

Bücherhallen Hamburg

Informationen zum Codierschema



Inhaltsverzeichnis

1. Einführung.....	2
2. Verwendete Abkürzungen.....	2
3. Beschreibung des DESFire Systemkonzepts.....	3
4. Basisdefinition des Organisationslevel.....	4
5. Dateistruktur der DESFire-Applikationen.....	4
6. Schlüsselmanagement.....	5
7. Card-Settings & Applikationen.....	7
7.1 Card Settings.....	7
7.2 Applikationen.....	7
8. Beschreibung der Applikationen auf dem Chip.....	8
8.1 AID Global / „Standard“ AID 0x F585D4.....	9
8.1.1 Schlüssel Global.....	9
8.1.2 Dateibeschreibung Globaldaten.....	10
8.1.3 Berechnung der 14-stelligen Kundennummer.....	11
9. Kontaktdaten für Rückfragen.....	12

1. Einführung

In dieser Dokumentation werden die **DESFire EV1/2** Applikationsdefinitionen für die „Bücherhallen Hamburg“ beschrieben. Sämtliche Informationen wurden mit dem System-Betreiber erarbeitet.

Alle Änderungen der Applikationsdefinitionen müssen in diesem Dokument niedergeschrieben werden. Informationen, die an Lieferanten / Kartenhersteller oder Automatenzulieferer zur Kodierung dieser Applikationen bzw. Ausweiskarten weitergegeben werden, liegen alleine in der Verantwortung des Systembetreibers.

2. Verwendete Abkürzungen

Abkürzung	Beschreibung
AID	Applikations Identifikation (3 Byte eindeutige registrierte Kennung)
ASCII (UTF-8)	American Standard Code for Information Interchange
PICC	Proximity Integrated Circuit Card
ZK	Zutrittskontrolle
ZE	Zeiterfassung
KANT	Kantine
AUT	Automaten
OL	Organisationslevel (OL1 bis OL3)
VAR	Variable Werte für jede Applikation
HEX	Werte als Hexadezimal Zahl
BCD	Binary Coded Decimal (Binärkodierte Dezimal Zahl)
XX	Werte aus System, oder als Eingabe
??	Werte nicht definiert
INT1	Die Daten sind hexadezimal codiert und werden „high endian“ codiert. Der Wertebereich ist 0 .. 255 (2^8)
INT2	Die Daten sind hexadezimal codiert und werden „high endian“ codiert. Der Wertebereich ist 0 .. 65.535 (2^{16})
INT3	Die Daten sind hexadezimal codiert und werden „high endian“ codiert. Der Wertebereich ist 0 .. 16.777.215 (2^{24})
INT4	Die Daten sind hexadezimal codiert und werden „high endian“ codiert. Der Wertebereich ist 0 .. 4.294.967.295 (2^{32})
INT5	Die Daten sind hexadezimal codiert und werden „high endian“ codiert. Der Wertebereich ist 0 .. 2^{40}
Amount(N)	Anzahl (0 .. N) Werteinheiten. Die Daten sind hexadezimal und werden „high endian“ codiert.
UID	Unique Identifier - Eindeutige Seriennummer des RFID-Chip in der Karte
UUID	Universally Unique Identifier (Universal generierte eindeutige Seriennummer)
CMK	Card Master Key
APK / AMK	Application Master Key
OSS / OSO	Open Security Standards / OSS STANDARD OFFLINE.

PrintableString ist eine Kette von druckbaren Zeichen. Die Zeichen werden durch ASCII (UTF-8) Codes repräsentiert. Pro Zeichen werden 1 Byte verwendet.

