzeichnet.

Als Salt-Länge (Länge des Initialwertes) ist die Länge des Hashwertes zu verwenden.

B.1.6 Sicherheitsklassen

Die Sicherheitsklasse gibt für jede Signatur den erforderlichen Sicherheitsdienst an. Als Sicherheitsdienst gelten derzeit „Authentikation“ und „Non-Repudiation“.

Der Sicherheitsdienst „Authentikation“ erfordert die Signatur mit der Schlüsselart „S“ (Schlüssel auf Kundenseite: S_K.CH.AUT$_{C/S}$). Der Sicherheitsdienst „Non-Repudiation“ erfordert die Signatur mit der Schlüsselart „D“ (Schlüssel auf Kundenseite: S_K.CH.DS).

Derzeit sind folgende Sicherheitsklassen zulässig:

Code	Bedeutung
0	kein Sicherheitsdienst erforderlich
1	Sicherheitsdienst „Authentikation“
2	Sicherheitsdienst „Authentikation“ mit fortgeschrittener elektronischer Signatur gemäß §2, SigG und optionaler Zertifikatsprüfung unter Verwendung des S-Schlüssels (Schlüssel S_K.CH.$AUT_{C/S}$)
3	Sicherheitsdienst „Non-Repudiation“ mit fortgeschrittener elektronischer Signatur gemäß §2, SigG und optionaler Zertifikatsprüfung unter Verwendung des DS-Schlüssels (S_K.CH.DS)
4	Sicherheitsdienst „Non-Repudiation“ mit fortgeschrittener bzw. qualifizierter elektronischer Signatur gemäß §2, SigG und zwingender Zertifikatsprüfung unter Verwendung des DS-Schlüssels (S_K.CH.DS)

Die Festlegungen durch die Sicherheitsklasse zum verpflichtenden Senden von Zertifikaten wird auch durch die Zertifikatssteuerung (Parametersegment HICERS) beeinflusst (siehe Kapitel B.3.1.1.1.1).

Folgende Zuordnungen von Sicherheitsklassen auf Sicherheitsprofile sind möglich:

Sicherheitsprofil	Sicherheitsklasse(n)
RAH-7	1, 2, 3, 4
RAH-9	1, 2
RAH-10	1

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

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

- Technische Signaturen (Dialoginitialisierung, Dialogendenachricht) erfolgen generell mit Sicherheitsklasse 1 (Authentikation)
- Bei Geschäftsvorfällen kann das Kreditinstitut die Sicherheitsklasse individuell festlegen (Die Sicherheitsklasse wird dem Kunden in den Bankparameterdaten des betreffenden Geschäftsvorfalls mitgeteilt)

Hinweis:
Sicherheitsklassen werden nur in Verbindung mit dem Sicherheitsverfahren HBCI benutzt. Unterstützt ein Kreditinstitut ausschließlich das PIN/TAN-Verfahren, so ist in das DE ‚Sicherheitsklasse’ des jeweiligen Geschäftsvorfallparametersegmentes als Füllwert ‚0’ einzustellen. Die Sicherheitsklasse hat bei PIN/TAN für die Verarbeitung keine Bedeutung und darf vom Kundenprodukt für PIN/TAN nicht ausgewertet werden. Stattdessen sind bei PIN/TAN die Informationen aus HIPINS für die Festlegung benötigter Sicherheitsmerkmale zu verwenden.

B.2 Mechanismen

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

B.2.1 Elektronische Signatur

Die Bildung der elektronischen Signatur erfolgt durch die Vorgänge

- Bildung des Hashwerts
- Ergänzen des Hashwerts auf eine vorgegebene Länge und
- Berechnung der elektronischen Signatur über den Hashwert.

Je nach Sicherheitsverfahren sind die Verarbeitungsschritte jeweils verschieden.

B.2.1.1 Hashing

Als Hash-Funktion kommt im Rahmen von HBCI derzeit ausschließlich SHA-256 [SHA-256] zum Einsatz.

♦ **SHA-256**

Der Hash-Algorithmus SHA-256 bildet Eingabe-Bitfolgen beliebiger Länge auf Bytefolgen von 32 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-256 verarbeitet die Eingabe-Bitfolgen in Blöcken von 64 Byte Länge.

B.2.1.2 Elektronische Signatur bei RAH (RSA-basierend)

1. Hashing der Nachricht

Als Hash-Funktion wird SHA-256 eingesetzt.

2. Formatierung des Hashwerts

Die Formatierung des Hashwerts erfolgt gemäß PKCS#1 PSS

3. Berechnung der elektronischen Signatur

Der Hashwert wird mittels RSA gemäß PKCS#1 PSS signiert.

B.2.2 Verschlüsselung

Bei jeder Verschlüsselung wird ein separater Einmalschlüssel verwendet. Die Verschlüsselung der FinTS-Nachricht erfolgt mittels AES-256 gemäß [AES]. Der hierfür benötigte Nachrichtenschlüssel wird mittels RSA (RAH) chiffriert und mit der verschlüsselten FinTS-Nachricht mitgeliefert.

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

Sollte bei der Verarbeitung des Nachrichtenschlüssels, insbesondere beim Padding ein Fehler auftreten, so sind außer dem negativen Prüfergebnis selbst keine weiteren Details an die aufrufende Funktion zurückzugeben, um keine Rückschlüsse über die Art des Fehlers und damit ggf. auf den Schlüssel selbst zu geben.

B.2.2.1 Verschlüsselung bei RAH-7, RAH-9 und RAH-10:

Die Verschlüsselung und Entschlüsselung erfolgt bei den RAH-Verfahren in den folgenden drei Schritten:

1. Der Sender erzeugt eine Zufallszahl als Nachrichtenschlüssel.
2. Dieser Nachrichtenschlüssel wird verwendet, um die Daten mittels AES im CBC Modus gemäß ISO 10116 (ANSI X3.106) zu verschlüsseln (vgl. Abbildung 1). Das Padding der Nachricht erfolgt gemäß den Vorgaben des Kryptokatalogs der Deutschen Kreditwirtschaft (vgl. [DK Krypto], Kapitel 4.3.1) (vgl. *Abbildung 2* und *Abbildung 3*).

„ZKA-Padding“ (vgl. [DK Krypto], Kapitel 4.3.1 auf S. 20):

Für die Verarbeitung von Daten durch einen kryptographischen Algorithmus kann deren Darstellung als Folge von Byte-Blöcken mit einer vorgegebenen Länge L erforderlich sein. Das ZKA-Padding ist eine Methode zur Formatierung des letzten, möglicherweise unvollständigen Datenblocks auf die Länge von L Byte. Die den Daten zugehörigen Bytes können eindeutig von den durch das Padding hinzugefügten Bytes unterschieden werden.

An die Daten M wird zunächst das Byte '80' angehängt. Falls M || '80' nun eine Byte-Länge besitzt, die kein Vielfaches von L ist, werden weitere Bytes '00' angehängt, bis das Ergebnis der Operation eine Byte-Länge besitzt, die ein Vielfaches von L ist.

$$\text{ZKA-Padding}(M) = M \,||\, '80' \,||\, \underbrace{'00' \,||\, ... \,||\, '00'}$$

Verkettung bis zur
Gesamtlänge der Byte-Folge
als Vielfaches von L Byte
(hier: AES-Blocklänge = 16 Byte)

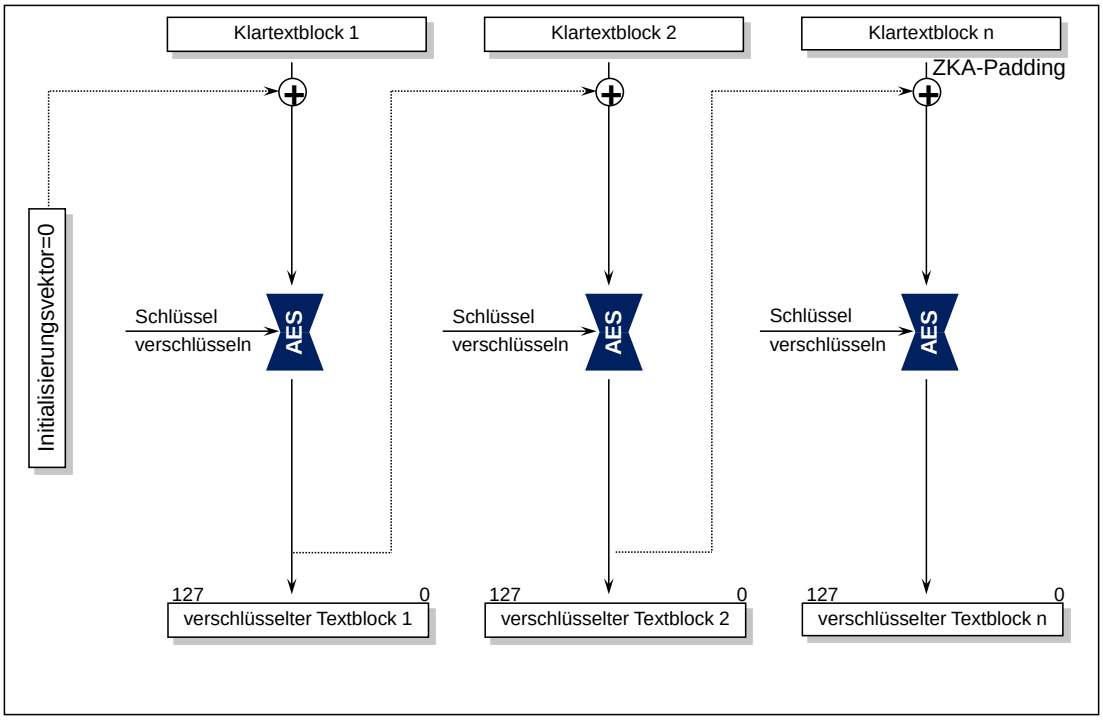


Abbildung 1: Nachrichtenverschlüsselung mit AES im CBC-Mode für RAH-Verfahren

Hinweis zu den Abbildungen: Die Angabe der Längen erfolgt in Form von Bitpositionen (z. B. 127 … 0), um grafisch zu zeigen, an welcher Stelle das Padding erfolgt.

3. Der aktuelle Nachrichtenschlüssel wird mit dem öffentlichen Schlüssel des Empfängers chiffriert. Da die Länge des Nachrichtenschlüssels bei AES nur 32 Byte, d.h. 256 Bit beträgt, muss er auf die Moduluslänge des verwendeten öffentlichen Chiffrierschlüssels ergänzt werden. Das Padding wird abhängig vom Sicherheitsprofil auf unterschiedliche Art und Weise vorgenommen, wie in den folgenden Abbildungen gezeigt.

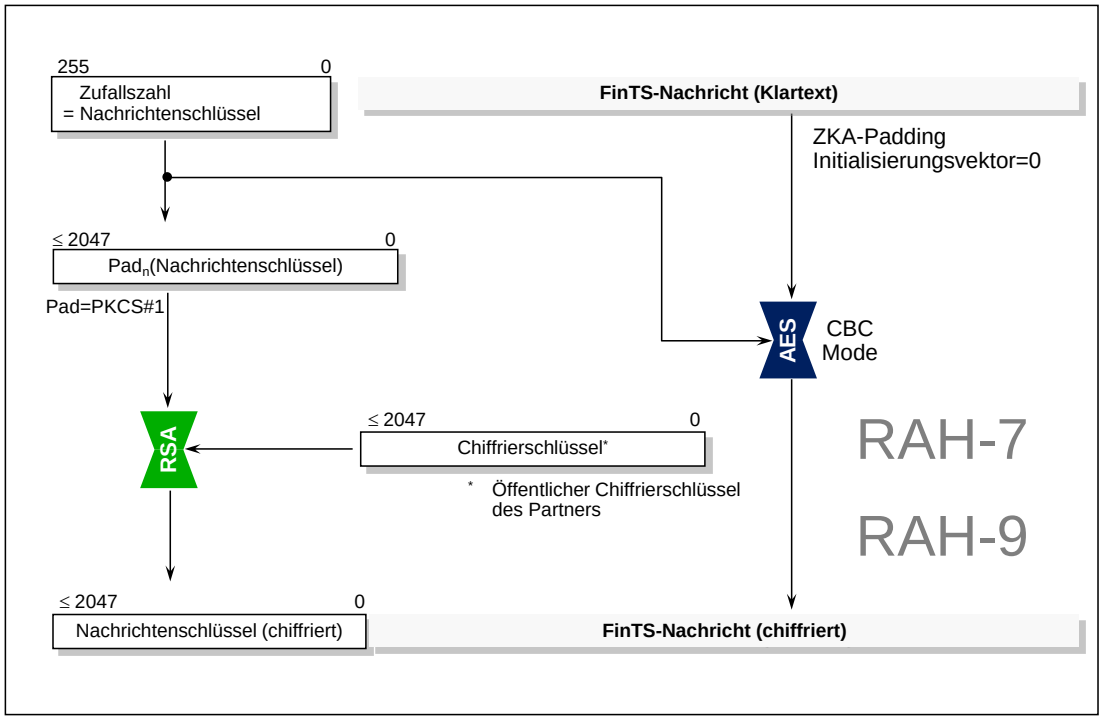


Abbildung 2: Verschlüsselung bei RAH-7 und RAH-9

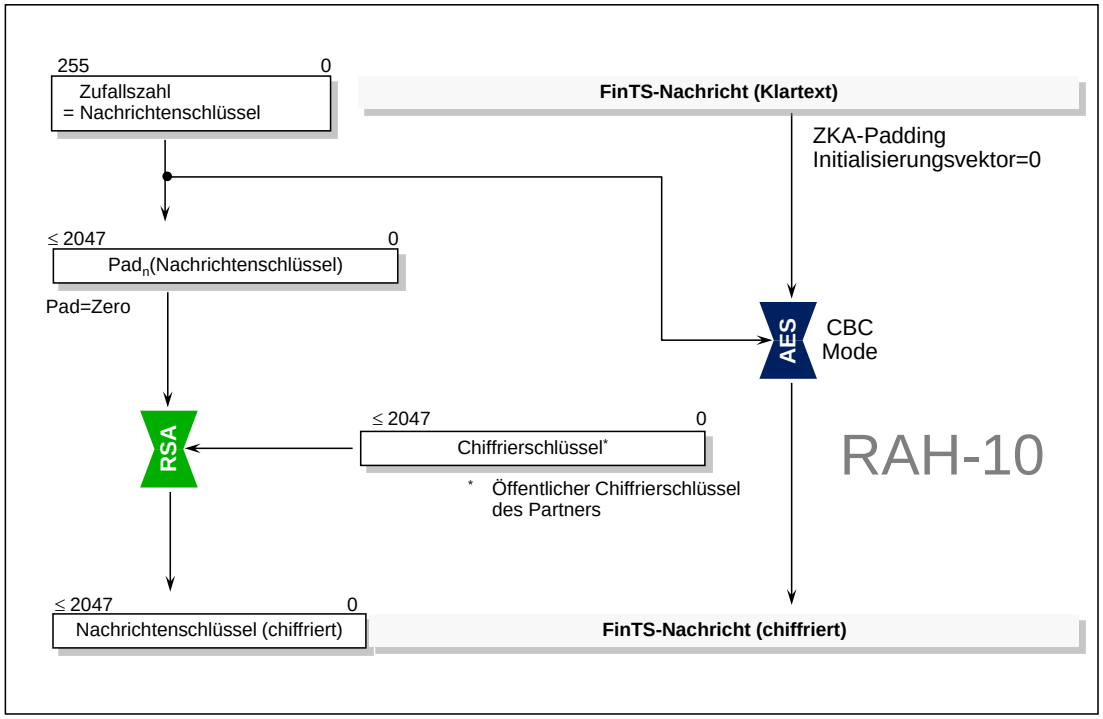


Abbildung 3: Verschlüsselung bei RAH-10

B.2.3 Sicherheitsmedien beim Kundenprodukt

Als Sicherheitsmedien können Bankensignaturkarten oder Softwareschlüssel verwendet werden.

Bei Verwendung einer vom Kreditinstitut ausgegebenen Signaturkarte muss die Berechnung der kryptographischen Funktionen so durchgeführt werden, dass die kartenindividuellen Schlüssel niemals die Chipkarte verlassen. Es wird die in Kap. C.1 beschriebene Bankensignaturkarte empfohlen.

Auf dem Sicherheitsmedium wird unter anderem der private Schlüssel des Kunden gespeichert. Es ist aber auch möglich, öffentliche Schlüssel des Kreditinstitutes darauf abzulegen oder aber im Falle einer Chipkarte die kryptographischen Operationen damit durchzuführen. Generell müssen die geheimen Daten (z. B. private Schlüssel, Passworte) gegen unberechtigtes Auslesen geschützt sein.

Es ist zwingend erforderlich, die Daten auf dem Sicherheitsmedium (kryptographisch) zu schützen. Speziell ist im Rahmen der Speicherung von Softwareschlüsseln sicherzustellen, dass die Daten unter Einbeziehung eines vom Benutzer frei wählbaren Kennwortes verschlüsselt werden und der Zugriff auf die verschlüsselten Daten nur über die manuelle Eingabe des entsprechenden Kennwortes möglich ist.

B.3 Abläufe

B.3.1 Schlüsselverwaltung

♦ **Schlüsselarten**

Bei den Sicherheitsverfahren RAH-9 und RAH-10 können Kunde und Kreditinstitut über zwei Schlüssel bzw. Schlüsselpaare verfügen:

- einen Signierschlüsselpaar
- einen Chiffrierschlüsselpaar

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

Bei RAH-7 können Kunde und Kreditinstitut über bis zu drei Schlüssel bzw. Schlüsselpaare verfügen:

- ein Schlüsselpaar für digitale Signaturen (DS-Schlüssel)
- ein Signierschlüsselpaar (Authentikation)
- ein Chiffrierschlüsselpaar

Abhängig von der Personalisierung der Chipkarte können Signier- und Chiffrierschlüsselpaare 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 Softwareschlüsseln gemäß Sicherheitsprofil RAH-10 wird dringend empfohlen, dass getrennte Signier- und Chiffrierschlüsselpaare 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 Kreditinstitutscode (Bankleitzahl), vgl. [Formals], Kapitel II.5.3.2)
- Benutzerkennung
  (max. 30 Byte, kann vom Kreditinstitut festgelegt werden, vgl. [Formals], Kapitel III.1.1)
- Schlüsselart
  (1 Byte, D: DS-Schlüssel; S: Signierschlüssel; V: Chiffrierschlüssel)
- Schlüsselnummer
  (max. 3 Byte)

- Versionsnummer
  (max. 3 Byte)

Falls kein öffentlicher Schlüssel des Kreditinstituts vorliegt, so ist als Versionsnummer der Wert „999“ einzustellen. 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 Nachrichtenschlüsseln**

Zur Chiffrierung von Nachrichten wird ein dynamisch erzeugter Nachrichtenschlüssel verwendet, der durch Generieren einer 32 Byte langen Zufallszahl gebildet wird.

♦ **Schlüsselgenerierung bei RAH**

Die Schlüsselpaare des Kunden sind vom Kundenprodukt 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:[7]

1. Es wird ein konstanter öffentlicher Exponent e und ein für jeden Kunden 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 RAH-7, RAH-9 und RAH-10 ≤ 2047 Bit.

Schlüsselgenerierung bei RAH10:
Das Kundensystem muss sicherstellen, dass die Schlüssellänge eines neu generierten Schlüsselpaares des Kunden gleich der Länge des öffentlichen Signierschlüssels des Instituts ist, falls das Institut Institutssignaturen unterstützt. Anderenfalls ist die Länge des Chiffrierschlüssels maßgebend.

