Vertrauenswürdige Kommunikation im elektronischen Zahlungsverkehr

Ein formales Rollen- und Aufgabenbasiertes Sicherheitsmodell für Anwendungen mit multifunktionalen Chipkarten

Dissertation

**zur Erlangung des Doktorgrades
am Fachbereich Informatik
der Universität Hamburg**

vorgelegt von

Kathrin Schier

Juni 1999

Universität Hamburg, Fachbereich Informatik

Genehmigt vom Fachbereich Informatik der Universität Hamburg auf Antrag von

Prof. Dr. Klaus Brunnstein, Universität Hamburg
Prof. Dr. Reinhard Posch, Technische Universität Graz
Dr. Hans-Joachim Mück, Universität Hamburg

Hamburg, den 21. Juni 1999
(Datum der Disputation)

Prof. Dr. L. Dreschler-Fischer
(Dekanin des Fachbereichs Informatik)

Für Hartmut

Vorwort

Als ich im September 1994 am Fachbereich Informatik der Universität Hamburg meine Tätigkeit als wissenschaftliche Mitarbeiterin begann, war dort die Thematik der Chipkarten und deren Anwendungen nicht sehr verbreitet. Im Laufe der Zeit wuchs jedoch das Interesse an diesem Thema und damit auch mein Wunsch, eine anwendungsbezogene Dissertation zum Thema Chipkartensicherheit anzufertigen. Allen, die mich bei dieser Arbeit unterstützt haben, möchte ich sehr danken.

Ich möchte meinem Betreuer, Herrn Prof. Dr. Klaus Brunnstein, herzlich für eine ausgezeichnete Betreuung danken. Er hat mich immer in meinen Ideen bestärkt und mit seinen konstruktiven Ratschlägen meinen Blick korrigiert und erweitert. Er hat mich stets unterstützt, meine Gedanken auf zahlreichen nationalen und internationalen Konferenzen vorzustellen und zu diskutieren. Ein Höhepunkt war die Präsentation meines Rollen- und Aufgabenbasierten Sicherheitsmodells (R&A-Modell) auf der IFIP World Conference - Security 1998 in Wien und Budapest und die Präsentation einer Anwendung des R&A-Modells auf der Annual Computer Security Applications Conference 1998. Auf beiden Konferenzen hatte ich die Gelegenheit, meine Ideen mit vielen internationalen Fachleuten zu diskutieren. An dieser Stelle möchte ich Herrn Prof. Dr. Reinhard Posch und Herrn Prof. Dr. Ravi Sandhu sehr herzlich danken.

Meinen Kollegen und ehemaligen Mitarbeitern des Arbeitsbereichs danke ich für eine gute Zusammenarbeit und ein nettes Arbeitsklima. Mein Kollege Klaus-Peter Kossakowski hat mich in zahllosen Diskussionen, nicht nur in der schweren Anfangsphase, fachlich und moralisch sehr gestützt. Ich danke Berndt Farwer, Dr. Daniel Moldt, Jürgen Dethloff und Dr. Rüdiger Grimm für interessante Diskussionen über ihre Fachgebiete. Großer Dank gilt den Mitarbeitern des Rechenzentrums und der Bibliothek für ihre Unterstützung.

Meiner Familie und meinen Freunden danke ich für ihr Verständnis für meine häufige Abwesenheit in dieser Zeit. Meinem Vater gilt besonderer Dank für ein sorgfältiges Korrekturlesen und viele hilfreiche Anmerkungen.

Mein größter Dank gilt jedoch Hartmut, ohne den diese Arbeit in dieser Form nie zustande gekommen wäre. Er hat mich durch diese schwere Zeit begleitet und mir die Kraft und den Mut gegeben, eine solche Arbeit anzufangen und vor allem auch zu beenden.

Ich hoffe, mit dieser Arbeit einen Beitrag für sichere und vertrauenswürdige Kommunikation im elektronischen Zahlungsverkehr geleistet zu haben und wünsche mir einen praktischen Einsatz des R&A-Modells.

Hamburg, im Februar 1999

Kathrin Schier

Inhalt

Abbildungen VII

Tabellen IX

1 Einleitung 11

1.1 Motivation 11
1.2 Übersicht über die Arbeit 13

2 Elektronischer Zahlungsverkehr 17

2.1 Begriffsbestimmung und Bedeutung des elektronischen Zahlungsverkehrs 18
2.2 Grundlegende Sicherheitsanforderungen an elektronische Zahlungssysteme 23
2.2.1 Traditionelle Sicherheitsanforderungen 25
2.2.2 Holistische Sicherheitsanforderungen 30
2.2.3 Spezielle Sicherheitsanforderungen 32
2.2.4 Übergeordnete Sicherheitsanforderungen 32
2.3 Existierende elektronische Zahlungssysteme 35
2.3.1 Kartenbasierte Zahlungssysteme 36
2.3.1.1 GeldKarte 37
2.3.1.2 Mondex 41
2.3.2 Netzbasierte Zahlungssysteme 44
2.3.2.1 Millicent 44
2.3.2.2 SET-Anwendung 48
2.3.2.3 Ecash 52
2.4 Bewertung anhand der grundlegenden Sicherheitsanforderungen 55
2.5 Multifunktionale Chipkarten als Vision 58