OctetString(N) Kette von Octets ohne spezielles Format. Im Englischen ist die Bezeichnung ByteArray üblich. Die Anzahl der Bytes wird als Parameter N in der Form OctetString(N) angegeben

3. Beschreibung des DESFire Systemkonzepts

Als Standard Karte wird ein Chip **DESFire** EV1/2 / 4 kByte von NXP verwendet. Dieser Chip wird in ISO PVC Ausweiskarten eingebettet. Um jede Ausweiskarte eindeutig identifizieren zu können, ist in jedem Chip eine einmalige Seriennummer fest und unveränderbar eingeprägt, die Teil des Sicherheitssystems ist. Genaue Angaben finden Sie in der **DESFire** Dokumentation.

Aus Dokumentation NXP DESFire EV1:

- Fully ISO 14443A compliant, up to part 4
- Unique 7 Byte Serial Number ISO cascade level 2 / Random UID 4 Byte
- 4 KByte EEPROM
- Fast Data Transfer, up to 848 Kbit/s
- Mutual Three Pass Authentication
- DES/3DES and 3KDES/AES Data Encryption
- Data Authenticity by 16 byte AES
- Flexible File System
- Up to 28 Applications per card
- Up to 32 Files per Application
- Up to 14 keys per Application, with key Versioning
- Automatic anti-tear mechanism for all available file types

Funktionsprinzip:

Aus Datenschutzgründen wird der Zugriff der einzelnen Daten nach dem „need to know“ Prinzip aufgebaut. Das System welches Daten auf dem DESFire lesen und eventuell auch bearbeiten darf, bekommt nur die notwendigen Informationen zur Verfügung gestellt welches es benötigt. Durch die einzelnen Applikationen (Container) kann dies gewährleistet werden. In diesen Containern werden die Daten vorgehalten welche das System benötigt, kein System wird mit Daten von weiteren Containern versorgt. Somit können auch unterschiedliche Kartengenerationen und Gruppen erstellt werden, welche nur die Container beinhalten die benötigt werden, oder welches das Benutzer-Profil erlaubt. Der DESFire Teil selbst wird mit einem Masterkey verwaltet, welcher die Grundstruktur auf dem Chip organisiert, jedoch keine Zugriffsrechte auf die eigentlichen Daten in den Containern erlaubt.

Jeder Container ist isoliert, da dieser ein eigenes Zugriffsrechtmanagement besitzt, dazu zählt ein Applikationsmasterkey als Administrator und mehrere Lese- und/oder Schreibschlüssel, welche für verschiedene Datenzugriffe eingesetzt werden.

Ein Zutrittssystem, zum Beispiel, besitzt nur den aktuellen Leseschlüssel für einen bestimmten Container und dort nur für ein bestimmtes Datenpaket. Aus Sicherheitsgründen können auch die Schlüssel des Containers gewechselt werden, somit haben nur die Leser Zugriff auf die Daten, wenn sie auch den neuen „aktuellen“ Schlüssel besitzen. Je nach System kann dieser Schlüssel „Online“ zur Verfügung gestellt werden, oder in einem sicheren Speicherbereich wiederum verschlüsselt im Leser abgelegt werden.