B.3.1.1.1 Behandlung von Zertifikaten

In FinTS ist die Verwendung von Zertifikaten durch die vorgesehenen Elemente unterstützt, es existieren derzeit jedoch außer der Zertifikatssteuerung durch das Parametersegment HICERS (siehe Kapitel B.3.1.1.1.1) keine Prozesse für das Zertifikatsmanagement.

Folgende Festlegungen gelten für die Belegung der Zertifikats-Datenelemente in den FinTS-Segmenten:

[7] Das Verfahren entspricht dem des DFÜ-Abkommens.

1. Bei Verwendung des Signaturschlüssels (D-Schlüssel) mit Sicherheitsklasse 3 bzw. 4 wird grundsätzlich in allen Nachrichten das entsprechende Zertifikat im Signaturkopf mit geschickt.

   Bei Verwendung des Authentifikationsschlüssels (S-Schlüssel) mit Sicherheitsklasse 2 kann das entsprechende Zertifikat in den Signaturkopf eingestellt werden.

   Ebenso kann ein Benutzer das Zertifikat seines Chiffrierschlüssels (V-Schlüssel) in den Verschlüsselungskopf einstellen. Ggf. dort eingestellte Verschlüsselungszertifikate können vom Institut ignoriert werden. Wird eine Kundennachricht nicht signiert (z. B. optional bei HKEND), so muss bei Verwendung von zertifikatsbasierten Sicherheitsverfahren das Zertifikat des Chiffrierschlüssels in den Verschlüsselungskopf eingestellt werden. Damit ist das Institut in der Lage, die Antwort an das Kundenprodukt verschlüsselt zu übertragen und muss den Dialog nicht mit einer Fehlermeldung wegen eines fehlenden Verschlüsselungsschlüssels des Benutzers beenden. Vorgaben für die Belegung der Zertifikatselemente werden außer durch die Sicherheitsklasse des Geschäftsvorfalls (vgl. Kapitel B.1.6) auch durch die Zertifikatssteuerung (vgl. Kapitel B.3.1.1.1.1) festgelegt.

2. Erstmalige Übermittlung Kundenschlüssel bzw. Schlüsseländerung

   Bei der erstmaligen Übermittlung der Kundenschlüssel bzw. bei der Schlüsseländerung wird grundsätzlich der Authentifikationsschlüssel (S-Schlüssel) und wahlweise das zugehörige Zertifikat verwendet. Das Zertifikat wird in diesem Fall nur in das vorgesehene Element im Geschäftsvorfall (HKSAK bzw. HKISA) eingestellt (nicht in den Signaturkopf).

3. Signaturkarten-Profil mit drei unterschiedlichen Schlüsseln

   Wenn ein Signaturkarten-Profil mit drei unterschiedlichen Schlüsseln verwendet wird, muss bei der erstmaligen Übermittlung der Kundenschlüssel bzw. der Schlüsseländerung auch die Möglichkeit bestehen, das Zertifikat für den eigenen Verschlüsselungsschlüssel im jeweiligen Geschäftsvorfall (HKSAK bzw. HKISA) mitzuschicken.

B.3.1.1.1.1 Parametersegment Zertifikatssteuerung (HICERS)

Das Parametersegment „Zertifikatssteuerung (HICERS)“ legt übergreifende Regeln für die Übermittlung von Zertifikatsinformationen zwischen Kunden- und Institutssystem fest und ermöglicht so die Verwendung von Zertifikaten in FinTS ohne institutseigene Schlüsselverwaltung.

Ohne Verwendung der Zertifikatssteuerung werden ggf. auf der Bankensignaturkarte enthaltene Zertifikate nur im Rahmen der Schlüsselersteinreichung bzw. –Änderung und bei Geschäftsvorfällen in Abhängigkeit von der Sicherheitsklasse vom Kundensystem zum Kreditinstitut gesendet.

Ist in den BPD ein Parametersegment „Zertifikatssteuerung (HICERS)“ enthalten, so befinden sich darin zusätzliche Informationen, die im Speziellen die Übertragung von Zertifikaten während der Dialoginitialisierung regeln. Durch entsprechende Parametrisierung kann aber auch die Steuerung über die GV-Sicherheitsklasse außer

Kraft gesetzt und damit sichergestellt werden, dass bei jedem Geschäftsvorfall relevante Zertifikatsinformationen mitgeschickt werden.

Abhängig von den Parametern der Zertifikatssteuerung gelten folgende Regeln für die Belegung der Zertifikatselemente in den Kundennachrichten:

D-Schlüssel	DEG „Zertifikat“ im Signaturkopf	bei Geschäftsvorfällen der Sicherheitsklasse 3 oder 4
S-Schlüssel	DEG „Zertifikat“ im Signaturkopf	Bei der Dialoginitialisierung und ggf. Geschäftsvorfällen der Sicherheitsklasse 1 oder 2
V-Schlüssel	DEG „Zertifikat“ im Verschlüsselungskopf	Bei der Dialoginitialisierung und ggf. Geschäftsvorfällen der Sicherheitsklasse 1 bis 4

Realisierung Bank: verpflichtend, falls ein Institut die Zertifikatssteuerung verwendet
Realisierung Kunde: verpflichtend, falls ein Institut die Zertifikatssteuerung verwendet

c) Bankparameterdaten

♦ Format

Name: Zertifikatssteuerung Parameter
Typ: Segment
Segmentart: Geschäftsvorfall mit Parametern
Kennung: HICERS
Bezugssegment: HKVVB
Version: 1
Sender: Kreditinstitut

Nr.	Name	Version	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	1	DEG			M	1	
2	Maximale Anzahl Aufträge	1	DE	num	..3	M	1	
3	Anzahl Signaturen mindestens	1	DE	num	1	M	1	0, 1, 2, 3
4	Sicherheitsklasse	1	DE	code	1	M	1	0, 1, 2, 3, 4
5	Parameter Zertifikatssteuerung	1	DEG			M	1	

♦ **Ausgewählte Beispiele für Rückmeldungscodes**

Code	Beispiel
9351	Zertifikat noch nicht gültig
9352	Zertifikat zurückgezogen bzw. gesperrt
9353	Zertifikatssignatur falsch
9354	Zertifizierungsinstanz (Herausgeber) nicht akzeptiert
9355	Fehler im Zertifikatsaufbau
9356	Zertifikatstyp nicht akzeptiert
9357	Zertifikat erwartet aber nicht im Signaturkopf enthalten
9357	Zertifikat erwartet aber nicht im Chiffrierkopf enthalten

B.3.1.1.2 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 Kunden eingegeben werden)
- Softwareschlüssel z. B. auf einem USB-Stick des Kreditinstitutes mit folgendem Inhalt:
  - Segment HIUPA der UPD inkl. Benutzerkennung
  - Segment HIKOM 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 Kunde und Kreditinstitut erfolgen. Alternativ erfolgt dieser Austausch durch eine Anforderung der Zertifikate bei den jeweiligen Zertifizierungsinstanzen. Dieser Prozess findet außerhalb des FinTS-Protokolls statt und wird daher hier nicht näher beschrieben.

Erfolgt der Schlüsselaustausch im Rahmen eines FinTS-Dialoges ist hierzu folgender Ablauf vorgesehen:

♦ **Initiale Schlüsselverteilung des Kreditinstituts**

1. Das Kreditinstitut übermittelt seinen öffentlichen Chiffrierschlüssel an den Kunden. 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 einer Chipkarte z. B. bei Vertragseröffnung.
   - Übertragung der Schlüssel beim Erstzugang (z. B. bei Softwareschlüsseln)

     (1) Der Benutzer fordert die öffentlichen Schlüssel und die BPD mit Hilfe der Key-Management-Nachricht „Erstmalige Anforderung der Schlüssel des

Kreditinstituts“ (s. Kap. B.6.2.1) an. Diese Nachricht ist weder signiert noch chiffriert.

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

Fall A: Das Kreditinstitut signiert

Der Kunde 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 Kunde 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 Kunde → Kreditinstitut

Diese Nachricht wird von einem Ini-Brief an den Benutzer begleitet. Die Gestaltung ist dem Kreditinstitut freigestellt, sollte sich aber am Muster in *Abbildung 5: Beispiel für die Gestaltung des Ini-Briefs bei RAH-9 bzw. RAH-10* orientieren. Der Ini-Brief enthält für den Fall A Exponent und Modulus des Signierschlüssels sowie dessen Hashwert und für den Fall B Exponent und Modulus des Chiffrierschlüssels sowie dessen Hashwert.

Exponent und Modulus sind dabei mit führenden Nullen (X’00’) auf die reale Länge des Modulus zu ergänzen. Ferner enthält der Ini-Brief den jeweiligen Schlüsselnamen.

Als Hashwertverfahren ist derzeit ausschließlich SHA-256 vorgesehen.

Bei der Hashwertbildung ist wie folgt vorzugehen:

a) Padding der höchstwertigen Bits des Exponenten mit Nullen (X’00’) auf die reale Länge des Modulus

b) Konkatenierung von Exponent und Modulus (Exponent || Modulus)

c) Bildung des Hashwerts mittels SHA-256 gemäß Kap. B.2.1.1 über diesen Ausdruck

Nach Erhalt des Ini-Briefs führt der Kunde einen Vergleich des im Ini-Brief aufgeführten Hashwerts mit dem Hashwert des vom Kreditinstitut übermittelten Schlüssels durch.

Bei Übereinstimmung der Hashwerte gelten der bzw. die öffentlichen Schlüssel des Kreditinstituts als authentisiert.

Das Kundenprodukt sollte den Hashwertvergleich für den Kunden in geeigneter Weise unterstützen.

Fall B: Übermittlung des Hashwerts auf der Chipkarte

Auf der Karte befindet sich in der Applikation EF_NOTEPAD (s. Kap. C.1.1) für Fall A der Hashwert des öffentlichen Signierschlüssels des Kreditinstituts und für Fall B der Hashwert des öffentlichen Chiffrierschlüssels des Kreditinstituts. Die Hashwertbildung erfolgt wie in Fall A.

Dieser Hashwert wird vom Kundenprodukt mit dem Hashwert des in der Nachricht übermittelten Schlüssels verglichen.

Das Kundenprodukt sollte den Kunden über das Ergebnis des Hashwertvergleichs informieren.

Bei Übereinstimmung der Hashwerte 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 Kunde 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 Hashwertvergleich wie in den beiden anderen Fällen ist nicht erforderlich.

Das Kundenprodukt sollte den Kunden über das Ergebnis der Zertifikatsprüfung informieren.

♦ Initiale Schlüsselverteilung des Kundensystems

2. Der Kunde übermittelt alle seine öffentlichen Schlüssel, die mit dem privaten Signierschlüssel unterzeichnet wurden, im Rahmen der Key-Management-Nachricht „Erstmalige Übermittlung der Schlüssel des Kunden“ an das Kreditinstitut (vgl. Kapitel B.6.2.3). Diese Nachricht muss sowohl signiert als auch chiffriert sein.
3. Um die Authentizität der Schlüssel zu gewährleisten, sind folgende Mechanismen möglich:

   Fall A: Ini-Brief Kreditinstitut → Kunde

   Der Benutzer erfährt anhand des Rückmeldungscodes 3310 („Ini-Brief erforderlich“) in der Kreditinstitutsnachricht, dass diese Nachricht durch einen Ini-Brief gemäß dem in *Abbildung 5: Beispiel für die Gestaltung des Ini-Briefs bei RAH-9* bzw. RAH-10 aufgeführten Muster begleitet werden muss. Im Ini-Brief bestätigt der Kunde ausschließlich den öffentlichen Signierschlüssel mit handschriftlicher Unterschrift. Eine Bestäti-

gung 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 Hashwert des Schlüssels aufgeführt. Dieser wird ebenso gebildet wie der Hashwert im Ini-Brief des Kreditinstituts (s.o.).

Im Kreditinstitut findet ein Vergleich zwischen dem im Ini-Brief aufgeführten Hashwert und dem Hashwert des vom Kunden übermittelten öffentlichen Signierschlüssels statt. Falls dieser Vergleich positiv verläuft, werden die öffentlichen Schlüssel des Kunden freigeschaltet.

Fall B: Prüfung des übermittelten Zertifikates

Der Kunde erfährt anhand des Rückmeldungscodes 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 Kunde 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.

Im nächsten Schritt werden die öffentlichen Schlüssel des Benutzers freigeschaltet. Ein Hashwertvergleich wie in Fall A ist nicht erforderlich.

4. Bei RAH-10 hat eine Synchronisierung der Kundensystem-ID zu erfolgen (s. [Formals], Kap. III.8).
5. Nachdem die Erstinitialisierung abgeschlossen ist kann der Benutzer Auftragsnachrichten senden.

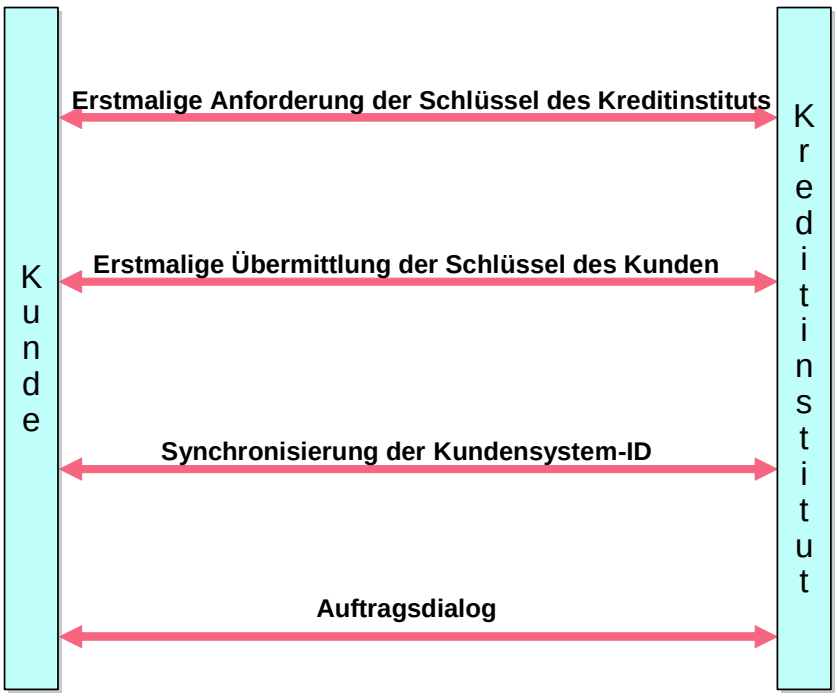


Abbildung 4: Ablauf der Erstinitialisierung bei RAH

Um die Multibankfähigkeit verschiedener Kundensysteme zu sichern, gelten für die Ini-Datei z. B auf einem USB-Stick folgende Namenskonventionen:

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

Falls die Benutzerkennung nicht im Dateisystem darstellbar ist, ist sie entsprechend zu kürzen. Der USB-Stick muss im Standardformat des jeweiligen Betriebssystems formatiert sein. Die Dateien sind im Stammverzeichnis des USB-Stick abzulegen.

Die folgende Abbildung zeigt ein Beispiel für einen Ini-Brief bei Verwendung der Sicherheitsprofile RAH-9 und RAH-10 mit SHA-256 als Hashwertverfahren.

Ini-Brief HBCI

Benutzername	____________	Kundensoftware-interner Name (Angabe freigestellt)
Datum	____________	Datum der Erstellung des Initialisierungsauftrags (TT.MM.JJJJ)
Uhrzeit	____________	Uhrzeit der Erstellung des Initialisierungsauftrags (hh:mm)
Empfänger	____________	Kreditinstitutskennung (wird vom jeweiligen Kreditinstitut mitgeteilt)
Benutzerkennung	____________	max. 30 Stellen alphanumerisch (wird vom jeweiligen Kreditinstitut mitgeteilt)
Schlüsselnummer	____________	Nummer des Signierschlüssels (max. 3 Stellen)
Schlüsselversion	____________	Version des Signierschlüssels (max. 3 Stellen)
HBCI-Version	____________	derzeit 3.0
Sicherheitsprofil	____________	RAH-9 oder RAH-10

Öffentlicher Schlüssel für die elektronische Signatur:

Exponent <Reale Moduluslänge>

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

Modulus ..2048

```
97 A9 D8 FD 68 46 BA 8C 5E DB E0 F5 1C 35 41 41
3C 4A B3 7C 95 6A CC 11 29 E3 A7 63 A6 7B A8 6D
E8 1D 99 D2 74 24 6E 6C C6 FD A9 9C A2 19 15 44
BF 2A 5D B6 93 5B FA FB 88 19 FE B2 1C 96 FB 4F
5E 66 6A 2D F8 80 5E 84 9B 42 8C E6 9A 76 AB 52
5A 36 C1 6D 81 D1 65 0A BC 86 35 86 E7 8E 5B B5
EF E0 84 A1 44 76 69 DE 34 D5 31 A2 D0 25 02 78
B5 BE AB C0 D9 30 B1 04 91 DF 0D 3B E1 33 3D E7
24 57 CD E5 A2 13 EF 95 4B 23 81 58 91 60 85 71
39 3E 00 92 7C F9 06 3A 97 3D 3F 87 52 80 AF 4E
37 39 F1 9E 14 A5 71 30 33 6C E4 DD 38 66 E8 B2
29 6B BF F2 25 36 3E 0F A0 55 39 4A 31 7A 69 88
9B D6 26 B6 B6 53 CA C7 40 4A 60 79 29 C4 97 DC
43 3B 54 A6 B6 44 33 E0 E8 05 17 43 50 13 EE D4
F2 EF F2 BD B2 2A B9 87 52 95 FC 56 93 24 75 A7
E7 D1 CD 90 9A D3 D8 78 99 1D C3 21 4D 2F 5A A3
```

Hashwert (SHA-256):

```
BF C2 A1 01 7C 65 31 9C 8B 1B D8 26 09 85 E6 1F
A9 99 1A 65 17 BF 67 86 17 D8 7C EE DC C3 61 11
```

Ich bestätige hiermit den obigen öffentlichen Schlüssel für meine elektronische Signatur.

____________	________________	________________
Ort / Datum	Firma/Name	Unterschrift

Abbildung 5: Beispiel für die Gestaltung des Ini-Briefs bei RAH-9 bzw. RAH-10

B.3.1.1.3 Schlüsseländerungen

- **Routinemäßige Schlüsseländerung des Kunden**

  Bei Speicherung der Schlüssel auf einer Chipkarte ist i.d.R. auf elektronische Weise keine Änderung einzelner kartenindividueller Schlüssel möglich. Im Falle einer routinemäßigen Schlüsseländerung (z. B. bei Ablauf des Zertifikates) oder einer vermuteten Kompromittierung muss daher ein Kartenaustausch oder ein Ersatz aller Schlüssel erfolgen.

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

  Der Kunde sendet je Kreditinstitut im Rahmen eines HBCI-Dialoges eine Nachricht, in welcher dieses über einen neuen öffentlichen Schlüssel informiert wird (vgl. Kapitel B.6.2.1). Die Nachricht ist mit dem alten (bei Wechsel des Signierschlüssels), respektive dem aktuellen (bei Wechsel des DS-Schlüssels oder des Chiffrierschlüssels) privaten Signierschlüssel des Kunden zu signieren und mit dem aktuellen Chiffrierschlüssel des Kreditinstituts zu chiffrieren.

  Das Kreditinstitut speichert diesen neuen öffentlichen Schlüssel des Kunden und verwendet ihn ab sofort (d. h. bereits in der Antwortnachricht) für alle Verschlüsselungen bzw. Verifikationen von Signaturen. Gleichzeitig kann der alte Schlüssel gesperrt werden. Zusätzlich ist es jedoch bei kartengestützten Verfahren – unabhängig von der Nutzung von Zertifikaten – erlaubt, einen Schlüssel für die Laufzeit der Karte weiter aktiv zu halten und somit zwei Schlüssel parallel zu unterstützen.