3 Chipkarten 63

3.1 Historie und Entwicklung 63
3.2 Identifikationskarten 65
3.2.1 Hochgeprägte Karten 65
3.2.2 Magnetstreifenkarten 66
3.2.3 Chipkarten 66
3.2.3.1 Speicherkarten 67
3.2.3.2 Mikroprozessorkarten 68
3.2.3.3 Optische Speicherkarten 71

3.3 Physikalische und elektrische Eigenschaften ... 72
3.3.1 Formate, Kartenmaterial ... 72
3.3.2 Kontaktlose Karten ... 73
3.3.3 Kontaktbehaftete Karten ... 74
3.3.4 Sicherheitsmerkmale ... 75
3.3.5 Spannungsversorgung ... 76
3.4 Lebenszyklus einer Chipkarte ... 76
3.5 Chipkarten-Betriebssysteme ... 79
3.5.1 Sicherheitsanforderungen ... 81
3.5.2 Ablaufsteuerung ... 85
3.5.3 Speicherorganisation ... 86
3.5.4 Dateistrukturen ... 88
3.5.5 Datenübertragung zur Chipkarte ... 90
3.5.5.1 Physikalische Schicht ... 91
3.5.5.2 Leitungsschicht ... 91
3.5.6 Das Chipkarten-Betriebssystem STARCOS ... 94
3.5.6.1 Datenstrukturen ... 94
3.5.6.2 Attribute ... 94
3.5.6.3 Befehlsablaufschema ... 95
3.5.7 Das Chipkarten-Betriebssystem der Java-Card ... 96
3.5.7.1 Systemarchitektur ... 97
3.5.7.2 Java-Card Klassen ... 98
3.5.7.3 Sicherheitskonzept ... 99

4 Kryptographische Verfahren ... 103

4.1 Symmetrische Algorithmen ... 104
4.2 Asymmetrische Algorithmen ... 108
4.3 Hash-Algorithmen ... 110
4.4 Digitale Signaturen ... 111
4.4.1 Erzeugen einer Signatur ... 112
4.4.2 Prüfen einer Signatur ... 112
4.5 Zertifikate und Infrastrukturen ... 113
4.6 Identifizierung und Authentisierung ... 116
4.6.1 Identifizierung durch Wissen ... 117
4.6.2 Biometrische Verfahren ... 117
4.6.3 Symmetrische Authentisierung ... 121
4.6.4 Asymmetrische Authentisierung ... 122

5 Allgemeine Sicherheitsmodelle ... 125

5.1 Bedeutung von Sicherheitsmodellen ... 125
5.2 Klassische Zugriffskontrollmodelle ... 129

5.3 Vertraulichkeitsmodell ... 130
5.3.1 Elemente ... 130
5.3.2 Sicherheitsklassen ... 131
5.3.3 Zugriffsarten ... 131
5.3.4 Systemzustand und Zustandsänderungen ... 132
5.3.5 „Sicheres“ System ... 134
5.4 Integritätsmodell ... 135
5.4.1 Elemente ... 136
5.4.2 Prozeduren ... 137
5.4.3 Regeln ... 137
5.5 Telekooperationsmodell ... 140
5.5.1 Begriffsdefinitionen ... 140
5.5.2 Darstellung des Telekooperationsmodells ... 142
5.6 Formales Datenschutzmodell ... 145
5.6.1 Zustandsvariablen ... 146
5.6.2 Sicherheitsinvarianten ... 149
5.6.3 Zustandsüberführungsfunktionen ... 150
5.7 Rollenbasiertes Zugriffsmodell ... 154
5.7.1 Begriffsdefinitionen ... 155
5.7.2 Grundlegende Vereinbarungen ... 155
5.7.2.1 Benutzer, Rollen und Transaktionen ... 155
5.7.2.2 Rollen und Rollenhierarchien ... 157
5.7.2.3 Rollenautorisierung ... 157
5.7.2.4 Rollenaktivierung ... 158
5.7.2.5 Operative Trennung von Pflichten ... 160
5.7.2.6 Zugriff auf Objekte ... 160
5.8 Bewertung anhand der grundlegenden Sicherheitsanforderungen ... 161
5.9 Folgerungen für ein neues Sicherheitsmodell ... 164

6 Rollen- und Aufgabenbasiertes Sicherheitsmodell ... 167

6.1 Grundidee des R&A-Modells ... 168
6.2 Begriffsdefinitionen ... 169
6.3 Bedeutung des R&A-Modells ... 171
6.4 Formale Beschreibung des R&A-Modells ... 172
6.4.1 Zustandsvariablen und Funktionen ... 173
6.4.2 Konsistenzregeln ... 180
6.4.3 Überführungsfunktionen ... 183
6.4.4 Beweisskizze ... 187
6.4.4.1 Induktionsanfang ... 188
6.4.4.2 Induktionsannahme ... 191
6.4.4.3 Induktionsschritt ... 191
6.4.5 Zustandsdiagramm ... 192
6.5 Zusätzliche Darstellung des R&A-Modells als Petrinetz ... 194
6.6 Bewertung anhand der grundlegenden Sicherheitsanforderungen ... 209
6.7 Erweiterung des R&A-Modells ... 210