4. Basisdefinition des Organisationslevel

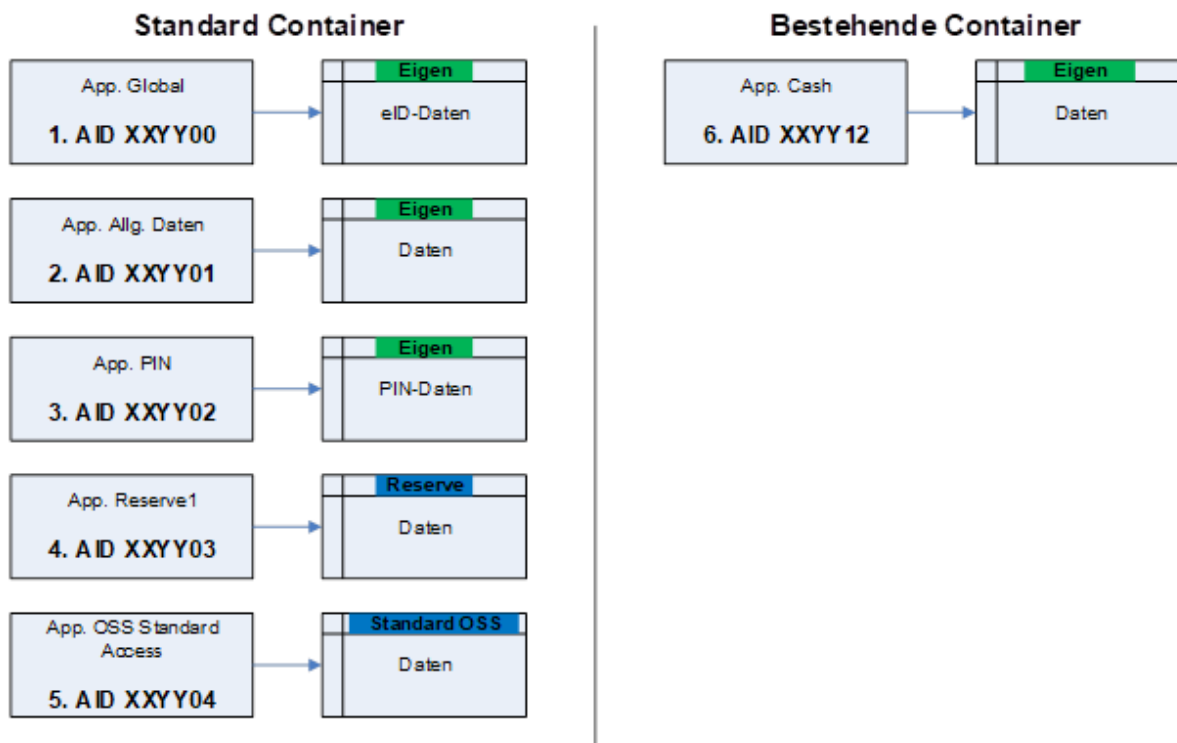
Version 1.0 mit NXP Registrierung

Level	Beschreibung	Werte
OL 1	MAD	XX / (Registrierung NXP)
OL 2	MAD	YY / (Registrierung NXP)
OL 3	Applikation / Level	00 - FF

Um die möglichen Applikation übersichtlich zu strukturieren, werden die AID's in verschiedene Blöcke (Standard, bestehende, etc.) eingeteilt. Die einzelnen Schlüssel bekommen den jeweiligen Applikationsnamen, so ist es möglich bei Weitergabe oder Deaktivierung mit schlüssigen Namen zu arbeiten.

5. Dateistruktur der DESFire-Applikationen

Bsp. Filestruktur MA-A usweis
Applikationen Version 1



6. Schlüsselmanagement

Der PICC_MASTERKEY ist der **Card Masterkey (CMK)** und ermöglicht das Erstellen von Applikationen auf der Karte. [MF3ICD8]

Um bestimmte Daten auf der Karte lesen zu können, muss der Leser sich zuvor mit dem Card Masterkey authentifizieren.

Dieser Card Masterkey ist der Zentralschlüssel der Chipcodierung, mit diesem Schlüssel wird am Ende der Initialisierung (aufbringen der Grundstruktur) die Karte „abgeschlossen“. Mit diesem Schlüssel kann die Datenstruktur des Chips geändert und gelöscht, jedoch keine Daten gelesen werden. Dieser Schlüssel wird nur im Personalisierungssystem gespeichert und wird sonst nie ausgegeben.

Auf der Applikationsebene (Anwendungen) befindet sich nun der **Application Masterkey (AMK / APK)**. Diese Key schützt die Einstellungen der Applikation selbst. Soll an diesen Einstellungen etwas geändert werden ist dieser Application Masterkey erforderlich. Selbst mit dem eigentlichen Card Masterkey können diese Änderungen nicht durchgeführt werden. Unter dem Application Masterkey werden nun die eigentlichen Lese- und Schreibschlüssel der Applikation selbst organisiert. Jede Applikation kann bis zu 14 verschiedene Schlüssel besitzen, welche für bestimmte Rechte in der Applikation und deren Dateien bestimmt werden können. Im Standardfall werden zwei unterschiedliche Lese-schlüssel und je nach dem bis zu zwei verschiedene Schreibschlüssel verwendet.