  Falls die Übermittlung der neuen Schlüssel aus irgendeinem Grunde fehlschlägt, kann der Kunde 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. Doppeleinreichungskontrolle) wird gelöscht.

- **Routinemäßige Schlüsseländerung des Kreditinstituts**

  Ein Kreditinstitut generiert bei Bedarf ein neues Schlüsselpaar.

  Der Kunde sendet jeweils bei der Dialoginitialisierung die Referenz auf die öffentlichen Schlüssel des Kreditinstitutes mit (vgl. [Formals], Kapitel III.3.1). Falls das Kreditinstitut über aktuellere öffentliche Schlüssel verfügt, werden diese in der Kreditinstitutsnachricht mitübertragen (vgl. [Formals], Kapitel III.3.2 respektive B.6.1.3). Die neuen Schlüssel gelten ab sofort, d. h. bereits für die erste Auftragsnachricht nach der Dialoginitialisierung. 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 Hashwert des öffentlichen Schlüssels analog der initialen Schlüsselverteilung an den Kunden übermitteln. Die Verifikation ist grundsätzlich optional.

  Für den Fall, dass der alte Kreditinstitutsschlüssel nicht mehr zur Verfügung steht oder gesperrt werden musste, wird dem Kunden - falls er den alten Kreditinstitutsschlüssel zur Chiffrierung der Dialoginitialisierung verwendet – der Rückmeldungscode "9030" mit dem Hinweis "Fehler beim Entschlüsseln" gesendet. Ggf. kann die Dialoginitialisierung vom Kreditinstitutssystem auch gar nicht verarbeitet werden, so dass keine Antwort gesendet wird. Daraufhin sollte das Kundenprodukt über den

anonymen Dialog mit Hilfe der Nachricht „Erstmalige Anforderung der Schlüssel des Kreditinstituts“ (s. Kap. B.6.2.1) die neuen Kreditinstitutsschlüssel anfordern. Zur Verifikation der neuen Schlüssel muss dem Kunden in diesem Fall zusätzlich ein Ini-Brief mit dem Hashwert des neuen Kreditinstitutsschlüssels zugeschickt werden.

B.3.1.1.4 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.

B.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 Kunden
- 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. Kapitel B.6.2.4) erfolgen, welche signiert sein muss.

♦ **Verlust des eigenen Schlüssels**

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

Eine nicht-signierungspflichtige Sperrmöglichkeit ist optional, da hierdurch die Gefahr des Mißbrauchs gegeben ist (absichtliche Sperrung fremder Anschlüsse). Der Segmentaufbau erfolgt analog der oben beschriebenen Nachricht, jedoch ist keine Signatur nötig (möglich). Die Steuerung hierfür erfolgt über das Feld „Anzahl benötigter Signaturen“ in der UPD.

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 Kundenschlüssel gesperrt. Sofern die Nachricht lediglich von einem einzigen Benutzer signiert wurde

oder falls bei einer mehrfach signierten Nachricht der Dialogführer von der Fehlsignaturensperre betroffen ist, wird der Dialog beendet. Der Dialogabbruch erfolgt dabei kreditinstitutsseitig im Anschluss an die Antwortnachricht, d.h. ein Austausch von Dialogbeendigungsnachrichten findet nicht statt. Die Antwort ist beim RAH-Verfahren signiert (sofern kreditinstitutsseitig signiert wird) aber nicht verschlüsselt. In der Antwortnachricht teilt das Kreditinstitut lediglich den Grund des Dialogendes mit. Antworten auf Aufträge dürfen nicht mitgesendet werden, da diese aufgrund der Sperrung nicht abgesichert werden können.

- **Information des Benutzers**

  Im Falle einer Sperrung aufgrund von Schlüsselkompromittierung oder Schlüsselverlust erhält der Kunde auf die Sperrnachricht eine Antwortnachricht (vgl. Kapitel B.6.2.4 b), 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 Kunden.

  Ist der Schlüssel kompromittiert oder nicht mehr auffindbar, so wird für den Benutzer eine neue Chipkarte, respektive neue Schlüssel und ein neues EF_ID oder ein neues Schlüsselpaar 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.

B.4 Bankfachliche Anforderungen

- **Zu signierende Nachrichten**

  Grundsätzlich sind alle Kundennachrichten gemäß der in der BPD vorgegebenen Sicherheitsklasse zu signieren. Ausnahmen gelten beim anonymen Zugang, bei der Erstinitialisierung und der Schlüsselsperrung.

  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 Kundensystem-ID 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 kundenseitig 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 kann unterschieden werden, ob die Reihenfolge der Unterzeichnung bedeutungslos oder relevant ist. Diese Unterscheidung muss nicht nur im Kundenprodukt gemacht werden können, sondern hat auch Einfluss auf die Verarbeitung und Kontrolle im Kreditinstitut. In der vorliegenden FinTS-Version 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 Erfasser einer Nachricht, 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“ ([Formals], Kap. IV.4) an.

B.5 Formate für Signatur und Verschlüsselung

Für die Speicherung der Sicherheitsinformationen für die Signatur(en) werden unmittelbar nach dem Nachrichtenkopf das (die) Segment(e) „Signaturkopf“ (HNSHK) und unmittelbar vor dem Nachrichtenabschluss das (die) Segment(e) „Signaturabschluss“ (HNSHA) in die bestehende Nachricht eingeschoben.

Dies entspricht dem in UN/EDIFACT definierten Vorgehen und kann folgendermaßen visualisiert werden:

HNHBK	HNSHK	HBCI-Nutzdaten	HNSHA	HNHBS

(Die grau hinterlegten Bereiche gehen in die Signatur mit ein.)

Falls mehrere Signaturen für HBCI-Nachrichten erforderlich sind, so wiederholen sich Signaturkopf und -abschluss entsprechend:

HNHBK	$HNSHK_2$	$HNSHK_1$	HBCI-Nutzdaten	$HNSHA_1$	$HNSHA_2$	HNHBS

(Die grau hinterlegten Bereiche bezeichnen die Daten für die Zweit-Signatur bei beliebiger Reihenfolge der Signaturen (vgl. Kapitel B.4)).

Bei der Verschlüsselung wird nach dem Nachrichtenkopf ein Verschlüsselungskopf-Segment (HNVSK) eingefügt. Dies bedeutet, dass alle Daten nach dem Segmentendekennzeichen des Nachrichtenkopfes bis zum letzten Byte vor dem Nachrichtenabschluss inklusive aller Signaturen in die Verschlüsselung eingehen:

HNHBK	HNVSK	e_k(HNSHK$_n$ | HBCI-Nutzdaten | HNSHA$_n$)	HNHBS

Grundsätzlich erfolgt die Reihenfolge der Sicherheitsverarbeitung in folgender Reihenfolge:

1. elektronische Signatur
2. evtl. Zweit- und Drittsignatur
3. (Komprimierung) und Verschlüsselung

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

B.5.1 Signaturkopf

♦ **Beschreibung**

Der Signaturkopf enthält Informationen über den damit verbundenen Sicherheitsservice, sowie über den Absender.

♦ **Format**

Name: Signaturkopf
Typ: Segment
Segmentart: Administration
Kennung: HNSHK
Bezugssegment: -
Version: 4
Sender: Kunde/Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Sicherheitsprofil	DEG			M	1	
3	Sicherheitsfunktion, kodiert	DE	code	..3	M	1	1, 2
4	Sicherheitskontrollreferenz	DE	an	..14	M	1	<>0
5	Bereich der Sicherheitsapplikation, kodiert	DE	code	..3	M	1	1
6	Rolle des Sicherheitslieferanten, kodiert	DE	code	..3	M	1	1, 3, 4
7	Sicherheitsidentifikation, Details	DEG			M	1	
8	Sicherheitsreferenznummer	DE	num	..16	M	1	
9	Sicherheitsdatum und -uhrzeit	DEG			M	1	
10	Hashalgorithmus	DEG			M	1	
11	Signaturalgorithmus	DEG			M	1	
12	Schlüsselname	DEG			M	1	
13	Zertifikat	DEG			C	1	M: bei RAH-7 in Verbindung mit mindestens einem zu signierenden Geschäftsvorfall, der Sicherheitsklasse 2, 3 oder 4 erfordert. O: bei RAH-9 in Verbindung mit zu signierenden Geschäftsvorfällen, die Sicherheitsklasse 1 bis 2 erfordern N: bei RAH-10

♦ **Belegungsrichtlinien**

Sicherheitsfunktion, kodiert

Abhängig von Sicherheitsprofil und Schlüsseltyp und HBCI-Version ist folgender Wert einzustellen:

Sicherheitsprofil	Schlüsseltyp	ab FinTS V3.0
RAH-7	S	2
RAH-7	D	1
RAH-9	S	2
RAH-10	S	2

Weitere Erläuterungen sind im Data Dictionary zu finden.

Bereich der Sicherheitsapplikation, kodiert

Der einzig zugelassene Wert ist "1", d.h. SHM (nur Signaturkopf und HBCI-Nutzdaten).

Rolle des Sicherheitslieferanten, kodiert

Der Inhalt dieses Feldes sollte derzeit nicht ausgewertet werden. Optional können aber die nachfolgenden Festlegungen angewendet werden, sofern dies zwischen Kunde und Kreditinstitut zuvor vereinbart wurde:

1. **Dialoginitialisierung und -ende:**

   Die Rolle wird durch den Dialogführenden bestimmt. Es ist nur eine Signatur erlaubt. Erlaubt ist nur der Wert ISS/wert1[11].

2. **Auftragsnachricht:**

   Grundsätzlich gilt: Sobald die Rolle „WIT“ verwendet wird, muss dieser Benutzer mit der Benutzerkennung aus der Dialoginitialisierung arbeiten. Auch der Benutzer „WIT“ muss bankseitig entsprechend der Auftragsart am Konto des Benutzers „ISS“ berechtigt sein.

   Die Reihenfolge der Signaturen ist beliebig.

Anzahl Signaturen	Erlaubte Kombinationen		
	1. Signatur	2. Signatur	3. Signatur
1	ISS/wert1	-	-
2	ISS/wert1	CON/beliebig	-
	WIT/wert1	ISS/beliebig	-
3	WIT/wert1	ISS/beliebig	CON/beliebig

Auch bei Belegung dieses Feldes kann das Kundenprodukt nicht davon ausgehen, dass das Feld kreditinstitutsseitig ausgewertet wird.

Sicherheitsidentifikation, Details

Wenn eine Synchronisierung der Kundensystem-ID durchgeführt wird, ist als Identifizierung der Partei ‚0’ einzustellen.

[11] Die Notation gibt die Rolle gefolgt von der Benutzerkennung an.

Sicherheitsdatum und -uhrzeit

Als Bezeichner wird „1“ eingestellt, da es sich um einen Sicherheitszeitstempel handelt.

Zertifikat

Im Falle der Bankensignaturkarte ist je nach Signaturanforderung der Geschäftsvorfälle entweder das Zertifikat C_X509.CH.DS oder das Zertifikat $C\_X509.CH.AUT_{C/S}$[&KE] anzugeben.

B.5.2 Signaturabschluss

- **Beschreibung**

  Der Signaturabschluss stellt die Verbindung mit dem dazugehörigen Signaturkopf her und enthält als "Validierungsresultat" die elektronische Signatur.

- **Format**

  Name: Signaturabschluss
  Typ: Segment
  Segmentart: Administration
  Kennung: HNSHA
  Bezugssegment: -
  Version: 2
  Sender: Kunde/Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Sicherheitskontrollreferenz	DE	an	..14	M	1	<>0
3	Validierungsresultat	DE	bin	..512	C	1	M: bei HBCI N: bei PINTAN
4	Benutzerdefinierte Signatur	DEG			C	1	N: bei HBCI M/N/O bei anderen Verfahren

B.5.3 Verschlüsselungskopf

♦ Beschreibung

Der Verschlüsselungskopf enthält Informationen über die Art des Sicherheitsservice, die Verschlüsselungsfunktion und die zu verwendenden Chiffrierschlüssel.

Zum Abgleich mit dem in den BPD definierten RAH-Verschlüsselungsverfahren wird das Feld „Bezeichner für Algorithmusparameter, Schlüssel“ in der DEG „Verschlüsselungsalgorithmus“ herangezogen.

♦ Format

Name: Verschlüsselungskopf
Typ: Segment
Segmentart: Administration
Kennung: HNVSK
Bezugssegment: -
Version: 3
Sender: Kunde/Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Sicherheitsprofil	DEG			M	1	
3	Sicherheitsfunktion, kodiert	DE	code	..3	M	1	4
4	Rolle des Sicherheitslieferanten, kodiert	DE	code	..3	M	1	1, 4
5	Sicherheitsidentifikation, Details	DEG			M	1	
6	Sicherheitsdatum und -uhrzeit	DEG			M	1	
7	Verschlüsselungsalgorithmus	DEG			M	1	
8	Schlüsselname	DEG			M	1	
9	Komprimierungsfunktion	DE	code	..3	M	1	
10	Zertifikat	DEG			C	1	O: kreditinstitutsseitig bei RAH-7 und RAH-9 (vgl.B.3.1.1.1) bzw. kundenseitig bei RAH-7 N: sonst

♦ Belegungsrichtlinien

Sicherheitsdatum und -uhrzeit

Als Bezeichner (DE Datum- und Zeitbezeichner, kodiert) wird „1“ (Sicherheitszeitstempel) eingestellt.

Zertifikat

Im Falle der Bankensignaturkarte ist das Zertifikat EF_C_X509.CH.KE anzugeben.

B.5.4 Verschlüsselte Daten

- **Beschreibung**

  Dieses Segment enthält die verschlüsselten (und komprimierten) Daten.

- **Format**

  Name: Verschlüsselte Daten
  Typ: Segment
  Segmentart: Administration
  Kennung: HNVSD
  Bezugssegment: -
  Version: 1
  Sender: Kunde/Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Daten, verschlüsselt	DE	bin	..	M	1	

B.6 Key-Management

B.6.1 Formate für Key-Management

Für die Schlüsseländerung, die Schlüsselverteilung sowie die Schlüsselsperrung sind die nachfolgenden Segmente vorgesehen. Diese dürfen nur im Rahmen der speziellen Key-Management-Nachrichten verwendet werden.

B.6.1.1 Änderung eines öffentlichen Schlüssels

- **Beschreibung**

  Dieses Segment enthält einen neuen öffentlichen Schlüssel des Kunden.

- **Format**

  Name: Schlüsseländerung
  Typ: Segment
  Segmentart: Administration
  Kennung: HKSAK
  Bezugssegment: -
  Version: 3
  Sender: Kunde

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Nachrichtenbeziehung, kodiert	DE	code	1	M	1	2
3	Bezeichner für Funktionstyp	DE	code	..3	M	1	112
4	Sicherheitsprofil	DEG			M	1	
5	Schlüsselname	DEG			M	1	
6	Öffentlicher Schlüssel	DEG			M	1	
7	Zertifikat	DEG			O	1	

- **Belegungsrichtlinien**

Nachrichtenbeziehung, kodiert

Im Zusammenhang mit der Schlüsseländerung ist immer folgender Wert vorgesehen: "2" (Key-Management-Nachricht erwartet Antwort)

Bezeichner für Funktionstyp

Im Zusammenhang mit der Schlüsseländerung ist folgender Wert vorgesehen: "112" (Certificate Replacement)

Sicherheitsprofil

Es wird das den Schlüsseln entsprechende Sicherheitsprofil eingestellt.

Schlüsselname

Es ist der Name des neuen öffentlichen Schlüssels des Kunden einzustellen.

Zertifikat

Falls für den neuen öffentlichen Schlüssel ein Zertifikat verfügbar ist, kann es dem Kreditinstitut auf diese Weise eingereicht werden.

B.6.1.2 Anforderung eines öffentlichen Schlüssels

♦ **Beschreibung**

Dieses Segment enthält die Anfrage nach einem öffentlichen Schlüssel des Kreditinstituts. Im Feld „Sicherheitsprofil“ gibt der Kunde an, für welches Profil er die Schlüssel anfordert. Das Segment wird entweder innerhalb der Dialoginitialisierung (vgl. [Formals], Kapitel III.3.1) oder im Rahmen der erstmaligen Schlüsselanforderung (vgl. Kapitel B.6.2.1) gesendet.

♦ **Format**

Name: Anforderung eines öffentlichen Schlüssels
Typ: Segment
Segmentart: Administration
Kennung: HKISA
Bezugssegment: -
Version: 3
Sender: Kunde

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Nachrichtenbeziehung, kodiert	DE	code	1	M	1	2
3	Bezeichner für Funktionstyp	DE	code	..3	M	1	124
4	Sicherheitsprofil	DEG			M	1	
5	Schlüsselname	DEG			M	1	
6	Zertifikat	DEG			O	1	

♦ **Belegungsrichtlinien**

Nachrichtenbeziehung, kodiert

Im Zusammenhang mit der Anfrage nach einem öffentlichen Schlüssel ist immer folgender Wert vorgesehen: "2" (Key-Management-Nachricht erwartet Antwort)

Bezeichner für Funktionstyp

Im Zusammenhang mit der Anfrage für einen öffentlichen Schlüssel ist folgender Wert vorgesehen: "124" (Certificate Status Request)

Schlüsselname

In den Schlüsselnamen ist die Schlüsselnummer und -version des Schlüssels einzustellen, den das Kundenprodukt als aktuellen öffentlichen Schlüssel des Kreditinstituts kennt. Falls dieser noch nicht vorliegt, ist in beide Felder der Wert „999“ einzustellen.

B.6.1.3 Übermittlung eines öffentlichen Schlüssels

♦ **Beschreibung**

Dieses Segment wird zum einen innerhalb der Dialoginitialisierungsantwort (vgl. [Formals], Kapitel III.3.2) an den Kunden übertragen, falls sich der öffentliche Schlüssel des Kreditinstituts geändert hat. Es enthält dann jeweils einen öffentlichen Schlüssel des Kreditinstituts.

Zum anderen wird das Segment im Rahmen der erstmaligen Anforderung der öffentlichen Schlüssel des Kreditinstituts (vgl. Kapitel B.6.2.1) benötigt.

♦ **Format**

Name: Übermittlung eines öffentlichen Schlüssels
Typ: Segment
Segmentart: Administration
Kennung: HIISA
Bezugssegment: HKISA
Version: 3
Sender: Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Nachrichtenbeziehung, kodiert	DE	code	1	M	1	1
3	Austauschkontrollreferenz	DE	id	#	M	1	
4	Nachrichtenreferenz-nummer	DE	num	..4	M	1	>0
5	Bezeichner für Funktions-typ	DE	code	..3	M	1	224
6	Schlüsselname	DEG			M	1	
7	Öffentlicher Schlüssel	DEG			M	1	
8	Zertifikat	DEG			O	1	

♦ **Belegungsrichtlinien**

Nachrichtenbeziehung, kodiert

Es ist folgender Wert vorgesehen: "1" (Key-Management-Nachricht ist Antwort)

Austauschkontrollreferenz

Dialog-ID der Anfragenachricht des Kunden nach einem öffentlichen Schlüssel (vgl. [Formals], Kapitel II.6.2).

Wird das Segment HIISA in einer Schlüsseldatei auf einem Medium abgelegt, so kann dieses Feld mit dem Wert "0" belegt werden.

Nachrichtenreferenznummer

Nachrichtennummer der Anfragenachricht des Kunden nach einem öffentlichen Schlüssel (vgl. [Formals], Kapitel II.6.2).