7 Modellierung von R&A-Anwendungen 215

7.1 Statischer Aspekt des R&A-Modells 216
7.1.1 Subjekte, Rollen und Aufgaben 217
7.1.2 Funktionen 222
7.1.3 Objekte 227
7.1.4 Prozeduren und Zugriffe 230
7.2 Dynamischer Aspekt des R&A-Modells 232
7.2.1 Bezahlen mit der Geldbörse 233
7.2.2 Geldbörse laden und Transaktionslimit für EC-Karte setzen 235

8 Ausblick 241

8.1 Zusammenfassung 241
8.2 Weiterführende Aufgaben 244

Anhang 249

Literatur 249
Normen 265
Abkürzungen 269

Abbildungen

Abbildung 2-1: Produkte, die via Internet gekauft werden 21
Abbildung 2-2: Probleme beim Einkauf im Internet 22
Abbildung 2-3: Aktionsmodell im elektronischen Zahlungsverkehr 23
Abbildung 2-4: Vertraulichkeit, Integrität und Verfügbarkeit 28
Abbildung 2-5: Bezahlen mit der GeldKarte 38
Abbildung 2-6: Mondex 42
Abbildung 2-7: Millicent 45
Abbildung 2-8: Secure Electronic Transactions (SET) 49
Abbildung 2-9: Ecash 53
Abbildung 2-10: Schätzung der Entwicklung des Weltmarktes für Chipkarten 58

Abbildung 3-1: Lage der Hochprägung auf der ID-1 Karte 65
Abbildung 3-2: Lage des Magnetstreifens auf der ID-1 Karte 66
Abbildung 3-3: Speicherkarte mit Sicherheitslogik 67
Abbildung 3-4: Mikroprozessorkarte 68
Abbildung 3-5: Chipkarten-Microcontroller 69
Abbildung 3-6: Vergleich des Platzbedarfes für ein Bit verschiedener Speicherarten 71
Abbildung 3-7: Lage des optischen Speicherbereiches auf der ID-1 Karte 72
Abbildung 3-8: Definition des ID-1 Formats für Chipkarten 73
Abbildung 3-9: Kontakte bei Chipkarten 74
Abbildung 3-10: Lebenszyklus einer Chipkarte 77
Abbildung 3-11: Befehlsabarbeitung innerhalb eines Chipkarten-Betriebssystems 86
Abbildung 3-12: Dateiorganisation 88
Abbildung 3-13: Datenstrukturen von Elementary Files 90
Abbildung 3-14: Aufbau eines Befehls bei T=0 92
Abbildung 3-15: Aufbau eines Übertragungsblocks bei T=1 93
Abbildung 3-16: Datenstrukturen object und compute 94
Abbildung 3-17: Befehlsablaufschema 96
Abbildung 3-18: Systemarchitektur der Java-Card 97
Abbildung 3-19: Gateway-Modell 100

Abbildung 4-1: Ver- und Entschlüsselung bei symmetrischen Algorithmen 103
Abbildung 4-2: DES im ECB-Modus 104
Abbildung 4-3: DES im CBC-Modus 105
Abbildung 4-4: Prinzipieller Ablauf bei Triple-DES 106
Abbildung 4-5: Ver- und Entschlüsselung bei asymmetrischen Algorithmen 108
Abbildung 4-6: Erzeugen und Prüfen einer digitalen Signatur 111
Abbildung 4-7: Erzeugen und Prüfen einer digitalen Signatur unter Verwendung einer Hash-Funktion 112
Abbildung 4-8: Generelles Zertifikat 114

Abbildung 4-9: Top-Down-Zertifizierungshierarchie 115
Abbildung 4-10: Einseitige, symmetrische Authentisierung der Chipkarte durch das Terminal 121
Abbildung 4-11: Gegenseitige symmetrische Authentisierung der Chipkarte durch das Terminal 122
Abbildung 4-12: Einseitige, statische und asymmetrische Authentisierung der Chipkarte durch das Terminal 123
Abbildung 4-13: Einseitige, dynamische und asymmetrische Authentisierung der Chipkarte durch das Terminal 124

Abbildung 5-1: Realität und Modell 126
Abbildung 5-2: Unerwünschter indirekter Informationsfluß 133
Abbildung 5-3: Zugriffsmöglichkeiten im Bell-LaPadula-Modell 133
Abbildung 5-4: Grafische Darstellung des Clark-Wilson-Modells 139
Abbildung 5-5: Beziehung zwischen Rollenziel und Handlungszweck 143
Abbildung 5-6: Hierarchie von Akteuren 144
Abbildung 5-7: Koordination von Akteuren 144
Abbildung 5-8: Beziehung zwischen Benutzern, Rollen und Transaktionen 156
Abbildung 5-9: Beziehung zwischen Benutzern und Subjekten 156
Abbildung 5-10: Beziehung zwischen Transaktionen und Objekten 157

Abbildung 6-1: Autorisierte Aufgaben und Rollen für Subjekt s_1 175
Abbildung 6-2: Beziehung zwischen Subjekten und Rollen im R&A-Modell 176
Abbildung 6-3: Beziehung zwischen Subjekten und Aufgaben im R&A-Modell 176
Abbildung 6-4: R&A-Modell als Zustandsdiagramm 193
Abbildung 6-5: Authentisierung eines Subjekts 197
Abbildung 6-6: R&A-Modell als gefärbtes Petrinetz 200