Jeder dieser Schlüssel ist ein AES 128 Bit Schlüssel und benötigt 16 Byte Speicherplatz, diese Schlüssel müssen mit in der Berechnung der Speicherbelegung berücksichtigt werden. In der vorhandenen Dokumentation werden diese einzelnen Schlüssel mit Namen der Applikation und Nummern versehen, damit wir diese auseinander halten können.

In den entsprechenden Lesern der Installation einer Anwendung werden nur die benötigten Schlüssel „nur Lesen“ oder auch „Schreiben und Lesen“ vorgehalten. Die Applikationen besitzen mehrere Schlüssel, wobei immer nur eine Schlüsselgeneration gültig ist. Um nun die Installation noch sicherer zu gestalten, können diese Schlüsselgenerationen auf der Karte gewechselt werden und ab diesem Zeitpunkt können nur Leser mit diesem neuen Schlüsselpaar die Daten aus der Applikation auf der Karte lesen. Typischerweise werden diese neuen Schlüssel erst zu diesem Zeitpunkt an die zuständigen Leser ausgegeben. Diese Ausgabe der Schlüssel an die Leser kann auf verschiedene Art und Weise erfolgen. Entweder „Online“ per Datenverbindung oder „offline“ per eigener Systemkarte usw.

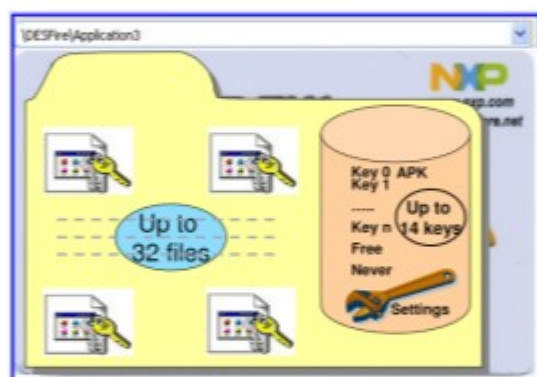
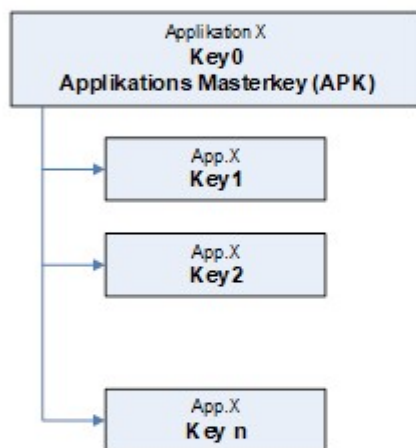
Das einzige System welches alle Schlüssel besitzt ist das Personalisierungssystem selbst. Ist die Karte einmal draußen im Feld, können die Einstellungen nur durch Leser geändert werden, welche im Besitz die erforderlichen Application Masterkeys sind.

Ein Beispiel: Das Schlüsselpaar einer bestehenden Applikation soll geändert werden. Dabei werden nicht die Schlüssel auf der Karte geändert, sondern die Einstellungen auf der Karte welche Schlüsselnummer für Lesen und Schreiben neu festgelegt. Die neuen Schlüssel selbst sind durch die Personalisierung schon auf der Karte vorhanden. Um diese Einstellung auf der Karte ändern zu können ist bei der „EV1“ Generation noch der Application Masterkey erforderlich (ab „EV2“ Generation ist dies nicht mehr notwendig).