Wird das Segment HIISA in einer Schlüsseldatei auf einem Medium abgelegt, so kann dieses Feld mit einem beliebigen gültigen Wert belegt werden.

Bezeichner für Funktionstyp

Es ist folgender Wert vorgesehen: "224" (Certificate Status Notice)

Schlüsselname

Der zurückgemeldete Schlüsselname enthält insbesondere die zugehörige Schlüssel- und Versionsnummer, die das Kundenprodukt für die Referenzierung des in der DEG „Öffentlicher Schlüssel“ übertragenen neuen öffentlichen Schlüssels verwendet.

Öffentlicher Schlüssel

Diese Datenelementgruppe enthält den neuen öffentlichen Schlüssel des Kreditinstitutes.

B.6.1.4 Schlüsselsperrung

- **Beschreibung**

  Dieses Segment enthält die Anforderung für das Sperren eines Schlüssels.

- **Format**

  Name: Schlüsselsperrung
  Typ: Segment
  Segmentart: Administration
  Kennung: HKSSP
  Bezugssegment: -
  Version: 3
  Sender: Kunde

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Nachrichtenbeziehung, kodiert	DE	code	1	M	1	2
3	Bezeichner für Funktionstyp	DE	code	..3	M	1	130
4	Sicherheitsprofil	DEG			M	1	
5	Schlüsselname	DEG			M	1	
6	Sperrenkennzeichen	DE	code	..3	M	1	1, 501, 999
7	Sicherheitsdatum und -uhrzeit	DEG			O	1	
8	Zertifikat	DEG			O	1	

- **Belegungsrichtlinien**

Nachrichtenbeziehung, kodiert

Im Zusammenhang mit der Schlüsselsperrung ist folgender Wert vorgesehen: "2" (Key-Management-Nachricht erwartet Antwort)

Bezeichner für Funktionstyp

Im Zusammenhang mit der Schlüsselsperrung ist folgender Wert vorgesehen: "130" (Certificate Revocation)

Sicherheitsprofil

Es wird das den Schlüsseln entsprechende Sicherheitsprofil eingestellt.

Schlüsselname

Es sind die Identifikationsmerkmale des zu sperrenden Signierschlüssels einzustellen, unabhängig davon, dass grundsätzlich immer sowohl Signier- als auch Chiffrierschlüssel gesperrt werden (s. Kap. B.6.2.4).

Sicherheitsdatum und -uhrzeit

Enthält optional Datum und Uhrzeit, ab welcher der Schlüssel nicht mehr gültig ist. Als Bedeutung wird „6“ (für CRT, Certificate Revocation Time) eingestellt.

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

B.6.1.5 Bestätigung der Schlüsselsperrung

♦ **Beschreibung**

Dieses Segment enthält die Bestätigung für eine Schlüsselsperrung.

♦ **Format**

Name: Bestätigung der Schlüsselsperrung
Typ: Segment
Segmentart: Administration
Kennung: HISSP
Bezugssegment: HKSSP
Version: 3
Sender: Kreditinstitut

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkopf	DEG			M	1	
2	Nachrichtenbeziehung, kodiert	DE	code	1	M	1	1
3	Austauschkontrollreferenz	DE	id	#	M	1	
4	Nachrichtenreferenz-nummer	DE	num	..4	M	1	>0
5	Bezeichner für Funktions-typ	DE	code	..3	M	1	231
6	Schlüsselname	DEG			M	1	
7	Sperrenkennzeichen	DE	code	..3	M	1	1, 501, 999
8	Sicherheitsdatum und -uhrzeit	DEG			M	1	
9	Zertifikat	DEG			O	1	

♦ **Belegungsrichtlinien**

Nachrichtenbeziehung, kodiert

Im Zusammenhang mit der Bestätigung der Schlüsselsperrung ist folgender Wert vorgesehen: "1" (Key-Management-Nachricht ist Antwort)

Austauschkontrollreferenz

Dialog-ID der Sperranforderung des Kunden (vgl. [Formals], Kapitel II.6.2).

Nachrichtenreferenznummer

Nachrichtennummer der Sperrenanforderung des Kunden (vgl. [Formals], Kapitel II.6.2).

Bezeichner für Funktionstyp

Im Zusammenhang mit der Bestätigung der Schlüsselsperrung ist folgender Wert vorgesehen: "231" (Revocation Confirmation)

Schlüsselname

Es sind die Identifikationsmerkmale des gesperrten Signierschlüssels einzustellen, unabhängig davon, dass grundsätzlich immer sowohl Signier- als auch Chiffrierschlüssel gesperrt werden (s. Kap. B.6.2.4).

Sicherheitsdatum und -uhrzeit

Enthält optional Datum und Uhrzeit, ab welcher der Schlüssel nicht mehr gültig ist. Als Bedeutung wird „6“ (für CRT, Certificate Revocation Time) eingestellt.

B.6.2 Key-Management-Nachrichten

Aufträge des Key-Managements dürfen nur in den folgenden separaten Nachrichten übertragen werden.

Hiervon abweichend wird der Auftrag „Anforderung eines öffentlichen Schlüssels des Kreditinstituts“ nicht als eigene Nachricht, sondern innerhalb der Dialoginitialisierung übertragen.

Die Nachrichten für das Key-Management müssen zum Teil kryptographisch geschützt werden. Alternativ können auch Offline-Sicherungsverfahren (z. B. Brief) zum Einsatz kommen (vgl. Kapitel B.3.1.1.3).

Es sind folgende Key-Management-Nachrichten vorgesehen:

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

B.6.2.1 Änderung eines öffentlichen Schlüssels des Kunden

Realisierung Bank: verpflichtend
Realisierung Kunde: verpflichtend

a) Kundennachricht

♦ **Beschreibung**

Der Nachricht muss eine Dialoginitialisierung vorausgehen. Der Auftrag muss mit dem alten Signierschlüssel signiert werden.

Es muss unterschieden werden, ob die Schlüsseländerung auch das Sicherheitsprofil wechselt oder nicht.

Die folgenden Wechselmöglichkeiten bestehen, falls Sicherheitsprofilwechsel unterstützt sind:

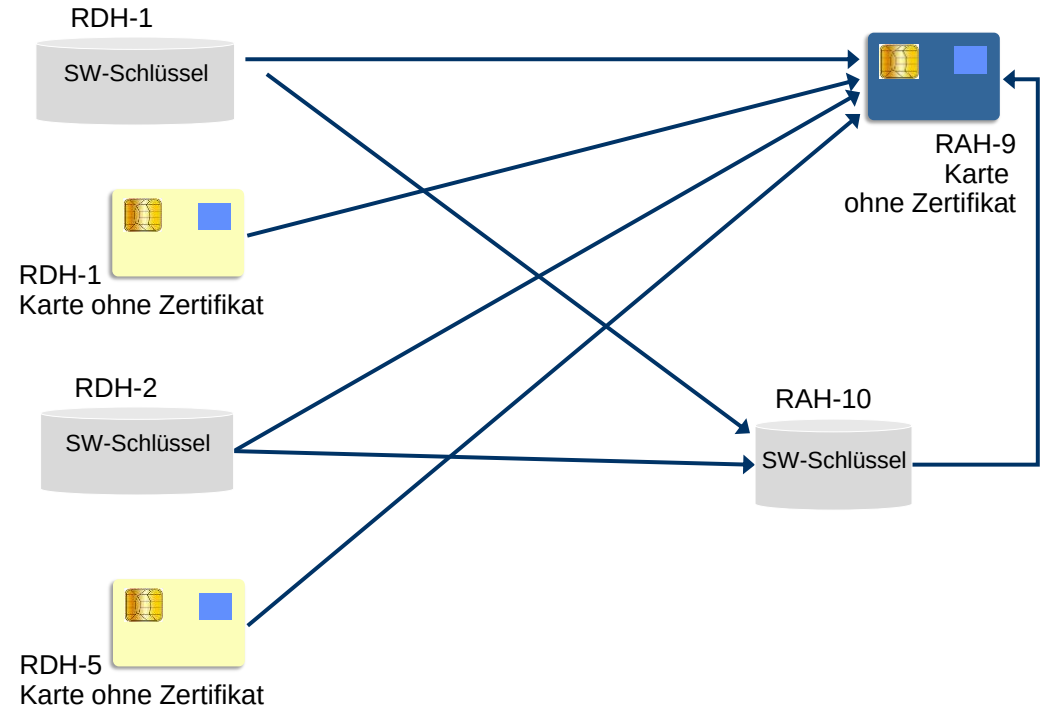


Abbildung 6: Unterstützte Sicherheitsprofilwechsel RDH-1, RDH-2 und RDH-5 auf RAH-9 und RAH-10

Zusammengefasst ergeben sich folgende Wechselmöglichkeiten:

1. ohne Wechsel des Sicherheitsprofils:

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

2. mit Wechsel des Sicherheitsprofils
   (vgl. Abbildung 6):

Bei einem Sicherheitsprofilwechsel muss der Kunde immer beide HKSAK-Segmente einstellen. Nach der erfolgreichen Durchführung der Schlüsseländerung wird durch das Kreditinstitut mitgeteilt, ob der vorher aktuelle RAH-x bzw. RDH-x-Schlüssel automatisch gesperrt wurde. Diese Nachricht wird mit den RAH-x bzw. RDH-x-Schlüsseln abgesichert. Wurden die RAH-x bzw. RDH-x-Schlüssel institutsseitig nicht gesperrt, wird der Dialog unter Absicherung der RAH-x bzw. RDH-x-Schlüssel beendet. Es ist darauf zu achten, dass die Nummer der RDH-2-Schlüssel 2 ist, die Version kann mit 1 beginnen. Ab RDH-5 und bei RAH-x sind Schlüsselnummer und –version vorgegeben.

Falls das Kreditinstitut nicht in der Lage ist, zwei Schlüsselpaare zu einem Kunden gleichzeitig zu halten und somit die Endenachricht mit den RAH-x bzw. RDH-x-Schlüsseln nicht mehr bedienen kann, ist dies dem Kundenprodukt durch den Rückmeldungscode 3250 mitzuteilen. Das Kundenprodukt soll dann keine Endenachricht mehr senden und den Bankdatensatz von der RAH-x bzw. RDH-x-Schlüsseldatei löschen.

Es empfiehlt sich, die RAH-x bzw. RDH-x-Schlüssel nach einem erfolgreichen Abschluss des Dialoges durch einen Sperrdialog ungültig zu machen.

Falls der Kunde eine Schlüsseländerungsnachricht sendet, diese aber aus kreditinstitutsinternen Verarbeitungsgründen nicht beantwortet wird, sollte das Kundenprodukt zunächst einen neuen Dialog auf Basis eines der Schlüsselpaare aufbauen. Falls diese Nachricht abgelehnt wird ist ein erneuter Versuch auf Basis eines anderen Schlüsselpaares vorzunehmen. Aus der Reaktion des Kreditinstituts ist für das Kundenprodukt ersichtlich, ob die Schlüsseländerung erfolgreich war oder wiederholt werden muss.
Da es nicht möglich ist, einen DS-Schlüssel, der ja eine natürliche Person identifiziert, über die HBCI-Schlüsseländerung zu ändern, dürften nur "1..2" HKSAK-Segmente eingestellt werden.

Wechsel des Sicherheitsprofils ohne Schlüsselwechsel

Diese Situation tritt bei der Migration von RDH- nach gleichrangigen RAH-Verfahren auf. Beim Übergang von gleichartigen Sicherheitsprofilen (z. B. RDH-9 auf RAH-9 oder RDH-10 auf RAH-10) muss zwar eine erneute Übermittlung der bestehenden öffentlichen Schlüssel durch entsprechende HKSAK-Segmente erfolgen, diese dienen jedoch nur dazu, die Änderung des Sicherheitsprofils bzgl. des Verschlüsselungsalgorithmus (RDH: 2-Key-Triple-DES nach RAH: AES-256) mitzuteilen. Die Schlüsselpaare selbst bleiben unverändert, d. h. weder im Kundenprodukt noch im Kreditinstitut werden Änderungen an den bestehenden Schlüsseln vorgenommen.

Beim Übergang von RDH- auf RAH-Verfahren ergeben sich folgende Möglichkeiten des Schlüsselwechsels (RDH-10 auf RAH-9) bzw. des Wechsel des Verschlüsselungsverfahrens von R**D**H auf R**A**H:

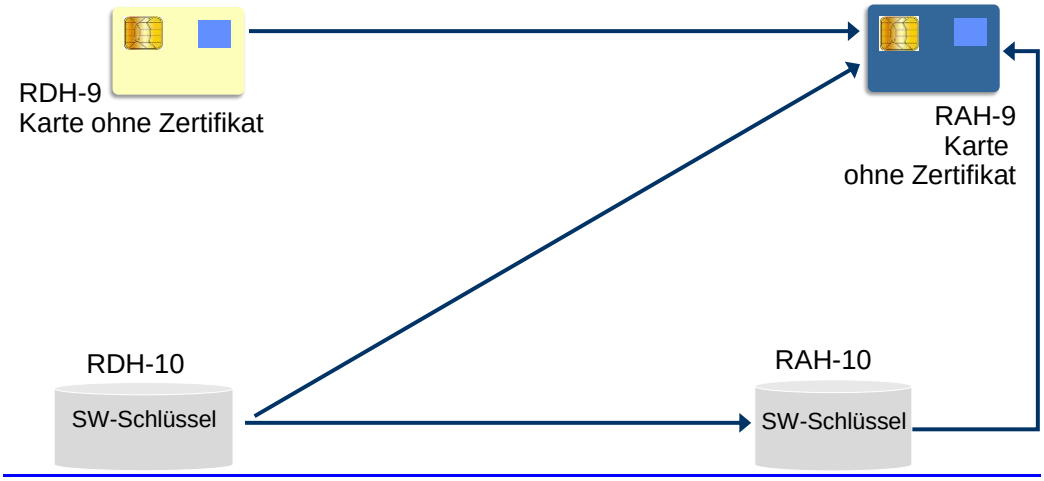


Abbildung 7: Unterstützte Sicherheitsprofilwechsel beim Übergang von RDH- auf RAH-Verfahren

Zum Verfahren s. Kap. B.3.1.1.3.

- **Format**

Name: Änderung eines öffentlichen Schlüssels des Kunden
Typ: Nachricht
Version: 4
Sender: Kunde

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Signaturkopf	SEG	HNSHK	M	1	
3	Schlüsseländerung	SEG	HKSAK	M	1..3	
4	Signaturabschluss	SEG	HNSHA	M	1	
5	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals], Kap. II.5.2

- **Belegungsrichtlinien**

Der Kunde stellt entweder seinen neuen öffentlichen Signierschlüssel, seinen neuen öffentlichen Chiffrierschlüssel oder beide Schlüssel ein.

b) Kreditinstitutsnachricht

- **Format**

Name: Kreditinstitutsnachricht allgemein
Typ: Nachricht
Format: s. [Formals], Kap. II.8.1

- **Erläuterungen**

Es werden keine Datensegmente zurückgemeldet.

♦ Ausgewählte Beispiele für Rückmeldungscodes

Code	Beispiel
0020	Öffentlicher Schlüssel wurde geändert
3250	RDH-1-Schlüssel wurden gesperrt. Endenachricht nicht mehr möglich.
3260	RDH-1-Schlüssel weiterhin gültig. Schlüsselsperre wird empfohlen.
9210	Schlüsseländerung von RDH-1 auf RDH-2 zur Zeit nicht möglich
9010	Schlüsseländerung zur Zeit nicht möglich
9010	Sicherheitsverfahren unterstützt keine öffentlichen Schlüssel
9210	Eingereichter Schlüssel ist mit dem aktuellen Schlüssel identisch

B.6.2.2 Erstmalige Anforderung der Schlüssel des Kreditinstituts

Mit Hilfe dieser Nachricht fordert der Kunde erstmalig den öffentlichen Signier- und Chiffrierschlüssel des Kreditinstituts an. Gleichzeitig erhält er die aktuellen Bankparameterdaten, die er benötigt, um die unterstützten Verschlüsselungsverfahren des Kreditinstituts in Erfahrung zu bringen. Mit Hilfe dieser Informationen wird der Kunde in die Lage versetzt, beliebige Nachrichten zu verschlüsseln.

Realisierung Bank: optional
Realisierung Kunde: verpflichtend

a) Kundennachricht

- **Beschreibung**

Diese Nachricht wird an Stelle einer Dialoginitialisierung gesendet. Es dürfen keine Auftragsnachrichten folgen. Der Dialog ist vom Kunden nach Erhalt der Antwortnachricht mit einer Dialogendenachricht zu beendigen. Die Nachricht wird weder signiert noch verschlüsselt.

- **Format**

Name: Erstmalige Anforderung der Schlüssel des Kreditinstituts
Typ: Nachricht
Version: 4
Sender: Kunde

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Identifikation	SEG	HKIDN	M	1	s. [Formals], Kap. III.3.1.2
3	Verarbeitungsvorbereitung	SEG	HKVVB	M	1	s. [Formals], Kap. III.3.1.3
4	Anforderung eines öffentlichen Schlüssels	SEG	HKISA	M	3	
5	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals],Kap. II.5.2

- **Belegungsrichtlinien**

Identifikation

Die Datenelemente des Segments sind wie beim anonymen Zugang zu belegen (s. [Formals], Kap. III.5).

Verarbeitungsvorbereitung

Mit diesem Segment fordert der Kunde die Bankparameterdaten an.

Anforderung eines öffentlichen Schlüssels

Mit diesen Segmenten fordert der Kunde jeweils den öffentlichen Signierschlüssel und den öffentlichen Chiffrierschlüssel des Kreditinstituts an. Es sind stets alle Schlüssel eines Sicherheitsprofils anzufordern, auch wenn das Kreditinstitut nicht signiert.

In die DEG „Schlüsselname“ ist für die Benutzerkennung der Standardwert ‘999’ einzustellen. In der Rückmeldung wird dem Kunden die korrekte Benutzerkennung des Kreditinstituts mitgeteilt.

Da bei der Erstinitialisierung noch keine BPD vorliegt, ist es für das Kundenprodukt evtl. problematisch, zu ermitteln welche Sicherheitsprofile das Kreditinstitut anbietet und - wenn mehrere möglich sind - welches Profil für den Kunden gilt. Falls dem Kunden diese Information nicht von seinem Kreditinstitut mitgeteilt wurde, sollte das Kundenprodukt versuchen, das Sicherheitsmedium zu lesen und daraus das richtige Sicherheitsprofil zu erschließen.
Da ein Kreditinstitut über keinen D-Schlüssel verfügt bzw. verfügen kann (Voraussetzung ist eine "natürliche Person"), dürfen nur zwei HKISA-Segmente eingestellt werden.

b) Kreditinstitutsnachricht

- **Format**

Name: Erstmalige Übermittlung der Schlüssel des Kreditinstituts
Typ: Nachricht
Version: 4
Sender: Kreditinstitut

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Signaturkopf	SEG	HNSHK	O	1	
3	Rückmeldungen zur Gesamtnachricht	SEG	HIRMG	M	1	s. [Formals], Kap. II.7.2
4	Rückmeldungen zu Segmenten	SEG	HIRMS	O	n	s. [Formals], Kap. II.7.3
5	Bankparameterdaten	SF	#	O	1	s. [Formals], Kap. III.3.2.2
6	Übermittlung eines öffentlichen Schlüssels	SEG	HIISA	M	1..3	
7	Kreditinstitutsmeldung	SEG	HIKIM	O	n	s. [Formals], Kap. III.3.2.5
8	Signaturabschluss	SEG	HNSHA	O	1	
9	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals], Kap. II.5.2

- **Belegungsrichtlinien**

Signaturkopf

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.).