Abbildung 7-1: Verwendete Symbole 217
Abbildung 7-2: Subjekte und Rollen der R&A-Anwendung 217
Abbildung 7-3: Subjekte und Aufgabenbereiche der R&A-Anwendung 218
Abbildung 7-4: Aufgabenbereich „Finanztransaktionen“ mit Aufgaben und Rollen 219
Abbildung 7-5: Aufgabenbereich „Administrieren“ mit Aufgaben und Rollen 221
Abbildung 7-6: Bezahlen mit der Geldbörse 233
Abbildung 7-7: Geldbörse laden und Transaktionslimit für EC-Karte setzen 236

Tabellen

Tabelle 2-1: Bewertung der beschriebenen Zahlungssysteme 55

Tabelle 3-1: Kontaktbelegung nach ISO-Norm 7816-2 74
Tabelle 3-2: Übersicht über gängige Sicherheitsmerkmale von Chipkarten 75
Tabelle 3-3: Normen für Chipkarten-Betriebssyssteme 81
Tabelle 3-4: Sicherheitsanforderungen an kryptobasierte Systeme 85
Tabelle 3-5: Kernklassen der Java-Card 99

Tabelle 4-1: Vorgeschlagene Algorithmen für den Advanced Encryption Standard (AES) 107
Tabelle 4-2: Physiologische Merkmale 119
Tabelle 4-3: Verhaltensbasierende Merkmale 120

Tabelle 5-1: Übersicht über Sicherheitsmodelle 128

Tabelle 6-1: Autorisierte Rollen und Aufgaben für ein Subjekt 177

Tabelle 7-1: Subjekte der R&A-Anwendung 217
Tabelle 7-2: Autorisierte Rollen-Aufgaben-Kombinationen für das Subjekt Karteninhaber 223
Tabelle 7-3: Autorisierte Rollen-Aufgaben-Kombinationen für das Subjekt Bank 224
Tabelle 7-4: Geldbörsenobjekte 227
Tabelle 7-5: EC-Kartenobjekte 228
Tabelle 7-6: Kreditkartenobjekte 229
Tabelle 7-7: Administrationsobjekte 230
Tabelle 7-8: Zulässige Prozeduren 231
Tabelle 7-9: Beispielhafte Zugriffe auf Objekte 232

1 Einleitung

1.1 Motivation

Die Bedeutung des elektronischen Handels (Electronic Commerce) nimmt für die Wirtschaft und für die Gesellschaft immer mehr zu. Dabei kann man bei Electronic Commerce zwischen den Bereichen elektronischer Zahlungsverkehr (Electronic Payment) und elektronischen Bankdiensten (Electronic Banking) unterscheiden. Während bei Electronic Banking in der Regel zwei Parteien beteiligt sind, nämlich der Kunde und die Bank, spielt bei Electronic Payment eine weitere Partei, der Händler, eine entscheidende Rolle.

Immer mehr Finanztransaktionen werden auf elektronischem Wege abgewickelt. Bankdienste können vom heimischen Computer aus per Homebanking getätigt werden. Für herkömmliche Zahlungsverfahren existieren elektronische Varianten. So ist es für den Kunden möglich, seine Ware per elektronischem Scheck, Koupon, elektronischer Kreditkarte oder per elektronischem Geld zu bezahlen. Es gibt elektronische Zahlungsverfahren sowohl für kleine Beträge (Micropayment) als auch für große Beträge (Macropayment).

Immer mehr elektronische Zahlungssysteme und ein Teil der Homebanking-Software basieren auf Chipkarten. Eine Chipkarte kann entweder als Träger der Geld- oder Verrechnungseinheiten dienen oder sie stellt das Zugangsmedium zu netzgestützten Diensten dar. Die Bedeutung der Chipkarte für den elektronischen Zahlungsverkehr nimmt immer mehr zu. Sie wird langfristig nicht nur die heute im Einsatz befindliche Magnetstreifenkarte ersetzen, sondern auch neue Anwendungsfelder erschließen.

Die Zahl der Karten (sowohl Magnetstreifen- als auch Chipkarten) im Portemonnaie eines Benutzers läßt sich jedoch nicht beliebig erhöhen. Deshalb werden in Zukunft mehrere Funktionen auf einer Karte zu einer multifunktionalen Chipkarte kombiniert werden. Ein erster Trend dieser Entwicklung ist in der Kombination der deutschen EC-Karte mit der GeldKarte zu sehen. Wann sich die Entwicklung einer multifunktionalen Chipkarte für den elektronischen Zahlungsverkehr konkret durchsetzen wird, hängt von den technischen und organisatorischen Rahmenbedingungen der Kreditwirtschaft, des Handels und rechtlicher Regelungen ab. In der vorligenden Arbeit wird diese Entwicklung einer multifunktionalen Chipkarte für den elektronischen Zahlungsverkehr für die weiteren Betrachtungen zugrunde gelegt.

Werden unterschiedliche Zahlungsverfahren und Homebanking-Software auf einer Chipkarte kombiniert, können für den Benutzer einer multifunktionalen Chipkarte Sicherheitsprobleme auftreten, da jedes Zahlungssystem und jede Bankensoftware unterschiedliche (zum Teil widersprüchliche) Anforderungen bezüglich Übertragungssicherheit, Vertraulichkeit, Integrität, Anonymität und einfacher Handhabung erfüllt. Als Folge daraus ergeben sich Anforderungen an die Gestaltung einer solchen Chipkarte. Sie

muß die folgenden herausgearbeiteten grundlegenden Sicherheitsanforderungen unterstützen.