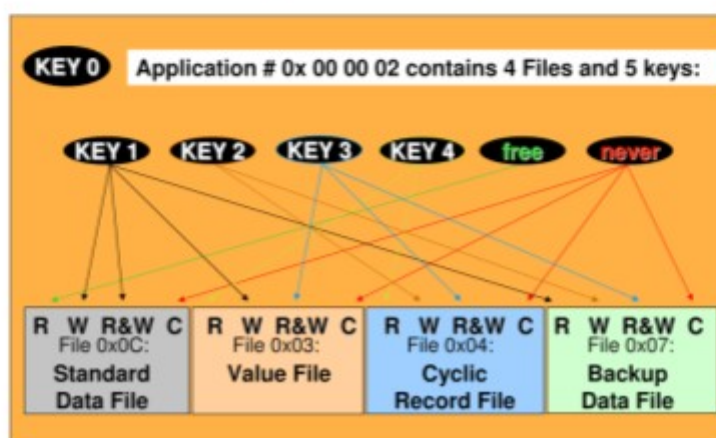
Für die Aktionen zur Änderung der gültigen Schlüssel können separate Stationen verwendet werden oder sie werden in die Terminals integriert. Da diese Stationen „Online“ im Netz betrieben werden, kann der zuständige „Application Masterkey“ (**AMK/APK**) über diese Verbindung bereitgestellt werden, um die Änderungen in der Applikation durchzuführen.

Keys truktur DESFire-Chipkarte Applikationen Version 1

Chip
Card Masterkey (CMK)



Bsp. Schlüsselorganisation auf Applikations ebene



7. Card-Settings & Applikationen

7.1 Card Settings

Für den Card Masterkey (CMK / PICC_MASTERKEY) sind die Einstellungen gemäß der folgenden Tabelle vorzunehmen.

Card Settings		
Konfiguration änderbar		Ja
Anlegen / Löschen von Applikationen ohne Authentisierung mit PICC_MASTERKEY		Nein
Auslesen der KeySettings und der AID's ohne Authentisierung mit PICC_MASTERKEY		Ja
Ändern des PICC_MASTERKEY möglich		Ja

7.2 Applikationen

Die folgenden Tabellen zeigen die Struktur der Applikationen und der darin enthaltenen Dateien und Schlüssel auf der Chipkarte.

Struktur „Standard“	Dateinummer SchlüsselNr.	Datei- / Verschlüsselung
Applikation Global F585D4	0x00	Globaldaten1 / AES 128
	0x0	Schlüssel K_APP_GLOBAL0
	0x1	Schlüssel K_LESEN_GLOBALDATEN1
	0x2	Schlüssel K_SCHREIBEN_GLOBALDATEN1

8. Beschreibung der Applikationen auf dem Chip

Applikationen / Standards

Die Applikationen auf der Karte beinhaltet Daten, die direkt mit der Karte verknüpft sind und als Verwaltungsinformationen des Kartenherausgebers dienen. Es werden knapp 20 Applikationen vorbelegt und jeweils mit einem Application Masterkey, Leseschlüsseln und Schreibschlüssel versehen.

Eine Applikation ist wie eine eigenständige Karte zu sehen, es werden alle relevanten Informationen für die Systeme in der Applikation abgelegt. Kein System kann die Berechtigungen in den weiteren Applikation lesen oder schreiben. So ist der Datenschutz gewährleistet, und es können keine Nutzerprofile erstellt werden, da die Systeme jeweils mit unterschiedlichen Daten aus ihrer Applikation arbeiten.

Jede **Applikation** enthält eine oder mehrere Dateien, welche mit unterschiedlichen Schlüsseln gelesen werden können. Jede der Dateien enthält die UID des Chip selbst und ein Ablaufdatum der Datei.

Global existiert darüber hinaus eine frei lesbare Datei mit der und dem Ablaufdatum der Karte. Diese Datei kann erst bei der Karten-Personalisierung geschrieben werden.

Zusätzlich Information können in jeder Applikation in weiteren FileID's abgelegt werden, Beispiel Zutritt usw., grundsätzlich ist aber der Aufbau der 20 Applikationen (mit Ausnahme der Global-Applikation) derselbe.