Übermittlung eines öffentlichen Schlüssels

In diesen Segmenten werden dem Kunden die öffentlichen Schlüssel des Kreditinstituts mitgeteilt.

Falls das Kreditinstitut seine Nachrichten nicht signiert, erhält der Kunde 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 Kunden mit dem Hashwert des Chiffrierschlüssels begleitet werden (s. Kap. B.3.1.1.2).

Falls das Kreditinstitut seine Nachrichten signiert, erhält der Kunde sowohl den öffentlichen Chiffrier- als auch 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 Hashwert des Signierschlüssels begleitet werden.

- **Ausgewählte Beispiele für Rückmeldungscodes**

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

B.6.2.3 Erstmalige Übermittlung der Schlüssel des Kunden

Mit Hilfe dieser Nachricht übermittelt der Kunde erstmalig seinen öffentlichen Signier- und Chiffrierschlüssel 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 Kunden signiert sein.

Das Kreditinstitut darf eine Nachricht nicht ablehnen, nur weil für den Kunden 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 Kunden 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 Kunden befinden muss. Ferner muss dem Kunden das Verschlüsselungsverfahren bekannt sein, das ihm in den Bankparameterdaten mitgeteilt wird. Um dem Kunden diese Daten vorab zukommen zu lassen bieten sich folgende Lösungen an:

- Das Kreditinstitut sendet dem Kunden eine Schlüsseldatei zu, die die Schlüssel und die aktuelle BPD enthält, wie in VI.3.1.3.2 beschrieben.
- Der Kunde sendet die Key-Management-Nachricht „Erstmalige Anforderung der Schlüssel des Kreditinstituts“ (s. Kap. B.6.2.1). 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 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.

Auf der Chipkarte können Kommunikationszugänge abgelegt werden (s. Kap. C). Da pro Institut jedoch mehrere Kommunikationszugänge gespeichert sein können (z. B. TCP/IP und HTTPS), muss ein Kundenprodukt zunächst prüfen, ob für dieses Institut bereits die Schlüssel eingereicht wurden, bevor eine erstmalige Übermittlung der Schlüssel des Kunden durchgeführt wird. Für den Fall, dass das Kundenprodukt die Schlüssel dennoch sendet, sollte das Institut die Warnung 3330 „Schlüssel liegen bereits vor“ zurückmelden.

Realisierung Bank: verpflichtend
Realisierung Kunde: verpflichtend

a) Kundennachricht

- **Beschreibung**

Diese Nachricht wird an Stelle einer Dialoginitialisierung gesendet. Es dürfen keine Auftragsnachrichten folgen. Die Nachricht muss signiert und verschlüsselt werden. Der Dialog ist vom Kunden nach Erhalt der Antwortnachricht mit einer Dialogendenachricht zu beendigen. Die Dialogendenachricht ist nicht zu signieren, da der übermittelte Kundenschlüssel zu diesem Zeitpunkt i.d.R. noch nicht freigeschaltet ist.

- **Format**

Name: Erstmalige Übermittlung der Schlüssel des Kunden
Typ: Nachricht
Version: 4
Sender: Kunde

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Signaturkopf	SEG	HNSHK	M	1	
3	Identifikation	SEG	HKIDN	M	1	s. [Formals], Kap. III.3.1.2
4	Schlüsseländerung	SEG	HKSAK	M	2-3	
5	Signaturabschluss	SEG	HNSHA	M	1	
6	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals], Kap. II.5.2

- **Belegungsrichtlinien**

Identifikation

Der Benutzer hat die ihm zur Initialisierung mitgeteilten Daten einzustellen. Wenn die Erstinitialisierung mit der alten Benutzerkennung durchgeführt wird, ist – sofern noch vorhanden – die alte Kundensystem-ID anzugeben, andernfalls ist als Kundensystem-ID der Wert ‚0' anzugeben. Falls zu diesem Zeitpunkt noch keine Synchronisierung durchgeführt wurde, ist als Kundensystem-ID der Wert '0' einzustellen.

Schlüsseländerung

Der Kunde stellt seine öffentlichen Schlüssel ein. Dies können Signier-, Chiffrier- oder Authentikationsschlüssel 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 Hashwert des Signierschlüssels begleitet werden (s. Kap. B.3.1.1.2).

b) Kreditinstitutsnachricht

- **Beschreibung**

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

- **Format**

Name:	Kreditinstitutsnachricht allgemein
Typ:	Nachricht
Format:	s. [Formals], Kap. II.8.1

- **Erläuterungen**

Es werden keine Datensegmente zurückgemeldet.

- **Ausgewählte Beispiele für Rückmeldungscodes**

Code	Beispiel
0010	Öffentlicher Schlüssel wurde entgegengenommen
0020	Öffentlicher Schlüssel wurde freigeschaltet
0020	Kunde wurde freigeschaltet
9010	Auftrag abgelehnt

B.6.2.4 Schlüsselsperrung durch den Kunden

Diese Nachricht beschreibt die Anforderung zum Sperren der Schlüssel durch den Kunden und die Bestätigung der Schlüsselsperrung durch das Kreditinstitut (vgl. Kapitel B.3.2).

Realisierung Bank: verpflichtend
Realisierung Kunde: verpflichtend

a) Kundennachricht

- **Beschreibung**

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

Der Nachricht muss eine Dialoginitialisierung vorausgehen. Die Nachricht muss bei Kompromittierung signiert sein. Es liegt in der Entscheidung des Kreditinstituts, ob es auch nicht signierte (anonyme) Schlüsselsperrungen erlaubt (z. B. bei Verlust des Sicherheitsmediums). Die Steuerung erfolgt in den Userparameterdaten über das Feld „Anzahl benötigter Signaturen“. Die Nachricht darf maximal eine Signatur tragen.

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 zur Referenzierung auf den aktuellen Schlüssel jeweils der Wert ‘999’ einzustellen. Es ist daher darauf zu achten, dass dieser Wert reserviert ist und nicht im Rahmen der Versionszählung belegt wird.

Falls das Kreditinstitut unsignierte Sperrungen zulässt, muss dem Benutzer darüber hinaus explizit seine Benutzerkennung mitgeteilt werden. Beim RAH-10-Verfahren erfolgt dies im Rahmen des Ini-Briefs. Beim RAH-7 und RAH-9 kann diese dem Benutzer bei der Aushändigung der Chipkarte mitgeteilt werden.

Im Anschluss an die Sperrnachricht wird …

- die Antwortnachricht sowie die Dialogendenachricht des Kreditinstituts nicht chiffriert, aber signiert (sofern das Kreditinstitut grundsätzlich signiert) und
- die Dialogendenachricht des Kunden chiffriert, aber nicht signiert

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

Anschließend muss der Kunde nach einer Schlüsselsperrung zur Entsperrung eine erneute Erstinitialisierung durchführen.

- **Format**

Name:	Sperrung eines Schlüssels durch den Kunden
Typ:	Nachricht
Version:	4
Sender:	Kunde

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Signaturkopf	SEG	HNSHK	O	1	
3	Schlüsselsperrung	SEG	HKSSP	M	1	
4	Signaturabschluss	SEG	HNSHA	O	1	
5	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals], Kap. II.5.2

- **Belegungsrichtlinien**

Schlüsselsperrung

Dieses Segment enthält die Anforderung für die Schlüsselsperrung.

Eine selektive Schlüsselsperrung ist gegenwärtig nicht zulässig, d.h. es werden immer alle Kundenschlüssel gleichzeitig gesperrt. In der DEG „Schlüsselname“ sind die Merkmale des Signierschlüssels einzustellen (s. Kap. B.6.1.4).

b) Kreditinstitutsnachricht

- **Format**

Name: Bestätigung der Schlüsselsperrung durch das Kreditinstitut
Typ: Nachricht
Version: 4
Sender: Kreditinstitut

Nr.	Name	Typ	Kennung	Status	Anzahl	Anmerkungen
1	Nachrichtenkopf	SEG	HNHBK	M	1	s. [Formals], Kap. II.5.1
2	Signaturkopf	SEG	HNSHK	O	1	
3	Rückmeldungen zur Gesamtnachricht	SEG	HIRMG	M	1	s. [Formals], Kap. II.7.2
4	Rückmeldungen zu Segmenten	SEG	HIRMS	O	n	s. [Formals], Kap. II.7.3
5	Bestätigung der Schlüsselsperrung	SEG	HISSP	M	1	
6	Signaturabschluss	SEG	HNSHA	O	1	
7	Nachrichtenabschluss	SEG	HNHBS	M	1	s. [Formals], Kap. II.5.2

- **Ausgewählte Beispiele für Rückmeldungscodes**

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

C. CHIPAPPLIKATIONEN

C.1 Chipapplikation für RAH

Kapitel C.1.1 dient als Überblick für die Datenstrukturen und Zugriffsregeln der Chipapplikation "DF_NOTEPAD" für SECCOS-Chipkarten [SECCOS] bzw. [SECCOS-6]. Die Spezifikation des DF_NOTEPAD selbst und die Terminalabläufe sind im Dokument [DF_NOTEPAD] enthalten.

Im Verlauf dieses Kapitels ist mit "Bankensignaturkarte" eine Chipkarte mit SECCOS-Betriebssystem und Signaturanwendung gemeint, die u.U. auch die Notepad-Applikation aus Kap. C.1.1 enthält. Weitere Applikationen, wie z. B. die elektronische Geldbörse, sind nicht notwendigerweise auf der Chipkarte enthalten. Ebenso kann die Bankensignaturkarte mit oder ohne Zertifikat ausgeliefert werden.

C.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. kundenspezifische Parameter abgelegt werden können, z. B. für die Bankverbindungsdaten in HBCI.

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 Kundenprodukt 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 HBCI-Bankverbindungsdaten (HBCI-Parameterblock) enthält.
- 'F1' bedeutet, dass der Record Bankverbindungsdaten analog dem DFÜ-Abkommen enthält.

Weitere Kennungen sind für den späteren Gebrauch durch andere Anwendungen vorgesehen (Tag 'F2' bis 'FE').

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

C.1.2 EF_NOTEPAD

Bei dem EF_NOTEPAD handelt es sich um ein lineares EF mit einer variablen Recordlänge, die aus technischen Gründen auf maximal 239[1] Byte begrenzt ist. Es dient der Ablage beliebiger Daten.

---

[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.

Die HBCI Anwendung nutzt das EF_NOTEPAD zur Speicherung von Zugangsspezifischen Daten, den HBCI-Parameterblöcken. So kann ein Online-Banking-Kundenprodukt in einem HBCI-Parameterblock und damit in einem Record des EF_NOTEPADs Informationen wie z. B. die Benutzerkennung ablegen. Darüber hinaus können vom Kundenprodukt in einem separaten weiteren Record aber auch (produktspezifische) Informationen zu Kundenpräferenzen und –einstellungen (z. B. Sprache, Anzeigeparameter etc.) abgelegt werden.

Den Herstellern von Kundensystemen wird vorgeschlagen, beim EF_NOTEPAD neben einer Länge von 239 Byte auch Karten mit einer Maximallänge von nur 200 Byte zu unterstützen. Zur Ermittlung der Maximallänge soll der Tag „82“ des Bereiches FCP ausgelesen werden.

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

Das Auslesen der Records erfolgt über ein *Read Record* auf alle vorhandenen Records. Wird ein HBCI-Parameterblock gesucht so ist anschließend ein Vergleich durchzuführen, ob der TAG des Records den Inhalt ’F0‘ enthält.

Alternativ können mit dem Kommando SEARCH RECORD mit dem Suchmuster ’F0‘ für das erste Byte des Recordinhalts genau die für HBCI relevanten Records ausgelesen 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 239 ('EF') Byte und XX Records
'83'	'02'	'A6 11'	Datei-ID des EF_NOTEPAD
'85'	'02'	'YY YY'	für Nutzdaten allokierter Speicherplatz in Byte (XX Records mal 239 Byte)[2]
'88'	'01'	'D0'	SFI '1A' für das EF_NOTEPAD
'A1'	'08'	'8B 06 00 30 01 04 02 05'	Zugriffsregel-Referenzen

Die maximale Anzahl der Records und deren maximale Länge werden bei der Produktion der Karte festgelegt.

[2] Beispiel: für XX = ‘05‘ a 239 Byte ist ein Datenbereich von 1195 Byte anzulegen →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).

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 $K_{Notepad\_Admin}$ (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 $K_{Notepad\_Admin}$.

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.

- **Aufbau eines Records**

POS	Länge	Wert	Erläuterung
1	1	'XX'	Tag
2	1 oder 2	'XX' oder '81 XX'	Länge (bei Längen über 127 Byte ist die Kodierung '81' 'xx' zu verwenden)
3	L	'XX..XX'	Nutzdaten

Als Tags werden festgelegt:

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

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

Initial werden alle Records mit '00..00' belegt und so als leere Records gekennzeichnet.

- **Beispiel eines EF_NOTEPADs**

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

Record	Eintrag	Erläuterung
1	’F0 XX...XX‘	Erste HBCI-Bankverbindung
2	’F0 XX...XX‘	Zweite HBCI-Bankverbindung
3	’F0 XX...XX‘	Dritte HBCI-Bankverbindung
4	’00..00‘	frei
5	’F1 XX..XX‘	belegt durch Anwendung mit Kennung ’F1‘
6	’00..00‘	frei
7	’F0 XX...XX‘	Vierte HBCI-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 von der Karte zurückgegeben.

  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 HBCI-Recordeintrag beginnt in diesem Fall mit dem Tag ‘F0‘ und einem Längenbyte.

C.1.2.1 Recordbelegung des EF_NOTEPAD mit einem HBCI-Parameterblock, Version 001

Die Belegung gemäß Version 001 wird aufgrund nicht mehr zugelassener Sicherheitseigenschaften wie z. B. RIPEMD-160 nicht mehr unterstützt.

C.1.2.2 Recordbelegung des EF_NOTEPAD mit einem HBCI-Parameterblock, Version 002

Ein HBCI-Recordeintrag hat bei V002 folgenden prinzipiellen Aufbau:

Tag		Länge (Byte)	Wert	Format	Status	Erläuterung
'F0'		Var. max 'EC'[3]				HBCI-Parameterblock
	'C0'	'03'	'30' '30' '32'	3an	M	Version 002 des HBCI-Parameterblocks
'E1'		Var. max. '5B'			M	HBCI-Institutsparameterblock
	'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'	'27'	Hashwert Instituts-schlüssel	39bin	O	
	'C5'	'01'	Schlüsselstatus	1bin	M	8 Statusflags
'E2'		Var. max. '37'			M	HBCI-Kommunikations-parameterblock
	'C6'	'01'	Kommunikations-dienst	1n	M	2 = TCP/IP
	'C7'	'01'-'32'	Kommunikations-adresse	..50an	M	
'E2'		Var. max. '37'			O	2. HBCI-Kommunikations-parameterblock
	'C6'	'01'	Kommunikations-dienst	1n	M	2 = TCP/IP
	'C7'	'01'-'32'	Kommunikations-adresse	..50an	M	
'E3'		Var. max. '54'			O	HBCI-Kundenparameterblock
	'C8'	'01'-'1E'	Benutzerkennung	..30an	M	
	'C9'	'01'-'1E'	Kunden-ID	..30an	O	
	'CA'	'0C' oder '12''	Info Inhaber-schlüssel	12an oder 18an	M	Schlüsselnummer und Schlüsselversion jeweils für den Signierschlüssel, den Chiffrierschlüssels und optional für den Signaturschlüssel des Karteninhabers

[3] Nettodatenlänge ‚EC'=236 Byte + 3 Byte Längenfeld ergibt die maximale Recordlänge von 239 Byte

C.1.2.2.1 Tag ‘F0‘: HBCI-Parameterblock

Durch das Tag ‘F0‘ wird ein Record mit HBCI-Parameterblock für die HBCI-Anwendung gekennzeichnet. Für Belegungen der EF_NOTEPAD-Records durch andere Anwendungen stehen die Tags ‘F1‘ bis ‘FE‘ zur Verfügung.

Ein HBCI-Parameterblock enthält in der angegebenen Reihenfolge:

- **optional** ein Versionskennzeichen
- genau einen HBCI-Institutsparameterblock mit **Tag ‘E1‘**
- genau einen HBCI-Kommunikationsparameterblöcke mit **Tag ‘E2‘**
- **optional** einen weiteren HBCI-Kommunikationsparameterblöcke mit **Tag ‘E2**[4]
- **optional** einen HBCI-Kundenparameterblock mit **Tag ‘E3‘**

Die maximale Länge des HBCI-Parameterblocks wird beschränkt durch die maximale Recordlänge von 239 Byte[5].

C.1.2.2.2 Tag ‘C0‘: HBCI-Version

In jedem ‘F0‘ Record kann zur Kennzeichnung der Version des EF-NOTEPAD ein Sub-Record (z. B ’C0’ ’03’ ’30’ ’30’ ’30’) aufgenommen werden. Die Zählung der Version beginnt bei 1. Ist kein Sub-Record ’C0’ vorhanden, so bedeutet dieses, dass die Belegung des EF-NOTEPAD gemäß der Version 1 erfolgt.

C.1.2.2.3 Tag ‘E1‘: HBCI-Institutsparameterblock

Durch das Tag ‘E1‘ wird der Block der institutsspezifischen Parameter gekennzeichnet. Ein HBCI-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 (gemäß FinTS Datenelement ‚Kreditinstitutscode‘) verwendet.
- **optional** einen Hashwert des öffentlichen Signierschlüssels des Instituts mit **Tag ‘C4‘**, binär mit genau 39 Byte für die Hashwertverfahren SHA-256.

  Der Eintrag besteht aus

  [3 Byte Schlüsselnummer | 3 Byte Schlüsselversion | 1 Byte Kennzeichen Hashverfahren | 32 Byte Hashwert].

  Als Kennzeichen für das Hashverfahren werden festgelegt:

---

4 Somit ist der erste HBCI-Kommunikationsparameterblock ist also verpflichtend, der zweite optional.

5 In einer konkreten Umsetzung ist es nicht möglich einen HBCI-Parameterblock mit allen Felder in der maximalen Länge zu nutzen. Dabei würde die maximale Recordlänge von 239 Byte überschritten.

- ‘03‘ = SHA-256 für RAH-7 und RAH-9

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 Kundenschlüssel notwendig	'1'b - Ja '0'b - Nein
Bit2	Institutsrechner erwartet Signaturen nach ISO9796 mit AnnexA	'1'b - Ja '0'b - Nein
Bit3	Institutsschlüssel validiert	'1'b - Ja '0'b - Nein
Bit4	Ausstehende Übermittlung des neuen öffentlichen Chiffrierschlüssels des Kunden bei Schlüsseländerung[7]	'1'b - Ja '0'b - Nein
Bit5	Ausstehende Übermittlung des neuen öffentlichen Signierschlüssels des Kunden bei Schlüsseländerung[8]	'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 HBCI-Institutsparameterblock belegt inklusive der Tag- und Längenbytes somit maximal 93 Byte.

C.1.2.2.4 Tag ‘E2‘: HBCI-Kommunikationsparameterblock

Durch das Tag ‘E2‘ wird der Block der generellen Kommunikations-Parameter gekennzeichnet. Ein HBCI-Kommunikationsparameterblock enthält in der angegebenen Reihenfolge:

- genau einen Kommunikationsdienst mit **Tag ‘C6‘**, 1 Stelle numerisch. Zurzeit definiert ist der numerische Wert 2 (TCP/IP)
- genau eine Kommunikationsadresse mit **Tag ‘C7‘**, alphanumerisch mit bis zu 50 Zeichen

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

---

[7] Nicht zu belegen, da die DK-Bankensignaturkarte keinen Wechsel der Kundenschlüssel unterstützt.
[8] Nicht zu belegen, da die DK-Bankensignaturkarte keinen Wechsel der Kundenschlüssel unterstützt.