Weiterhin muß gewährleistet sein, daß die einzelnen Anwendungen auf der Chipkarte sich nicht (gewollt oder ungewollt) gegenseitig beeinflussen. Es muß sichergestellt werden, daß nur auf die zulässigen Datenbereiche der jeweiligen Anwendung zugegriffen wird.

Werden netzgestützte Finanzdienste genutzt, muß gewährleistet werden, daß nur die gewünschten Dienste und die dazu notwendigen Ressourcen zur Verfügung gestellt werden. Die Implementierung einer multifunktionalen Chipkarte stellt hohe Anforderungen an die Gestaltung der Benutzungsoberfläche. Die Handhabung einer solchen Chipkarte muß einfach und intuitiv erfolgen. Sie muß das informationelle Selbstbestimmungsrecht des Benutzers unterstützen. Das bedeutet, daß der Benutzer selbst entscheiden kann, wer wann welche Informationen über ihn erhält. Für eine multifunktionale Chipkarte im elektronischen Zahlungsverkehr bedeutet das, daß der Benutzer selbst entscheiden können muß, welches Zahlungsmittel er einsetzt oder welche Bankdienste er nutzt, und welche Informationen er damit über sich freigibt.

Diese grundlegenden Sicherheitsanforderungen können mit Hilfe eines Sicherheitsmodells für multifunktionale Chipkarten erfüllt werden. Nach ausführlicher Recherche bestehender Sicherheitsmodelle wird festgestellt, daß diese nicht ausreichen, um die gestellten Anforderungen zu erfüllen. Aus diesem Grund wird ein neues, spezifisches Sicherheitsmodell entwickelt, das dem Benutzer Rollen und Aufgaben zur Verfügung stellt. Der Benutzer kann verschiedene Aufgaben in unterschiedlichen Rollen erledigen.

Je nach Rollen-Aufgaben-Kombination bewegt er sich in einem individuellen Sicherheitsumfeld. Durch aktive Wahl einer solchen Kombination entscheidet der Benutzer selbst, welche Information er durch Nutzung von Zahlungssystemen oder Bankdiensten freigibt. Durch die Kombination von unabhängigen Rollen und Aufgaben hat dieses Sicherheitsmodell (R&A-Modell) zwei Dimensionen, im Gegensatz zu bestehenden eindimensionalen Sicherheitsmodellen. Das R&A-Modell wird für multifunktionale Chipkarten im elektronischen Zahlungsverkehr entwickelt (R&A-Chipkarte). Eine Anwendung des R&A-Modells (R&A-Anwendung) beschränkt sich jedoch nicht auf den elektronischen Zahlungsverkehr.

Diese Arbeit hat drei innovative Ansätze:

1) Formulierung von grundlegenden Sicherheitsanforderungen für elektronische Zahlungssysteme und für multifunktionale Chipkarten im elektronischen Zahlungsverkehr

2) Entwicklung eines neuen formalen zweidimensionalen, spezifischen Rollen- und Aufgabenbasierten Sicherheitsmodells (R&A-Modell) und dessen Beweisskizze

3) Modellierung von R&A-Anwendungen auf einer multifunktionalen Chipkarte für den elektronischen Zahlungsverkehr (R&A-Chipkarte) auf der Basis des R&A-Modells

Es handelt sich um ein generelles Sicherheitsmodell, das universell einsetzbar ist. Das R&A-Modell wird so allgemein formuliert, daß es sich auf alle Anwendungsbereiche übertragen läßt, in dem verschiedene Anwendungen unterschiedliche Sicherheitsanforderungen unterstützen sollen.

Neben der naheliegenden Implementierung auf einer Chipkarte, ist die Implementierung in einem Personal Computer (PC) denkbar. Da in einem PC genügend Speicherplatz und ausreichende Prozessorleistung zur Verfügung steht, kann das R&A-Modell in seinem vollen Umfang genutzt werden. Eine Implementierung auf einer Chipkarte mit heutigen Speicher- und Prozessor-Generationen kann eine technisch bedingte Einschränkung der Möglichkeiten des R&A-Modells darstellen, die jedoch mit zukünftigen Chipkarten-Generationen in naher Zukunft überwunden werden wird.

1.2 Übersicht über die Arbeit

Die vorliegende Arbeit teilt sich in acht Kapitel auf, die nachfolgend kurz erläutert werden. Nach diesem Einleitungskapitel erfolgt in Kapitel 2 eine Beschreibung der Situation im elektronischen Zahlungsverkehr. Es wird die Bedeutung des elektronischen Zahlungsverkehrs für die Gesellschaft und die Wirtschaft beschrieben, sowie seine bisherige Entwicklung und mögliche zukünftige Entwicklungen.