Durch die generelle Überprüfung des Ablaufdatums können ungültige Karten sofort abgewiesen werden. Der Karteninhaber wird so gezwungen sich wieder an eine kartenausgebende Stelle zu wenden. So kann der Life Cycle einer Karte zu bestimmten Zeitpunkten neu bestimmt werden.

Die Applikation Global beinhaltet relevante allgemeine Daten, welche sich auf die App-Inst. Nr, UUID und Ablaufdatum beschränken.

Sperrliste: Auf Basis diese Konzeptes ist es möglich eine Sperrliste für alle DESFire-Applikationen anhand der UID (Seriennummer) der Kartenchips zu führen. Jeder Leser müsste dann diese Sperrliste abarbeiten bevor eine Karte als „Gültig“ betrachtet werden kann. Diese Sperrliste wird aus dem Kartenmanagementprogramm generiert und den einzelnen Systemen zur Verfügung gestellt. Die UID selbst ist anonym, es werden keine persönlichen Daten ausgegeben, es geht nur um die Sperrung der einzelnen Karte unabhängig von Nutzer oder Besitzer der Karte.

Diese Liste müsste auch von Offline-Systeme als täglicher Import verarbeitet werden!

8.1 AID Global / „Standard“ AID 0x F585D4

Die Applikation Global besitzt die AID 0x F585D4

8.1.1 Schlüssel Global

Die Applikation Global beinhaltet folgenden Schlüssel: Der K_APP_GLOBAL0 ist der Application Masterkey. Er ermöglicht Struktur- und Schlüsseländerungen innerhalb der Applikation.

Die Applikation wird AES verschlüsselt.

Schlüsselname	Schlüsselnummer	Beschreibung
K_APP_GLOBAL0 „ApplicationMasterKey“	0x0	Nach Authentisierung mit diesem Schlüssel sind Änderungen an der Applikation (Dateistruktur usw.) möglich.
K_LESEN_GLOBALDATEN1	0x1	Nach Authentisierung mit diesem Schlüssel ist der lesende Zugriff auf die Daten möglich.
K_SCHREIBEN_GLOBALDATEN1	0x2	Nach Authentisierung mit diesem Schlüssel ist der schreibende Zugriff auf die Daten möglich.

Schlüssel sind nach Authentifizierung mit K_APP_GLOBAL0 änderbar		Ja
Konfiguration änderbar		Ja
Anlegen / Löschen von Dateien ohne Authentisierung mit K_APP_GLOBAL0		Nein
Auslesen der Verzeichnisstruktur ohne Authentisierung mit K_APP_GLOBAL0		Nein
Ändern des K_APP_GLOBAL0 möglich		Ja

8.1.2 Dateibeschreibung Globaldaten

In diesem Standard Data File werden weitere Card-Daten abgelegt. Jedoch ist diese Applikation mit Schlüsseln lesegeschützt. Unterschiedliche AppVersionen können mit einem Byte unterschieden werden. Zusätzlich wird die Kommunikation verschlüsselt.