C.1.2.2.5 Tag ‘E3‘: HBCI-Kundenparameterblock

Durch den Tag ‘E3‘ wird der **optional** vorhandene Block der kundenspezifischen Parameter gekennzeichnet. Ist der Block nicht vorhanden, so handelt es sich um eine im Rahmen der HBCI-Anwendung Bankensignaturkarte ohne Zertifikat. Ein HBCI-Kundenparameterblock 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 oder 18 numerischen Zeichen**.**

  Bei 12 Byte Länge des Blocks ist der Inhalt wie folgt definiert:

Schlüsselnummer Signierschlüssel [3n]
Schlüsselversion Signierschlüssel [3n]
Schlüsselnummer Chiffrierschlüssel [3n]
Schlüsselversion Chiffrierschlüssel [3n]

  Bei 18 Byte Lange ist der Inhalt wie folgt definiert:

Schlüsselnummer Signierschlüssel [3n]
Schlüsselversion Signierschlüssel [3n]
Schlüsselnummer Chiffrierschlüssel [3n]
Schlüsselversion Chiffrierschlüssel [3n]
Schlüsselnummer Signaturschlüssel [3n]
Schlüsselversion Signaturschlüssel [3n]

Die Parameter Schlüsselnummer und Schlüsselversion werden in je 3 Byte numerisch rechtsbündig mit führenden Nullen angegeben. (z. B. Schlüsselnummer 1 → "001" → die Bytefolge '30' '30' '31'.

Fehlen die Angaben für den Signaturschlüssel (CA Record der Länge 12 Byte) so werden als Schlüsselnummer und Schlüsselversion des Signaturschlüssels die Schlüsselnummer und Schlüsselversion des Signierschlüssels übernommen.

Fehlt der Teilrecord mit dem Tag 'CA' (nicht vorhandener Record E3 oder Record CA oder fehlendes EF_NOTEPAD) und liegen somit weder für den Signierschlüssel und den Chiffrierschlüssel noch für den Signaturschlüssel Schlüsselnummer und Schlüsselversion vor so sind vom FinTS-Kundensystem die Schlüsselnummern und Schlüsselversionen aller Schlüssel nach folgenden Mechanismen vorzubesetzen.

Die Schlüsselnummer wird gemäß dem genutzten RAH-Verfahren besetzt. Die Schlüsselversion wird gängigerweise im ersten Ausgabejahr mit "001" vorbesetzt und anschließend im jährlichen Turnus um 1 erhöht.

RAH Verfahren	Schlüsselnummer	Schlüsselversion
RAH7	"007" → '30' '30' 37'	"001" → '30' '30' '31'
RAH9	"009" → '30' '30' 39'	"001" → '30' '30' '31'
RAH10	"010" → '30' '31' 30'	"001" → '30' '30' '31'

Über die Schlüsselnummer im EF_NOTEPAD kann das zu verwendende Sicherheitsprofil ermittelt werden.

Wichtiger Hinweis:
Bei allen Verfahren müssen für die Schlüsselnummer die entsprechenden Werte aus der obigen Tabelle verwendet werden. Die Nutzung von Schlüsselnummer „001“ ist nicht erlaubt.

Ein HBCI-Kundenparameterblock belegt inklusive der Tag- und Längenbytes somit maximal 86 Byte.

C.1.2.2.6 Beispiel

Beispiel für eine Recordbelegung (Tags und Längenbytes sind fett markiert)

Inhalt	Erläuterung
F0 81 76	HBCI-Parameterblock
E1 3D	Institutsparameterblock
C1 0C 54 45 53 54 49 4E 53 54 49 54 55 54	Institutsbezeichnung “TESTINSTITUT“
C2 03 32 38 30	Länderkennzeichen “280“
C3 08 31 32 33 34 35 36 37 38	BLZ 12345678
C4 1B 30 30 31 30 30 31 03 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20	Schlüsselnummer 1, Schlüssel-version 1, Hashverfahren SHA-256, Hashwert
C5 01 01	Schlüsselstatus '01'
E2 12	Kommunikationsparameterblock
C5 01 02	Kommunikationsdienst TCP/IP
C6 0D 31 39 32 2E 31 36 38 2E 31 31 2E 32 32	Kommunikationsadresse 192.168.11.22
E3 21	Kundenparameterblock
C8 0A 31 32 33 34 35 36 37 38 39 30	Benutzerkennung "1234567890"
C9 05 31 32 33 34 35	Kunden-ID "12345"
CA 0C 30 30 31 30 30 31 30 30 31 30 30 31	Info Inhaberschlüssel Schlüsselnummer SIG 1, Schlüsselversion SIG 1 Schlüsselnummer CHIF 1, Schlüsselversion CHIF 1

C.1.2.2.7 Erreichen der maximalen Recordlänge

Bei Ausnutzung aller Maximallängen und Aufnahme aller optionalen Felder und Angabe zweier Kommunikationsparameterblöcke und eines Kundenparameterblocks ergibt sich ein maximaler Platzbedarf von 297 Byte. Dieser Platzbedarf ist aber in einem Record nicht abbildbar. Normalerweise wird aber nur ein Kommunikationsparameterblock verwendet sowie selten alle Maximallängen ausgereizt, so dass meistens die maximale Recordlänge von 239 Byte genügt. Bei älteren bereits ausgegebenen Bankensignaturkarten ist nur eine maximale Recordlänge von 200 Byte vorgesehen.

C.1.3 Terminalabläufe

Dieses Kapitel spezifiziert die Terminalabläufe im Umgang mit dem RAH-Verfahren auf SECCOS-Chipkarten [SECCOS-6] bzw. [SECCOS-7]. Ein Online-Banking-Kundenprodukt nutzt

- zur Verschlüsselung und Signierung von HBCI-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 Kap. C.1.3.1),
- als Datenspeicher für die Zugangsdaten ein auf der Chipkarte optional vorhandenes DF_NOTEPAD ([DF_NOTEPAD], siehe Kap. C.1.1).

C.1.3.1 Verfahren zur Ermittlung der Sicherheitsreferenznummern

Auf der Bankensignaturkarte wird kein eigenständiger Sequenzzähler verwaltet, sondern es werden jeweils chipkarteninterne „Usage Counter“ der beiden zur Signatur verwendeten Schlüssel S_K.CH.DS und S_K.CH.$AUT_{C/S}$ herangezogen.

Für jedes Signaturschlüsselpaar wird ein separater Usage Counter verwaltet.Dieser kann jeweils zwei, drei oder vier Byte lang sein.

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

1. Auslesen des 2 bis 4 Byte langen Usage Counter (UC) UC_{DS} des Schlüssels S_K.CH.DS bzw. UC_{AUT} des Schlüssels S_K.CH.$AUT_{C/S}$.
2. Sei **neg**(UC) die bitweise logische Negation von UC. Dann ist die Sicherheitsreferenznummer (SRN)

$$SRN_{DS} = \mathbf{neg}(UC_{DS})$$

$$SRN_{AUT} = \mathbf{neg}(UC_{AUT})$$

Die einzelnen Usage Counter haben folgende Wertebereiche:

von 0 bis 65.535	bei Länge(UC) = 2 Byte
von 0 bis 16.777.215	bei Länge(UC) = 3 Byte
von 0 bis 4.294.967.295	bei Länge(UC) = 4 Byte

Damit muss die Sicherheitsreferenznummer SRN über die entsprechenden Wertebereiche verfügen und benötigt zur Darstellung ebenfalls mindestens 2, 3 oder 4 Byte.

Ein Wrap-Around bei Erreichen des jeweiligen Maximalwerts findet nicht statt, da das Erreichen eines Usage Counter 0 den Schlüssel der Chipkarte für die weitere Verwendung sperrt.

Beispiel:

UC_{DS} = ’00 0A‘ (dezimal 10) $\Rightarrow$ SRN_{DS} = **neg**(UC_{DS}) = ‘FF F5‘ (dezimal 65.525)

UC_{AUT} = ‘FA 1D‘ (dezimal 64.029) $\Rightarrow$ SRN_{AUT} = **neg**(UC_{AUT}) = ’05 E2‘(dezimal 1506)

Dieser Algorithmus ist in der jeweiligen Anwendungssoftware zu realisieren.

C.1.3.2 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-Authentikation zwischen Chipkarte und Geschäftsterminal mit Aushandlung eines Sessionkey-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 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 Fehlbedienungszählers) werden von den Kreditinstituten bzw. den Kartenemittenten Softwarekomponenten zur Verfügung gestellt werden, die in der privaten Kundenumgebung zum Einsatz kommen sollen. In Kundenprodukten, die nicht von den Kartenemittenten herausgegeben werden, sollen diese Administrationsfunktionen nicht realisiert werden.

Für die kreditinstitutsseitige Realisierung dieser Softwarekomponenten hat die Deutsche Kreditiwirtschaft Anforderungen und Festlegungen formuliert, die bei Bedarf über die jeweiligen Ansprechpartner der Standards erhältlich sind.

C.1.3.2.1 Signatur einleiten

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

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 ungleich ’00‘, so sind Parameterdaten gemäß Kap. C.1.1 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

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

C.1.3.2.2 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“ (Kap. C.1.3.2.1) geschehen. Für die folgende Ablaufbeschreibung wird angenommen, dass die Anwendung bereits Auftrags-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 jeweiligen Spezifikationen eingefügt.

C.1.3.2.3 Nachrichten signieren

C.1.3.2.3.1 Nachrichten signieren bei HBCI

Die folgenden Abläufe können im Falle von HBCI offline, d.h. außerhalb des Übertragungsdialogs vollzogen werden. Dies gilt für alle Nachrichten mit Ausnahme der Dialoginitialisierung. Der Grund besteht darin, dass für die Absicherung aller Kreditinstitutsnachrichten der Schlüssel des Senders der Dialoginitialisierungsnachricht erforderlich ist. Daher muss auch die Chipkarte des Senders während des gesamten Dialogs im Endgerät stecken.

Die Abläufe für die Signatur der Dialoginitialisierungsnachricht sind grundsätzlich identisch mit den im Folgenden beschriebenen Abläufen für die Signatur von Auftragsnachrichten. Da aber für die Dialoginitialisierung anwendungsseitig noch weitere Chipkartendaten (Benutzerkennung, Dialog-ID, Kommunikationszugang etc.) benötigt werden, wird der komplette Ablauf einschließlich der Signatur der Dialoginitialisierung im Kap. C.1.3.2.6 "Übertragungsdialog" noch einmal beschrieben.

Chipkarte			Endgerät	
R1	BZ	→ ←	M1	Sequenzzähler (Signatur-ID) ermitteln durch Aufruf der ZKA-SIG-API-Funktion *zka_sig_read_key_usage_counter* und anschließende Invertierung des Rückgabewerts gemäß Abschnitt C.1.3.1)
			A2	Signaturkopf aufbauen und in HBCI-Nachricht einfügen
			A3	Daten (Signaturkopf, HBCI-Nutzdaten) für Signatur bereitstellen
		→ ←	M4	Signaturerstellung (siehe Kap. C.1.3.3.1)
			A5	Signaturabschluss aufbauen und in HBCI-Nachricht einfügen
			A6	ggf. M1 bis A5 für weitere Nachrichten wiederholen
			A7	signierte HBCI-Nachrichten zur Weiterverarbeitung speichern

♦ **Erläuterung**

1. 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 Kap. C.1.3.1 zu SZ = **neg**(BZ) invertiert und als Sequenzzähler gespeichert.

2. Der Signaturkopf wird aufgebaut und in die FinTS-Nachricht eingefügt.
3. Die Daten (Signaturkopf, FinTS-Nutzdaten) für die Signaturerstellung werden bereitgestellt.
4. Die Signatur wird berechnet (siehe hierzu Kap. C.1.3.3).
5. Der Signaturabschluss wird aufgebaut und in die FinTS-Nachricht eingefügt.
6. Ggf. können die Schritte 1 bis 5 für weitere Nachrichten wiederholt werden.
7. 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 (Kap. C.1.3.2.4) durchgeführt wird.

C.1.3.2.4 Nachrichten verschlüsseln

C.1.3.2.5 Nachrichten verschlüsseln bei RAH

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 wurde[9].

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

[9] [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.“

Chipkarte			Endgerät	
			A1	Daten (FinTS-Nutzdaten und ggf. Signatur) für die Verschlüsselung bereitstellen
R2	RND	←	C2	Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge*
		→	A2	RND als Einmalschlüssel-Fragment KS_{LL} speichern
R3	RND	←	C3	Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge*
		→	A3	RND als Einmalschlüssel-Fragment KS_{LR} speichern
R4	RND	←	C4	Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge*
		→	A4	RND als Einmalschlüssel-Fragment KS_{RL} speichern
R5	RND	←	C5	Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge*
		→	A5	RND als Einmalschlüssel-Fragment KS_{RR} speichern
			A6	KS_{LL}, KS_{LR}, KS_{RL} und KS_{RR} zu KS konkatenieren und speichern
			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

♦ **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 HBCI-Karte geben.

   Wenn das Kommando erfolgreich ausgeführt wurde, gibt die HBCI-Karte eine 8 Byte lange Zufallszahl als Antwortdatum aus, die als Einmalschlüssel-Fragment KS_{LL} gespeichert wird.
3. Mit dem Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge* lässt sich das Terminal eine zweite Zufallszahl von der HBCI-Karte geben, die als Einmalschlüssel-Fragment KS_{LR} gespeichert wird.
4. Mit dem Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge* lässt sich das Terminal eine dritte Zufallszahl von der HBCI-Karte geben, die als Einmalschlüssel-Fragment KS_{RL} gespeichert wird.
5. Mit dem Aufruf der ZKA-SIG-API-Funktion *zka_sig_get_challenge* lässt sich das Terminal eine vierte Zufallszahl von der HBCI-Karte geben, die als Einmalschlüssel-Fragment KS_{LL} gespeichert wird.
6. $KS_{LL,}$ $KS_{LR,}$ $KS_{RL,}$ und $KS_{RR,}$ werden zu KS konkateniert und gespeichert.
7. Die zu übertragenden Daten werden mit KS symmetrisch verschlüsselt (AES CBC-Mode, IV=0, ZKA-Padding).

8. Der Einmalschlüssel KS wird linksbündig mit Nullbits auf die Schlüssellänge 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. Das Ergebnis wird mit führenden Nullbits auf die Schlüssellänge erweitert.
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.

C.1.3.2.6 Übertragungsdialog

Chipkarte			Endgerät			Kreditinstitut	
			A1	Benutzerkennung aus der bereits gelesenen Bankverbindung extrahieren			
		→ ←	M2	Nachricht signieren (s. Kap. C.1.3.2.3)			
			A3	Kommunikationszugang aus Bankverbindung herstellen			
			C4	Nachricht (beginnend mit Dialoginitialisierungsnachricht) senden	→ ←	R4	Antwortnachricht
			A5	falls Antwortnachricht verschlüsselt: Daten (Binärdaten nach dem Verschlüsselungskopf) und verschlüsselten Session-Key enc(KS) aus dem Signaturkopf für die Entschlüsselung bereitstellen			
		→ ←	M6	Ausführung der ZKA-SIG-API-Funktion *zka_sig_decrypt* zur Session-Key-Entschlüsselung, Resultat ist der Session-Key KS			
			A7	Daten mit Session-Key KS entschlüsseln.			
			A8	falls Kreditinstitutsnachricht signiert: Daten (Signaturkopf, Nutzdaten, Signatur) für Signatur-Prüfung bereitstellen			
		→ ←	M9	Signatur-Prüfung (siehe KapC.1.3.3.2)			
			A10	C4 bis M9 für alle weiteren FinTS-Nachrichten wiederholen			

C.1.3.2.7 Signatur beenden

Chipkarte	

Endgerät	
M1	Aufruf der ZKA-SIG-API-Funktion *zka_sig_fini_signature_application*
M2	Aufruf der ZKA-SIG-API-Funktion *zka_sig_close*

- **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.

C.1.3.3 Makros

C.1.3.3.1 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-Authentikations-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 ist dies insbesondere eine vorhergehende Benutzer-Authentikation 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-Authentikation zu wiederholen ist. Eine Benutzer-Authentikation wird bei Bedarf innerhalb der ZKA-SIG-API-Funktionen *zka_sig_digital_signature* bzw. *zka_sig_cs_authentication* durchgeführt.

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

♦ **Erläuterung**

1. Die Berechnung des Hashwertes erfolgt in der Regel außerhalb der Chipkarte (Hashalgorithmus gemäß Vorgabe für den Sicherheitsdienst bzw. vom Institut übermittelter BPD). Optional ist es auch möglich, den letzen Schritt oder alle Schritte der Hashwert-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 = 2.16.840.1.101.3.4.2.1 für SHA-256

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 Paddingverfahrens erfolgt gemäß Vorgabe für den Sicherheitsdienst bzw. vom Institut übermittelter BPD. Die Signaturanwendung der Bankensignaturkarte bietet ab SECCOS 6 das Signaturverfahren RSASSA-PSS an.

Falls der Hashwert 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 Paddingformat gemäß PKCS#1 ([SECCOS, Kapitel 8.3.2.1][11]), 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' bzw. '31'		Tag und Länge von SEQUENCE (SHA-256)
'30'	'09' bzw. '0D'		Tag und Länge von SEQUENCE (SHA-256)
'06'	'09'	'60 86 48 01 65 03 04 02 01'	OID des SHA-256 (2 16 840 1 101 3 4 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 Hashwerts ist bei der Signatur im Sicherheitsdienst „Authentication“ nicht möglich. Der Hashwert (als Ergebnis von Schritt 1) muss daher explizit als Aufrufparameter in der oben beschriebenen Form in Schritt 2 übergeben werden.

[11] 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)

C.1.3.3.2 Signatur-Prüfung

Die Bankensignaturkarte selbst unterstützt zurzeit keine Signatur-Prüfung[12]. Die Prüfung einer Signatur wird vom Kundenterminal-Makro “Überprüfen der Korrektheit der elektronischen Unterschrift” durchgeführt.

Die (mathematische) Korrektheit einer elektronischen Unterschrift wird überprüft, in dem sie mit dem entsprechenden öffentlichen Schlüssel entschlüsselt wird und das Ergebnis mit dem Hashwert 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].

[12] [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.“

D. DATA DICTIONARY

A

Austauschkontrollreferenz

Dialog-ID der korrespondierenden Nachricht des Kunden (vgl. Kapitel B.6.1.3).

Typ:	DE
Format:	id
Länge:	#
Version:	1

B

Benutzerdefinierte Signatur

Bei nicht-schlüsselbasierten Sicherheitsverfahren kann der Benutzer hier Angaben zur Authentisierung machen. Ob das Feld verpflichtend ist, ist vom jeweiligen Sicherheitsverfahren abhängig.

Format: s. Spezifikation „Sicherheitsverfahren PIN/TAN“

Typ:	DEG
Format:	
Länge:	
Version:	1

Benutzerkennung

Eindeutig vergebene Kennung, anhand deren die Identifizierung des Benutzers erfolgt. Die Vergabe obliegt dem Kreditinstitut. Das Kreditinstitut hat zu gewährleisten, dass die Benutzerkennung institutsweit eindeutig ist. Sie kann beliebige Informationen enthalten, darf aber bei Verwendung des RAH-Verfahrens aus Sicherheitsgründen nicht aus benutzer- oder kreditinstitutsspezifischen Merkmalen hergeleitet werden.