Nach einer Einführung in das Thema Sicherheit werden allgemeine grundlegende Sicherheitsanforderungen formuliert. Diese grundlegenden Sicherheitsanforderungen werden auf den Bereich des elektronischen Zahlungsverkehrs abgebildet und als grundlegende anwendungsspezifische Sicherheitsanforderungen formuliert. Anschliessend werden einzelne elektronische Zahlungssysteme erläutert und anhand der zentralen Sicherheitsanforderungen bewertet. Danach erfolgt eine Abschätzung der zukünftigen Entwicklung im elektronischen Zahlungsverkehr, aus der die Vision für multifunktionale Chipkarten abgeleitet wird.

Das nachfolgende Kapitel ist ein Grundlagenkapitel zum Thema Chipkarten (siehe Kapitel 3). Da Chipkarten auch in Zukunft als Basis für etliche Zahlungssysteme verstärkt genutzt werden, werden in diesem Kapitel die Grundlagen für das Verständnis für die Chipkarten-Technologie gelegt. Neben den technischen Grundlagen der Chipkarten-Technologie, wie den verschiedenen Kartenarten, Datenübertragung zur Chipkarte und Lebenszyklus der Chipkarte, wird auf Prinzipien der Chipkarten-Betriebssysteme eingegangen.

Ebenso werden die kryptographischen Verfahren erläutert (siehe Kapitel 4), die im Zusammenhang mit Chipkarten große Bedeutung haben. Dabei werden symmetrische und asymmetrische kryprographische Verfahren erläutert, Verfahren zur Erzeugung und Prüfung digitaler Signaturen, sowie Techniken der Identifizierung und Authentisierung.

In Kapitel 5 werden die bestehenden Sicherheitsmodelle vorgestellt und abschließend bewertet. Neben dem klassischen Vertraulichkeitsmodell von Bell und LaPadula und dem Integritätsmodell von Clark und Wilson, werden ein Telekooperationsmodell und ein formales Datenschutzmodell vorgestellt. Die größte Bedeutung für die vorliegende Arbeit hat das rollenbasierte Zugriffsmodell, das ebenfalls in diesem Kapitel erläutert

wird. Aus der Bewertung der vorgestellten Sicherheitsmodelle folgt jedoch, daß die vorgestellten Sicherheitsmodelle den genannten Anforderungen nicht vollständig genügen, so daß die Notwendigkeit für die Entwicklung eines neuen, spezifischen Sicherheitsmodells besteht.

Dieses neue, spezifische Sicherheitsmodell wird in Kapitel 6 entwickelt. Es erfolgt eine formale Spezifikation des R&A-Modells als Zustandsautomat. Für das R&A-Modell werden Zustandsvariablen, Regeln und Überführungsfunktionen definiert. Die Korrektheit dieses Modells wird in einer formalen Beweisskizze nach dem Prinzip der vollständigen Induktion gezeigt. Eine graphische Darstellung des R&A-Modells erfolgt in Kapitel 6.4.5 als Zustandsdiagramm für die Auswahl genau einer Rollen-Aufgaben-Kombination.

In der Fachliteratur werden Sicherheitsmodelle meist informal beschrieben oder formal als Zustandsautomaten spezifiziert, selten jedoch bewiesen. Zusätzlich zur Darstellung des Zustandsautomatens als Zustandsdiagramm und der Beweisskizze, wird in diesem Kapitel eine weitere Darstellungsmöglichkeit vorgestellt, mit der komplexe Zusammenhänge graphisch deutlich gemacht werden können. Das R&A-Modell wird als Petrinetz dargestellt, um zusätzlich gleichzeitige Ausführung (Nebenläufigkeit) graphisch beschreiben zu können.

Das R&A-Modell wird auf das Gebiet des elektronischen Zahlungsverkehrs auf einer multifunktionalen R&A-Chipkarte abgebildet. Kapitel 7 beschreibt die Modellierung einer R&A-Anwendung im Detail. Es werden die Rollen und Aufgaben beschrieben, die für eine multifunktionale R&A-Chipkarte zur Verfügung stehen. Diese Chipkarte ermöglicht dem Benutzer die Nutzung von verschiedenen kartenbasierten Zahlungsverfahren und den Zugang zu netzgestützten Zahlungsverfahren und Homebanking-Diensten.

Die Modellierung der R&A-Chipkarte enthält einen statischen und einen dynamischen Aspekt. Der statische Aspekt beschreibt, welche Rollen und Aufgaben prinzipiell für einen Benutzer autorisiert sind, und der dynamische Aspekt beschreibt, welche gleichzeitige Ausführung von konfliktfreien Anwendungen für einen Benutzer möglich ist.

Das abschließende Kapitel 8 liefert eine kurze Zusammenfassung der Arbeit und erlaubt einen Blick in die Zukunft. Dabei werden offene Probleme und Fragen diskutiert und Hinweise für weitere Arbeiten gegeben.

Mit einem Rollen- und Aufgabenbasierten Sicherheitsmodell, das mit Chipkarten realisiert wird, erhöht sich die Sicherheit und der Datenschutz in der elektronischen Kommunikation wesentlich. Das R&A-Modell erlaubt das geregelte Zusammenwirken verschiedener Anwendungen mit unterschiedlichen, sich zum Teil widersprechenden, Sicherheitsanforderungen auf einer einzigen Chipkarte.

Für technisch wenig interessierte Benutzer stellt das R&A-Modell eine Standardkonfiguration zur Verfügung, so daß eine aktive Benutzung und Administration des Modells nicht notwendig ist und trotzdem ein Mindestmaß an Sicherheit gewährleistet wird.