Byte	Wert	Kodiert	Wertbereich	Beschreibung
0	01	HEX	00 - FF	AppVersion
1	D	ASCII	fix	Bibliothekssiegel
2	E	ASCII	fix	Bibliothekssiegel
3	-	ASCII	fix	Bibliothekssiegel
4	H	ASCII	fix	Bibliothekssiegel
5	1	ASCII	fix	Bibliothekssiegel
6	0	ASCII	fix	Bibliothekssiegel
7		ASCII	00 - FF	Kundennummer Zeichen 1
8		ASCII	00 - FF	Kundennummer Zeichen 2
9		ASCII	00 - FF	Kundennummer Zeichen 3
10		ASCII	00 - FF	Kundennummer Zeichen 4
11		ASCII	00 - FF	Kundennummer Zeichen 5
12		ASCII	00 - FF	Kundennummer Zeichen 6
13		ASCII	00 - FF	Kundennummer Zeichen 7
14		ASCII	00 - FF	Kundennummer Zeichen 8
15		ASCII	00 - FF	Kundennummer Zeichen 9
16		ASCII	00 - FF	Kundennummer Zeichen 10
17		ASCII	00 - FF	Kundennummer Zeichen 11
18		ASCII	00 - FF	Kundennummer Zeichen 12
19		ASCII	00 - FF	Kundennummer Zeichen 13
20		ASCII	00 - FF	Kundennummer Zeichen 14
21	0	HEX	00 - FF	0x00 String-Terminierung für zukünftige Nutzung
22	0	HEX	00 - FF	0x00 String-Terminierung für zukünftige Nutzung
...				
31	0	HEX	00 - FF	0x00 String-Terminierung für zukünftige Nutzung

Datenfelder:

- Bibliothekssiegel: „DE-H10“
- Kundennummer: 16-stellig alphanumerisch (derzeit 14 Stellen verwendet)

Der Nummernkreis für die 250.000 Kundennummern (Druckformat) lautet:

→ A80 000 000 6 bis A80 249 999 6

8.1.3 Berechnung der 14-stelligen Kundennummer

Die zu codierende Kundennummer wurde 1-zu-1 aus der bisherigen Magnetstreifen-Codierung übernommen. Dadurch wird sichergestellt, dass alle bisherigen Anwendung weiterhin die selbe Nummer bzw. das selbe Nummernformat erhalten und somit ein problemloser Übergang vom Magnetstreifen auf RFID-Codierung stattfindet.

Die Kundennummer hat 14 Zeichen, die sich wie folgt aufteilen.

10-stelliger Basiswert				4-steliger Zusatz	
fest	fest	fortlaufend	fortlaufend	Modulo	fest
00	80	###	###	##	03

Modulo → Prüfsummenberechnung (Beispiel):

Jede Stelle (Ziffer) des 10-stelligen Basiswertes erhält eine Gewichtung gemäß folgender Tabelle:

Stelle	0	0	8	0	1	2	3	4	5	6
Gewichtung	3	5	9	7	8	4	6	3	5	2

Jede Stelle wird mit der dazugehörigen Gewichtung multipliziert und dann die Summe gebildet:

Stelle	0	0	8	0	1	2	3	4	5	6	
Gewichtung	3	5	9	7	8	4	6	3	5	2	
Multiplikation	0	0	72	0	8	8	18	12	25	12	155

Die Summe wird durch 11 geteilt und der Rest ergibt die 2-stellige Modulo / Prüfziffer

155 dividiert durch 11 ergibt 14 mit einem Rest von 1 → Prüfziffer „01“

14-stellige Kundennummer: 00 80 123 456 01 03

Für die Codierung der 250.000 Chipkarten würden sich somit folgende Datensätze ergeben:

lfd. Nr	Basiswert	Aufdruck	Codierung
1	0080000000	A80 000 000 6	008000000000603
2	0080000001	A80 000 001 8	00800000010803
3	0080000002	A80 000 002 X	00800000021003
...			
249998	0080249998	A80 249 998 4	00802499980403
249999	0080249999	A80 249 999 6	00802499990603

Achtung: In der Prüfziffer der Bedruckung wird für einen Wert 10 das Zeichen „X“ verwendet.

9. Kontaktdaten für Rückfragen

Bücherhallen Hamburg

Bereichsleitung IT und Organisation

Carolin Rohrßen

Hühnerposten 1 | 20097 Hamburg

Tel.: +49 40 42606-118

E-Mail: Carolin.Rohrssen@buecherhallen.de

Bücherhallen Hamburg

Bereichs IT und Organisation | Projektmanagement

Angela Rustemeier

Hühnerposten 1 | 20097 Hamburg

Tel.: +49 40 42606-222

E-Mail: Angela.Rustemeier@buecherhallen.de