Typ:	DE
Format:	id
Länge:	#
Version:	1

Bereich der Sicherheitsapplikation, kodiert

Information darüber, welche Daten vom kryptographischen Prozess verarbeitet werden. Diese Information wird benötigt um z. B. zwischen relevanter und belangloser Reihenfolge von Signaturen zu unterscheiden (vgl. [HBCI], Kapitel VI.4).

Wenn SHM gewählt wird, so bedeutet dies, dass nur über den eigenen Signaturkopf sowie die HBCI-Nutzdaten ein Hashwert gebildet wird, der in die Signatur eingeht. Dies entspricht bei Mehrfachsignaturen einer bedeutungslosen Reihenfolge.

Wenn SHT gewählt wird, dann werden auch alle schon vorhandenen Signaturköpfe und -abschlüsse mitsigniert. Das heißt, dass die Reihenfolge der Signaturen relevant ist.

Codierung:

1: Signaturkopf und HBCI-Nutzdaten (SHM)

2: Von Signaturkopf bis Signaturabschluss (SHT)

Typ: DE
Format: code
Länge: ..3
Version: 2

Bezeichner für Algorithmusparameter, IV

Eigenschaft betreffend den Initialisierungswert für die RAH-Verfahren (Die Steuerung erfolgt in den BPD, vgl. [Formals]).

Codierung:

1: Initialization value, clear text (IVC)

Typ: DE
Format: code
Länge: ..3
Version: 2

Bezeichner für Algorithmusparameter, Schlüssel

Eigenschaft des Schlüssels für die RAH-Verfahren (Die Steuerung erfolgt in den BPD, vgl. [Formals]).

Codierung:

5: Symmetrischer Schlüssel (nicht zugelassen)

6: Symmetrischer Schlüssel, verschlüsselt mit einem öffentlichen Schlüssel bei RAH und RDH (KYP).

Typ: DE
Format: code
Länge: ..3
Version: 2

Bezeichner für Exponent

Enthält den Bezeichner für den Exponent des öffentlichen Schlüssels.

Codierung:

13: Exponent (EXP)

Typ: DE
Format: code
Länge: ..3
Version: 2

Bezeichner für Funktionstyp

Enthält den Bezeichner für den Funktionstyp des Key-Management.

Codierung:

112: ‘Certificate Replacement’ (Ersatz des Zertifikats) im Zusammenhang mit der Schlüsseländerung

124: ‘Certificate Status Request’ im Zusammenhang mit der Anfrage für einen öffentlichen Schlüssel

224: ‘Certificate Status Notice’ im Zusammenhang mit der Übermittlung eines öffentlichen Schlüssels

130 : ‘Certificate Revocation’ (Zertifikatswiderruf) im Zusammenhang mit der Schlüsselsperrung

231: ‘Revocation Confirmation’ (Bestätigung des Zertifikatswiderrufs) im Zusammenhang mit der Bestätigung der Schlüsselsperrung

Typ:	DE
Format:	code
Länge:	..3
Version:	2

Bezeichner für Hashalgorithmusparameter

Bezeichner für den Hashalgorithmusparameter.

Codierung:

1: IVC (Initialization value, clear text)

Typ:	DE
Format:	code
Länge:	..3
Version:	2

Bezeichner für Modulus

Bezeichner für den Modulus des öffentlichen Schlüssels.

Codierung:

12: Modulus (MOD)

Typ:	DE
Format:	code
Länge:	..3
Version:	2

Bezeichner für Sicherheitspartei

Identifikation der Funktion der beschriebenen Partei, in diesem Falle des Kunden.

Codierung:

1: Message Sender (MS), wenn ein Kunde etwas an sein Kreditinstitut sendet.

2: Message Receiver (MR), wenn das Kreditinstitut etwas an seinen Kunden sendet.

Typ: DE
Format: code
Länge: ..3
Version: 2

Bezugssegment

Sofern sich ein Kreditinstitutssegment auf ein bestimmtes Kundensegment bezieht (z. B. Antwortrückmeldung auf einen Kundenauftrag) hat das Kreditinstitut die Segmentnummer des Segments der Kundennachricht einzustellen, auf das sich das aktuelle Segment bezieht (s. DE „Segmentnummer“). In Zusammenhang mit den Angaben zur Bezugsnachricht aus dem Nachrichtenkopf ist hierdurch eine eindeutige Referenz auf das Segment einer Kundennachricht möglich.

Falls die Angabe eines Bezugssegments erforderlich ist, ist dieses bei der Formatbeschreibung eines Kreditinstitutsegments angegeben.

Typ: DE
Format: num
Länge: ..3
Version: 1

C

CID

(Cardholder Identification) Identifikation der verwendeten Chipkarte. Die CID steht im EF_ID der Karte.

Typ: DE
Format: bin
Länge: ..256
Version: 1

D

Daten, verschlüsselt

Enthält die verschlüsselten (und komprimierten) Daten.

Typ: DE
Format: bin
Länge: ..
Version: 1

Datum

Datumsangabe, zur Bestimmung eines Zeitpunktes.

Typ: DE
Format: dat
Länge: #
Version: 1

Datum- und Zeitbezeichner, kodiert

Enthält die Bedeutung des Zeitstempels.

Codierung:

1: Sicherheitszeitstempel (STS)

6: Certificate Revocation Time (CRT)

Typ: DE
Format: code
Länge: ..3
Version: 2

F

Filterfunktion

Falls das Übertragungsverfahren eine Umwandlung der Nachricht in eine 7 Bit-Zeichendarstellung erfordert (z. B. Internet), so ist hier das anzuwendende Filterverfahren anzugeben. Die Nachricht ist stets komplett zu filtern, auch wenn eine Filterung nicht notwendig wäre, da bspw. keine binären Daten enthalten sind. Ein Kreditinstitut darf jeweils nur eine Filterfunktion unterstützen.

Codierung:

MIM: MIME Base 64

UUE: Uuencode/Uudecode

Typ: DE
Format: an
Länge: 3
Version: 1

H

Hashalgorithmus

Angaben zu einem kryptographischen Algorithmus, seinen Operationsmodus, sowie dessen Einsatz.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Verwendung des Hashalgorithmus, kodiert	DE	code	..3	M	1	1
2	Hashalgorithmus, kodiert	DE	code	..3	M	1	3, 4, 5, 6
3	Bezeichner für Hashalgorithmusparameter	DE	code	..3	M	1	1
4	Wert des Hashalgorithmusparameters	DE	bin	..512	O	1	

Typ: DEG
Format:
Länge:
Version: 2

Hashalgorithmus, kodiert

Code des verwendeten Hash-Algorithmus.

Codierung:

1: SHA-1 (nicht zugelassen)

2: belegt

3: SHA-256

4: SHA-384

5: SHA-512

6: SHA-256 / SHA-256

999: Gegenseitig vereinbart (ZZZ); (nicht zugelassen)

Typ: DE
Format: code
Länge: ..3
Version: 2

Wird als „Hashalgorithmus, kodiert“ die Option „6: SHA-256 / SHA256“ gewählt, so findet ein Hashing sowohl in Software als auch in der Bankensignaturkarte statt.
Die Anwendung muss dafür Sorge tragen, dass in der Karte das gewünschte Hashverfahren – hier SHA-256 – selektiert wird; ansonsten würde in dort das Default-Hashverfahren angewendet, was nicht zulässig ist.

I

Identifizierung der Partei

Code, welcher die (Kommunikations-)Partei identifiziert. Bei Verwendung des RAH-Verfahrens ist die Kundensystem-ID einzustellen.

Typ: DE
Format: id
Länge: #
Version: 1

K

Kommunikationsadresse

Beim Zugang über TCP/IP ist die IP-Adresse als alphanumerischer Wert (z. B. '123.123.123.123') einzustellen.

Beim Zugang über https ist die Adresse des Servlets als alphanumerischer Wert (z. B. „https://www.xyz.de:7000/Servlet") einzustellen.

Typ: DE
Format: an
Länge: ..512
Version: 1

Kommunikationsadressenzusatz

Beim Zugang über TCP/IP und https wird das Feld nicht belegt.

Typ: DE
Format: an
Länge: ..512
Version: 1

Kommunikationsdienst

Unterstütztes Kommunikationsverfahren (Protokollstack).

Zur Zeit unterstützte Kommunikationsverfahren:

1: nicht belegt

2: TCP/IP (Protokollstack SLIP/PPP)

3: https

Typ: DE
Format: code
Länge: ..2
Version: 3

Kommunikationsparameter

Die Kommunikationsparameter enthalten Informationen für den Aufbau der Transportverbindung.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Kommunikationsdienst	DE	num	..2	M	1	1,2,3
2	Kommunikationsadresse	DE	an	..512	M	1	
3	Kommunikationsadressenzusatz	DE	an	..512	C	1	O: wird im Kreditinstitut ignoriert
4	Filterfunktion	DE	an	3	C	1	MIM, UUE M: ‚Kommunikationsdienst' = 2 N: sonst
5	Version der Filterfunktion	DE	num	..3	C	1	O: ‚Filterfunktion' belegt N: sonst

Typ: DEG
Format:
Länge:
Version: 2

Komprimierungsfunktion

Code der unterstützten Komprimierungsfunktion.

Codierung:

0: Keine Kompression (NULL)

1: Lempel, Ziv, Welch (LZW)

2: Optimized LZW (COM)

3: Lempel, Ziv (LZSS)

4: LZ + Huffman Coding (LZHuf)

5: PKZIP (ZIP)

6: deflate (GZIP) (http://www.gzip.org/zlib)

7: bzip2 (http://sourceware.cygnus.com/bzip2/)

999: Gegenseitig vereinbart (ZZZ)

Typ: DE
Format: code
Länge: ..3
Version: 2

Komprimierungsversion

Version der unterstützten Komprimierungsfunktion.

Momentan werden alle zulässigen Komprimierungsfunktionen mit Version 1 verwendet. Falls keine Komprimierung verwendet wird (Komprimierungsfunktion 0), wird Version 0 angegeben.

Typ: DE
Format: num
Länge: ..3
Version: 1

Kreditinstitutscode

Landesspezifische Kennung, die das Kreditinstitut eindeutig identifiziert. In Deutschland wird die Bankleitzahl eingestellt. Bei Kreditinstituten, die in Ländern ohne Institutskennungssystem beheimatet sind, kann die Belegung entfallen.

Typ: DE
Format: an
Länge: ..30
Version: 1

Kreditinstitutskennung

Kennung eines Kreditinstituts.

Typ:	DEG
Formatkennung	kik
Länge:	#
Version:	1

Kunden-ID

Institutsweit eindeutige Identifikation des Kunden. Die Vergabe obliegt dem Kreditinstitut. Die Kunden-ID kann beliebige Informationen enthalten. Es steht dem Kreditinstitut frei, ob es jedem Kunden genau eine Kunden-ID zuordnet oder dem Kunden in Abhängigkeit vom Benutzer jeweils eine unterschiedliche Kunden-ID zuordnet.

Typ:	DE
Format:	id
Länge:	#
Version:	1

L

Länderkennzeichen

Länderkennzeichen gemäß ISO 3166-1 (numerischer Code) (s. [Formals], Kap. „Anlagen“). Für Deutschland wird der Code 280 verwendet da dieser im Kreditgewerbe gebräuchlicher als der neue Code 276 ist.

Typ:	DE
Format:	ctr
Länge:	#
Version:	1

N

Nachrichtenbeziehung, kodiert

Code der Nachrichtenbeziehung. Im Zusammenhang mit der Übermittlung eines öffentlichen Schlüssels oder mit der Bestätigung der Schlüsselsperrung ist der Wert „1“ vorgesehen. Im Zusammenhang mit der Schlüsseländerung, mit der Anfrage nach einem öffentlichen Schlüssel oder mit der Schlüsselsperrung ist der Wert „2“ vorgesehen.

Codierung:

1: Key-Management-Nachricht ist Antwort

2: Key-Management-Nachricht erwartet Antwort

Typ:	DE
Format:	code
Länge:	1
Version:	2

Nachrichtenreferenznummer

Nachrichtennummer der korrespondierenden Nachricht des Kunden.

Typ: DE
Format: num
Länge: ..4
Version: 1

O

Öffentlicher Schlüssel

Information, die beim RAH-Key-Management zum Transport des öffentlichen Schlüssels zwischen Kunde und Kreditinstitut bzw. umgekehrt dient.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Verwendungszweck für öffentlichen Schlüssel	DE	code	..3	M	1	5, 6
2	Operationsmodus, kodiert	DE	code	..3	M	1	2, 18, 19
3	Verfahren Benutzer	DE	code	..3	M	1	10
4	Wert für Modulus	DE	bin	..512	M	1	
5	Bezeichner für Modulus	DE	code	..3	M	1	12
6	Wert für Exponent	DE	bin	..512	M	1	65537
7	Bezeichner für Exponent	DE	code	..3	M	1	13

Typ: DEG
Format:
Länge:
Version: 2

Operationsmodus, kodiert

Information über den Operationsmodus für den jeweils verwendeten Kryptoalgorithmus (zur Signaturbildung oder zur Verschlüsselung).

Codierung:

Code	Operationsmodus	Verwendung
2:	Cipher Block Chaining (CBC)	Nur für Verschlüsselung erlaubt (vgl. [HBCI], Kapitel VI.2.2)
16:	ISO 9796-1 (bei RDH),	nicht zugelassen
17:	ISO 9796-2 mit Zufallszahl (bei RDH)	nicht zugelassen
18:	RSASSA-PKCS#1 V1.5 (bei RDH) bzw. RSAES-PKCS#1 V1.5 (bei RAH, RDH)	nicht zugelassen Nur für Verschlüsselung erlaubt
19:	RSASSA-PSS (bei RAH, RDH)	Nur für Signatur erlaubt
999:	Gegenseitig vereinbart (ZZZ)	nicht zugelassen

Typ: DE
Format: code
Länge: ..3
Version: 2

P

Parameter Zertifikatssteuerung

Festlegung der Parameter für die Zertifikatssteuerung.

Nr.	Name	Version	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Priorität der Zertifikatssteuerung	1	DE	code	1	M	1	0, 1
2	Zertifikatsverarbeitung verpflichtend	1	DE	jn	#	M	1	
3	Unterstützte Zertifikatsarten	1	DEG			O	1	
4	Unterstützte Sicherheitsverfahren	3	DEG			O	1..9	

♦ Belegungsrichtlinien

Unterstützte Zertifikatsarten

Ist diese Datenelementgruppe vorhanden, so dürfen nur Zertifikate der dort enthaltenen Zertifikatsarten (D, S, V) belegt und gesendet werden. Wird die DEG weggelassen sind situationsabhängig alle in der Bankensignaturkarte enthaltenen Zertifikate zu senden. Sind auf der Bankensignaturkarte weniger Zertifikate enthalten, als in den unterstützten Zertifikatsarten angegeben, so sind die nicht vorhandenen Zertifikatsarten zu ignorieren.

> Beispiel: Laut Element „Unterstützte Zertifikatsarten“ sind die Zertifikate des Signier- und Chiffrierschlüssels zu übertragen. Es existieren jedoch Kartengenerationen, die nur einen gemeinsamen Signier-/Chiffrierschlüssel enthalten. In diesem Fall ist das Zertifikat des Signierschlüssels im Signaturkopf einzustellen und zu übertragen. Die DEG „Zertifikat“ im Verschlüsselungskopf bleibt hingegen leer. Für die Verschlüsselung der Nachricht auf Kreditinstitutsseite wird dann der öffentliche Schlüssel aus dem Zertifikat des Signierschlüssels verwendet.
> Ausnahme: Handelt es sich um eine nicht signierte Kundennachricht (z. B. optional bei HKEND), so ist das Zertifikat des gemeinsamen Signier-/Chiffrierschlüssels in den Verschlüsselungskopf einzustellen.

Wird in „Unterstützte Zertifikatsarten“ nur das Signierzertifikat (S-Schlüssel) eingestellt, so wird dieses bei Geschäftsvorfällen der Sicherheitsklasse 1 oder 2 auch für die Verschlüsselung der Kreditinstitutsnachricht(en) genutzt. Handelt es sich jedoch um einen Geschäftsvorfall der Sicherheitsklasse 3 oder 4 mit verpflichtender Übertragung des Signaturzertifikats (D-Schlüssel), so ist das Signaturzertifikat in den Signaturkopf und das Verschlüsselungszertifikat in den Verschlüsselungskopf einzustellen.

Unterstützte Sicherheitsverfahren

Ist diese Datenelementgruppe vorhanden, so gelten die Parameter der Zertifikatssteuerung für die dort enthaltenen Sicherheitsprofile. Wird die DEG weggelassen, gilt die Parametrisierung für alle Sicherheitsprofile, die Zertifikate unterstützen.

Es dürfen nur Sicherheitsverfahren belegt werden, die auch Zertifikate unterstützen. Aktuell sind für die in der DEG enthaltenen Elemente also folgende Kombinationen für die Verfahren RAH-7, RDH-6 und RDH-7 möglich:

Sicherheitsverfahren, Code: RAH, RDH

Version des Sicherheitsverfahrens: 6, 7

Typ: DEG
Format:
Länge:
Version: 1

Priorität der Zertifikatssteuerung

Festlegung des Wirkungsbereichs der Zertifikatssteuerung.

Code-Bedeutung :
0 : Zertifikat(e) Senden ist abhängig von der Sicherheitsklasse
1 : Zertifikat(e) Senden ist verpflichtend

Bei „0“ gelten die Angaben für die Zertifikatssteuerung nur für die Dialoginitialisierung und alle damit in Verbindung stehenden Segmente. Die Steuerung, ob ein Zertifikat bei einem signierten Geschäftsvorfall mit gesendet werden soll erfolgt über dessen Sicherheitsklasse.

Bei „1“ wird die Angabe der Sicherheitsklasse bei Geschäftsvorfällen ignoriert und Zertifikatsinformationen müssen für die definierten Sicherheitsprofile immer gesendet werden, wie in HICERS festgelegt.

Bei Geschäftsvorfällen in Segmentversionen, die vor FinTS V3.0 spezifiziert wurden ist die Sicherheitsklasse noch nicht als DE enthalten. Daher kann mit „Priorität der Zertifikatssteuerung“=1 bei Bedarf das generelle Senden von Zertifikaten erzwungen werden.

Typ: DE
Format: code
Länge: 1
Version: 1

R

Rolle des Sicherheitslieferanten, kodiert

Kodierte Information über das Verhältnis desjenigen, der bezüglich der zu sichernden Nachricht die Sicherheit gewährleistet.

Die Wahl ist von der bankfachlichen Auslegung der Signatur, respektive vom vertraglichen Zustand zwischen Kunde und Kreditinstitut abhängig.

Codierung:

1: Der Unterzeichner ist Herausgeber der signierten Nachricht, z. B. Erfasser oder Erstsignatur (ISS)

3: Der Unterzeichner unterstützt den Inhalt der Nachricht, z. B. bei Zweitsignatur (CON)

4: Der Unterzeichner ist Zeuge, aber für den Inhalt der Nachricht nicht verantwortlich, z. B. Übermittler, welcher nicht Erfasser ist (WIT)

Typ:	DE
Format:	code
Länge:	..3
Version:	2

S

Schlüsselart

Information über die Art des Schlüssels.

Beim Sicherheitsverfahren RAH steht die Schlüsselart in engem Zusammenhang mit dem Datenelement "Verwendungszweck für öffentlichen Schlüssel". Die Inhalte beider Datenelemente sind konsistent zu halten.