Für technisch interessierte Benutzer bietet das R&A-Modell vielfältige Möglichkeiten in der Gestaltung des individuellen Sicherheitsniveaus für unterschiedliche Anwendungen. Die Benutzer können aktiv ihre Chipkarte nutzen und administrieren, um damit eine maximale Sicherheit bei der Benutzung von multifunktionalen Chipkarten zu erhalten.

Das R&A-Modell leistet somit einen wesentlichen Beitrag zur Erhöhung der Sicherheit für den Benutzer im elektronischen Zahlungsverkehr.

2 Elektronischer Zahlungsverkehr

Viele Geld- und Banktransaktionen werden zunehmend auf elektronische Wege verlagert. Dabei handelt es sich nicht nur um kommerzielle, sondern auch um private Transaktionen. Der Großteil dieser Transaktionen ist jedoch kommerziell. Bereits 1997 wurden rund um den Globus Tag für Tag geschäftliche Transaktionen im Wert von mehr als drei Billionen DM auf elektronischem Weg über Netze abgewickelt. Experten erwarten eine dramatische Zunahme der Nutzung von Electronic Commerce über Netze [Peuckert 1997]. Trotz dieser Prognosen verläuft das Wachstum eher zögerlich. Das vorrangige Hindernis für die Verbreitung von Electronic Commerce ist die fehlende Sicherheit.

Zunächst soll der Begriff Electronic Commerce definiert werden, da darunter vieles und oft nicht dasselbe verstanden wird. Zu Electronic Commerce gehört jede elektronische Transaktion, die per Telekommunikationstechnik elektronisch durchgeführt wird. Das beginnt mit der Werbung und Information für eine Leistung, reicht über Angebote und Ausschreibungen, kann den Bestell- und Lieferprozeß umfassen und betrifft zunehmend auch das Bezahlen (Electronic Payment). Unter Electronic Commerce wird auch das Erledigen von Bankgeschäften auf elektronische Weise verstanden (Electronic Banking). Eines der wesentlichen Merkmale von Electronic Commerce ist, daß es keine örtliche Beschränkung für die Nutzung gibt.

Im Rahmen dieser Arbeit soll in diesem Kapitel im wesentlichen auf die Bedeutung des elektronischen Zahlungsverkehrs (Electronic Payment) eingegangen werden. Electronic Banking und elektronische Angebote, beziehungsweise Marketing sollen nicht den Schwerpunkt dieser Arbeit bilden, auch wenn sie für Electronic Commerce von großer Bedeutung sind.

Der elektronische Zahlungsverkehr ist überall dort von großer Bedeutung, wo auf elektronischem Wege etwas bezahlt werden soll. Dies ist in fast allen Bereichen des täglichen Lebens denkbar, prinzipiell dort, wo auch heute mit herkömmlichen Zahlungssystemen, wie Bargeld, Schecks oder Kreditkarte, bezahlt wird.

Elektronische Zahlungssysteme kann man nach vielen verschiedenen Kriterien unterteilen. In dieser Arbeit wird zwischen kartenbasierten Zahlungssystemen (siehe Kapitel 2.3.1) und netzbasierten Zahlungssystemen (siehe Kapitel 2.3.2) unterschieden.

Es stellt sich die Frage, welche Bedeutung die Veränderungen des täglichen Lebens, weg von herkömmlichen Zahlungssystemen und hin zu elektronischen Zahlungssystemen, für die Bevölkerung haben (siehe Kapitel 2.1). Es ist zu klären, welche Anforderungen an Zahlungssysteme gestellt werden müssen (siehe Kapitel 2.2). Dabei geht es nicht nur um sicherheitstechnische, sondern auch um rechtliche und organisatorische Anforderungen.

Der Kunde als Benutzer der elektronischen Zahlungssysteme steht nun im Mittelpunkt der Wechselbeziehung zwischen Banken und Handel auf dem öffentlichen Präsentier-

teller der Netzwerke. Der Zahlungsverkehr, der früher als intime Vertrauensbeziehung galt, gerät nun in die Öffentlichkeit. Diese Situation erfordert eine verstärkte Konzentration auf die Gewährleistung der Sicherheitsbedürfnisse eines Benutzers bei der Benutzung von Zahlungssystemen. Es müssen Modelle und Werkzeuge zur Verfügung gestellt werden, die dem Benutzer eine sichere Nutzung von Zahlungssystemen ermöglichen, ohne daß er Gefahr läuft ein „gläserner" Kunde zu werden oder sein informationelles Selbstbestimmungsrecht (siehe Kapitel 2.2.4) zu verlieren.

2.1 Begriffsbestimmung und Bedeutung des elektronischen Zahlungsverkehrs

Geld im herkömmlichen Sinne ist nach allgemeiner Auffassung ein Zahlungsmittel, eine Recheneinheit und dient zur Wertaufbewahrung. Beim heutigen, klassischen Zahlungsverkehr kann man zwischen barem und unbarem Zahlungsverkehr unterscheiden.