Codierung:

D: Schlüssel zur Erzeugung digitaler Signaturen (DS-Schlüssel)

S: Signierschlüssel

V: Chiffrierschlüssel

Der DS-Schlüssel steht nur im Zusammenhang mit einer Bankensignaturkarte zur Verfügung.

Im Falle der Bankensignaturkarte ergibt sich folgende Zuordnung zu den Kartenschlüsseln:

- DS-Schlüssel: SK.CH.DS
- Signierschlüssel: SK.CH.AUT
- Chiffrierschlüssel: SK.CH.KE

Typ: DE
Format: code
Länge: 1
Version: 2

Schlüsselname

Verwendeter Schlüsselname in strukturierter Form. Mit dieser Information kann die Referenz auf einen Schlüssel hergestellt werden.

Dabei enthält das DE „Benutzerkennung“ bei Schlüsseln des Kunden die Benutzerkennung, mit der der Kunde eindeutig identifiziert wird. Bei Schlüsseln des Kreditinstituts ist dagegen eine beliebige Kennung einzustellen, die dazu dient, den Kreditinstitutsschlüssel eindeutig zu identifizieren. Diese Kennung darf weder einer anderen gültigen Benutzerkennung des Kreditinstituts noch der Benutzerkennung für den anonymen Zugang entsprechen.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Kreditinstitutskennung	DEG	kik	#	M	1	
2	Benutzerkennung	DE	id	#	M	1	
3	Schlüsselart	DE	code	1	M	1	D, S, V
4	Schlüsselnummer	DE	num	..3	M	1	
5	Schlüsselversion	DE	num	..3	M	1	

Typ: DEG
Format:
Länge:
Version: 3

Schlüsselnummer

Schlüsselnummer des entsprechenden Schlüssels.

Typ: DE
Format: num
Länge: ..3
Version: 1

Schlüsselversion

Versionsnummer des entsprechenden Schlüssels.

Typ: DE
Format: num
Länge: ..3
Version: 1

Segmentkennung

Segmentspezifische Kennung, die jedem Segment bzw. Auftrag zugeordnet ist (z. B. "HKUEB" für "Einzelüberweisung"). Die Angabe hat in Großschreibung zu erfolgen.

Typ: DE
Format: an
Länge: ..6
Version: 1

Falls der Kunde ein Segment mit einer veralteten Versionsnummer einreicht, sollte ihm in einer entsprechenden Warnung rückgemeldet werden, dass sein Kundenprodukt aktualisiert werden sollte.

Segmentkopf

Informationen, die jedem Segment als Kopfteil vorangestellt sind. Im Unterschied zu Nachrichten enthalten Segmente jedoch keinen Abschlussteil, da das Segmentende durch das Segmentende-Zeichen markiert ist.

Im Segmentkopf stehen die Segmentkennung und Segmentversion unabhängig von der HBCI-Version (s. DE HBCI-Version) immer an derselben Stelle, damit ein Segment auch in späteren HBCI-Versionen immer eindeutig als solches identifiziert werden kann.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Segmentkennung	DE	an	..6	M	1	
2	Segmentnummer	DE	num	..3	M	1	>=1
3	Segmentversion	DE	num	..3	M	1	
4	Bezugssegment	DE	num	..3	C	1	>=1 O: Verwendung in Kreditinstitutsnachricht N: Verwendung in Kundennachricht

Typ: DEG
Format:
Länge:
Version: 1

Segmentnummer

Information zur eindeutigen Identifizierung eines Segments innerhalb einer Nachricht. Die Segmente einer Nachricht werden in Einerschritten streng monoton aufsteigend nummeriert. Die Nummerierung beginnt mit 1 im ersten Segment der Nachricht (Nachrichtenkopf).

Typ: DE
Format: num
Länge: ..3
Version: 1

Segmentversion

Versionsnummer zur Dokumentation von Änderungen eines Segmentformats.

Die Segmentversion von administrativen Segmenten (die Segmentart ‘Administration’ bzw. ‘Geschäftsvorfall’ ist bei jeder Segmentbeschreibung angegeben) wird bei jeder Änderung des Segmentformats inkrementiert.

Bei Geschäftsvorfallssegmenten wird die Segmentversion auf logischer Ebene verwaltet, d.h. sie ist für das Auftrags-, das Antwort- und das Parametersegment des Geschäftsvorfalls stets identisch und wird inkrementiert, wenn sich das Format von mindestens einem der drei Segmente ändert.

Dieses Verfahren gilt bei Standardsegmenten einheitlich für alle Kreditinstitute. Bei verbandsindividuellen Segmenten obliegt die Versionssteuerung dem jeweiligen Verband. Der Zeitpunkt der Unterstützung einer neuen Segmentversion kann jedoch zwischen den Verbänden variieren.

Die für die jeweilige HBCI-Version gültige Segmentversion ist bei der jeweiligen Segmentbeschreibung vermerkt.

Falls der Kunde ein Segment mit einer veralteten Versionsnummer einreicht, sollte ihm in einer entsprechenden Warnung rückgemeldet werden, dass sein Kundenprodukt aktualisiert werden sollte.

Typ: DE
Format: num
Länge: ..3
Version: 1

Sicherheitsdatum und -uhrzeit

Zeitstempel, beispielsweise Datum und Uhrzeit des lokalen Rechners, an dem die elektronische Unterschrift geleistet wurde, sowie die Bedeutung des Zeitstempels.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Datum- und Zeitbezeichner, kodiert	DE	code	..3	M	1	1, 6
2	Datum	DE	dat	#	O	1	
3	Uhrzeit	DE	tim	#	C	1	O: ‚Datum’ belegt N: sonst

Typ: DEG
Format:
Länge:
Version: 2

Sicherheitsfunktion, kodiert

Ab FinTS 3.0 existieren beim RAH-Verfahren drei Schlüssel (DS-Schlüssel für Non-Repudiation, Signierschlüssel für Authentication und Chiffrierschlüssel für Verschlüsselung) und somit auch drei Sicherheitsfunktionen (Sicherheitsfunktion 1 bei Verwendung des DS-Schlüssels, Sicherheitsfunktion 2 bei Verwendung des Signierschlüssels und Sicherheitsfunktion 4 bei Verwendung des Chiffrierschlüssels) beim RAH-Verfahren.

Die Sicherheitsfunktion hat ab FinTS 3.0 lediglich informatorischen Wert, da die eigentliche Steuerung über die Sicherheitsprofile und –Klassen erfolgt.

Kodierte Information über die Sicherheitsfunktion, die auf die Nachricht angewendet wird.

Codierung:

1: Non-Repudiation of Origin (NRO)

2: Message Origin Authentication (AUT)

4: Encryption, Verschlüsselung und evtl. Komprimierung (ENC)

Typ: DE
Format: code
Länge: ..3
Version: 2

Sicherheitsidentifikation, Details

Identifikation der im Sicherheitsprozess involvierten Parteien. Dient zur Übermittlung der CID bei kartenbasierten Sicherheitsverfahren bzw. der Kundensystem-ID bei softwarebasierten Verfahren (z. B. Speicherung der Schlüssel in einer Schlüsseldatei).

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Bezeichner für Sicherheitspartei	DE	code	..3	M	1	1, 2
2	CID	DE	bin	..256	C	1	M: Sicherheitsmedium = Chipkarte N: sonst
3	Identifizierung der Partei	DE	id	#	C	1	O: wird vom Kreditinstitut ignoriert

Typ: DEG
Format:
Länge:
Version: 2

Sicherheitskontrollreferenz

Referenzinformation, mit der die Verbindung zwischen Signaturkopf und dazu gehörigem Signaturabschluss hergestellt werden kann. Die Sicherheitskontrollreferenz im Signaturkopf muss mit der entsprechenden Information im Signaturabschluss übereinstimmen.

Typ: DE
Format: an
Länge: ..14
Version: 1

Sicherheitsprofil

Verfahren zur Absicherung der Transaktionen, das zwischen Kunde und Kreditinstitut vereinbar wurde. Das Sicherheitsprofil wird anhand der Kombination der beiden Elemente „Sicherheitsverfahren“ und „Version“ bestimmt (z. B. RDH-9). Für das Sicherheitsverfahren PINTAN ist als Code der Wert PIN und als Version der Wert 1 einzustellen.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Sicherheitsverfahren, Code	DE	code	3	M	1	RAH, PIN
2	Version des Sicherheitsverfahrens	DE	num	..3	M	1	1, 7, 9, 10

Typ: DEG
Format:
Länge:
Version: 1

Sicherheitsreferenznummer

Sicherheitsrelevante Nachrichtenidentifikation (Signatur-ID), welche zur Verhinderung der Doppeleinreichung, respektive Garantie der Nachrichtensequenzintegrität eingesetzt werden kann.

Bei chipkartenbasierten Verfahren ist der Sequenzzähler der Chipkarte einzustellen. Dies ist bei Typ-1 Karten der Wert „EF_SEQ“ in der Application DF_BANKING und bei SECCOS Banken-Signaturkarten der Wert „usage counter“ der beiden Signierschlüssel SK.CH.DS und SK.CH.AUT.

Bei softwarebasierten Verfahren wird die Sicherheitsreferenznummer auf Basis des DE Kundensystem-ID und des DE Benutzerkennung der DEG Schlüsselnamen verwaltet.

Typ: DE
Format: num
Länge: ..16
Version: 1

Sicherheitsverfahren, Code

Code des unterstützten Signatur- bzw. Verschlüsselungsalgorithmus.

Weitere Informationen zu den Verfahren sind Kapitel B.1 zu entnehmen.

Codierung:

RAH: RSA-AES-Hybridverfahren

PIN: PIN/TAN-Verfahren

Typ: DE
Format: code
Länge: 3
Version: 3

Signaturalgorithmus

Angaben zum kryptographischen Algorithmus, zu seinem Operationsmodus, so wie zu dessen Einsatz, in diesem Fall für die Signaturbildung über RAH.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Verwendung des Signaturalgorithmus, kodiert	DE	code	..3	M	1	6
2	Signaturalgorithmus, kodiert	DE	code	..3	M	1	1, 10
3	Operationsmodus, kodiert	DE	code	..3	M	1	19

Typ: DEG
Format:
Länge:
Version: 2

Signaturalgorithmus, kodiert

Kodierte Information über den Signaturalgorithmus.

Codierung:

1: nicht zugelassen

10: RSA-Algorithmus (bei RAH)

Typ: DE
Format: code
Länge: ..3
Version: 2

Sperrenkennzeichen

Information zur Begründung der Sperrung.

Codierung:

1: Schlüssel des Zertifikatseigentümers kompromittiert

501: Zertifikat ungültig wegen Verdacht auf Kompromittierung

999: gesperrt aus sonstigen Gründen

Typ: DE
Format: code
Länge: ..3
Version: 2

U

Unterstützte Zertifikatsarten

Diese DEG enthält die relevanten Zertifikatsarten für die Zertifikatssteuerung. Es werden nur die dort definierten Zertifikatsarten (D, S, V) berücksichtigt.

Nr.	Name	Version	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Zertifikat D-Schlüssel vorhanden	1	DE	jn	#	M	1	
2	Zertifikat S-Schlüssel vorhanden	1	DE	jn	#	M	1	
3	Zertifikat V-Schlüssel vorhanden	1	DE	jn	#	M	1	

Typ: DEG
Format:
Länge:
Version: 1

Uhrzeit

Uhrzeit eines Ereignisses (meist zusammen mit „Datum“ verwendet).

Typ: DE
Format: tim
Länge: #
Version: 1

V

Validierungsresultat

Elektronische Signatur, die zur Validierung berechnet wurde.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Verfahren Benutzer

Information über das Benutzer-Verfahren, die beim öffentlichen Schlüssel angegeben wird.

Es ist nur der folgende Wert zugelassen:

10: RSA-Verfahren

Typ: DE
Format: code
Länge: ..3
Version: 2

Verschlüsselungsalgorithmus

Angaben zum kryptographischen Algorithmus, zu seinem Operationsmodus, so wie zu dessen Einsatz, in diesem Fall für die Nachrichtenverschlüsselung.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Verwendung des Verschlüsselungsalgorithmus, kodiert	DE	code	..3	M	1	2
2	Operationsmodus, kodiert	DE	code	..3	M	1	2, 18, 19
3	Verschlüsselungsalgorithmus, kodiert	DE	code	..3	M	1	13, 14
4	Wert des Algorithmusparameters, Schlüssel	DE	bin	..512	M	1	
5	Bezeichner für Algorithmusparameter, Schlüssel	DE	code	..3	M	1	6
6	Bezeichner für Algorithmusparameter, IV	DE	code	..3	M	1	1
7	Wert des Algorithmusparameters, IV	DE	bin	..512	O	1	

Typ: DEG
Format:
Länge:
Version: 2

Verschlüsselungsalgorithmus, kodiert

Kodierte Information über den verwendeten Verschlüsselungsalgorithmus.

Codierung:

13: 2-Key-Triple-DES (nicht zugelassen)

14: AES-256 [AES]

Typ: DE
Format: code
Länge: ..3
Version: 3

Version der Filterfunktion

Version der Filterfunktion.

Typ: DE
Format: num
Länge: ..3
Version: 1

Version des Sicherheitsverfahrens

Version des unterstützten Sicherheitsverfahrens (s. „Sicherheitsverfahren, Code“).

In Kombination mit dem Sicherheitsverfahren RAH sind die folgenden Versionen gültig:

Version	Signaturverfahren	Schlüssellänge (bit)	Hashverfahren	Schlüsselart*
7	PKCS#1 PSS	..2048	SHA-256	D, S, V
9	PKCS#1 PSS	..2048	SHA-256	S, V
10	PKCS#1 PSS	..2048	SHA-256	S, V

* s. Element „Schlüsselart“

Andere als die genannten Profile sind nicht zulässig.

Um Multibankfähigkeit zu gewährleisten, ist die Unterstützung des Verfahrens RAH-9 kunden- und kreditinstitutsseitig verpflichtend.

Typ: DE
Format: num
Länge: ..3
Version: 2

Verwendung des Hashalgorithmus, kodiert

Kodierte Information über die Verwendung des Hashalgorithmus.

Im Zusammenhang mit Hash-Funktionen ist derzeit nur folgender Wert möglich:

Codierung:

1: Owner Hashing (OHA)

Typ: DE
Format: code
Länge: ..3
Version: 2

Verwendung des Signaturalgorithmus, kodiert

Kodierte Information über die Verwendung des Signaturalgorithmus.

Im Zusammenhang mit Signaturbildung ist derzeit nur folgender Wert möglich:

Codierung:

6: Owner Signing (OSG)

Typ: DE
Format: code
Länge: ..3
Version: 2

Verwendung des Verschlüsselungsalgorithmus, kodiert

Kodierte Information über die Verwendung des Verschlüsselungsalgorithmus.

Im Zusammenhang mit der Verschlüsselung sind derzeit folgende Werte möglich:

Codierung:

2: Owner Symmetric (OSY)

Typ: DE
Format: code
Länge: ..3
Version: 2

Verwendungszweck für öffentlichen Schlüssel

Kodierte Information über die Verwendung des öffentlichen Schlüssels. Diese Information muss konsistent zur Schlüsselart gehalten werden.

Codierung:

5: Owner Ciphering (Chiffrierschlüssel)

6: Owner Signing (Signierschlüssel)

Typ: DE
Format: code
Länge: ..3
Version: 2

W

Wert des Algorithmusparameters, IV

Initialisierungswert für den kryptographischen Algorithmusparameter. Zur Zeit ist die Angabe eines Wertes nicht zulässig; es wird dafür folgender Initialisierungswert als Default verwendet: X'00 00 00 00 00 00 00 00'

In einer zukünftigen Version kann ein abweichender Initialisierungswert definiert werden.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Wert des Algorithmusparameters, Schlüssel

Verschlüsselter Nachrichtenschlüssel für den kryptographischen Algorithmusparameter.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Wert des Hashalgorithmusparameters

Initialisierungswert für den Hashalgorithmusparameter. Zur Zeit ist die Angabe eines Wertes nicht zulässig.

In einer zukünftigen Version kann ein abweichender Initialisierungswert definiert werden.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Wert für Exponent

Exponent des öffentlichen Schlüssels (z. Zt. 65537). Die Kürzung um führende 0-Bytes ist empfehlenswert, aber nicht verbindlich.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Wert für Modulus

Modulus des öffentlichen Schlüssels. Die Kürzung um führende 0-Bytes ist empfehlenswert, aber nicht verbindlich.

Typ: DE
Format: bin
Länge: ..512
Version: 1

Z

Zertifikat

Zertifikat eines öffentlichen Schlüssels.

Da Zertifikate Informationen beinhalten, die auch in den HBCI-Formaten enthalten sind (z. B. Zertifikatsreferenz respektive Schlüsselnamen), können Daten redundant vorkommen. Diese müssen dann auf Konsistenz überprüft werden. Bei Unstimmigkeiten hat das Zertifikat Vorrang.

Nr.	Name	Typ	Format	Länge	Status	Anzahl	Restriktionen
1	Zertifikatstyp	DE	code	1	M	1	1, 2, 3
2	Zertifikatsinhalt	DE	bin	.. 4096	M	1	

Typ: DEG
Format:
Länge:
Version: 2

Zertifikat D-Schlüssel vorhanden

Information, ob ein Zertifikat für den D-Schlüssel (Signaturschlüssel) vorhanden ist.

Typ: DE
Format: jn
Länge: #
Version: 1

Zertifikat S-Schlüssel vorhanden

Information, ob ein Zertifikat für den S-Schlüssel (Signierschlüssel) vorhanden ist.

Typ: DE
Format: jn
Länge: #
Version: 1

Zertifikat V-Schlüssel vorhanden

Information, ob ein Zertifikat für den V-Schlüssel (Chiffrierschlüssel) vorhanden ist.

Typ: DE
Format: jn
Länge: #
Version: 1

Zertifikatsinhalt

Transparenter Inhalt eines Zertifikats.

Bei der Bankensignaturkarte handelt es sich hier um

- das Signaturzertifikat C_X509.CH.DS,

- das CSA-(KE-)Zertifikat C_X509.CH.AUTC/S[&KE]

- und das KE-Zertifikat C_X509.CH.KE

Typ: DE
Format: bin
Länge: ..4096
Version: 1

Zertifikatstyp

Information über Aufbau und Inhalt eines Zertifikats.

Codierung:

1: ZKA

2: UN/EDIFACT

3: X.509 v3 (gemäß [ISIS/MTT])

Typ: DE
Format: code
Länge: 1
Version: 2

Zertifikatsverarbeitung verpflichtend

Festlegung, ob vom Kreditinstitut übertragene Zertifikate vom Kundensystem verpflichtend zu prüfen und zu verwenden sind. In diesem Fall werden über den FinTS-Key-Management-Geschäftsvorfall HKISA keine neuen Schlüssel des Kreditinstituts zur Verfügung gestellt.

Derzeit ist nur der Wert „n“ zugelassen.

Typ: DE
Format: jn
Länge: #
Version: 1

E. ANLAGEN

E.1 Übersicht der Segmente

Nr.	Segmentname	Kennung	Sender[1]	Version
1	Anforderung eines öffentlichen Schlüssels	HKISA	K	3
2	Bestätigung der Schlüsselsperrung	HISSP	I	3
3	Schlüsseländerung	HKSAK	K	3
4	Schlüsselsperrung	HKSSP	K	3
5	Signaturkopf	HNSHK	K/I	4
6	Übermittlung eines öffentlichen Schlüssels	HIISA	I	3
7	Verschlüsselte Daten	HNVSD	K/I	1
8	Verschlüsselungskopf	HNVSK	K/I	3

[1] K: Kunde, I: Kreditinstitut