Geld ist bekannt in seiner Form als Bargeld (barer Zahlungsverkehr). Bargeld wird von der Zentralbank herausgegeben und ist mit einem staatlichen Annahmezwang versehen [Böhle, Riehm 1998]. Bargeld, in Form von Scheinen und Münzen, ist das einzige gesetzliche Zahlungsmittel. Bargeld dient dabei immer auch der Wertaufbewahrung, denn Zahlungen mit Bargeld erfordern immer eine vorherige Beschaffung dieses Bargeldes. Der Kunde beschafft sich sein Bargeld bei der Bank am Schalter oder am Geldausgabeautomaten.

Im unbaren Zahlungsverkehr ist das Giralgeld, auch als Buchgeld bezeichnet, als allgemeines Zahlungsmittel anerkannt. Es stellt zwar kein gesetzliches, aber allgemein anerkanntes Zahlungsmittel dar. Für den Einsatz im Zahlungsverkehr bedarf das Giralgeld spezieller Zahlungsinstrumente, wie zum Beispiel den Scheck, die Lastschrift oder die Überweisung [Grill, Gramlich, Eller 1996]. Beim unbaren Zahlungsverkehr ist das Girokonto des einzelnen Bankkunden der übliche Zugangspunkt zum Netz des unbaren Zahlungsverkehrs (Gironetz). Scheck, Überweisung und Lastschrift sind die traditionellen Zahlungsinstrumente, um Übertragungen von Geldeinheiten innerhalb der, von Banken organisierten, Netze auszuführen [Bibow, Wichmann 1998].

Aufbauend auf die einzelnen Zahlungsinstrumente werden durch technische und rechtliche Ausgestaltungen einzelne Zahlungsverfahren des unbaren Zahlungsverkehrs definiert. Von Zahlungssystemen spricht man dann, wenn es sich um spezielle Informationsverarbeitungssysteme handelt, in denen Informationen über finanzielle Ansprüche verarbeitet werden ([Böhle, Riehm 1998], S. 139).

Die Banken haben ihren Kunden in den letzten Jahren neue Möglichkeiten eröffnet, unbare Zahlungen über ihr Girokonto zu veranlassen, zum Beispiel über Telefon, per Computer, sowie durch Debitkarten. In Deutschland existierten 1997 etwa 67 Millionen Debitkarten, wobei die EC-Karte besonders stark vertreten ist. Debitkarten haben auf einem Magnetstreifen Bankleitzahl und Kontonummer des Besitzers gespeichert. Sie werden häufig in Verbindung mit einer „Personal Identification Number" (PIN) eingesetzt, die bei Zahlungen in den Kartenleser eingegeben werden muß.

Bei Zahlung mit der EC-Karte unterscheidet man zwischen zwei Verfahren: „Electronic Fund Transfer at Point of Sale“ (EFTPOS) und „Point of Sale“ (POS):

- Das erste Verfahren (EFTPOS) arbeitet mit einer Online-Autorisierung der Zahlung durch die Bank. Die Online-Autorisierung führt zu einer Verzögerung und Verteuerung der Zahlung, stellt jedoch eine Reduzierung des Risikos für den Verkäufer dar. Ähnlich wie bei der traditionellen Überweisung erfolgt bei dieser Art der Zahlung die Girokontenbelastung des Karteninhabers sofort.
- Bei dem alternativen POS-Verfahren wird dagegen auf die Eingabe der PIN und die Online-Autorisierung verzichtet und statt dessen auf die Unterschrift des Karteninhabers vertraut. Die Übertragung erfolgt ähnlich dem traditionellen Lastschriftverfahren etwas verspätet, womit eine kurzfristige Kreditstellung durch den Handel verbunden ist, der dadurch auch das Zahlungsrisiko trägt.

Im Vergleich dazu beinhaltet die Verwendung einer Kreditkarte zur Zahlung immer eine Kreditinanspruchnahme. Die Belastung des Girokontos erfolgt im Durchschnitt erst einige Wochen nach dem Kauf.

Seit einiger Zeit ist es zu neuen Erscheinungsformen des Geldes gekommen, die den Begriff „elektronisches Geld“ prägen. Der gesamte Zahlungsverkehr wird zunehmend „elektronisch“ abgewickelt. Elektronische Zahlungssysteme haben eines gemeinsam: Sie repräsentieren oder verrechnen elektronisches Geld. Elektronisches Geld hat dabei die gleiche Funktionalität wie herkömmliches Geld, wobei zwischen Recheneinheit, Zahlungsmittel und Wertaufbewahrungsmittel unterschieden wird. Nicht jedes elektronische Zahlungssystem, das elektronisches Geld verarbeitet, erfüllt alle drei Funktionen, zum Beispiel bietet nicht jedes elektronische Zahlungsverfahren die Funktionalität der Wertaufbewahrung. Auf abstrakter Ebene kann man an elektronisches Geld folgende Anforderungen stellen, beziehungsweise Abgrenzungen gegen andere Zahlungsarten ziehen ([Böhle, Riehm 1998], S. 142):

- Die Zahlung mit elektronischem Geld erfordert kein Girokonto
- Im Moment des Kaufakts muß kein Kontakt mit einer Bank zur Autorisierung aufgenommen werden
- Das Bezahlverfahren ist so ausgestaltet, daß die einzelnen Bezahlvorgänge ohne persönliche Daten des Bezahlenden auskommen und dem Kunden dadurch Anonymität gewähren
- Systeme für elektronisches Geld sind kostengünstiger, weil sie keine Autorisierung erfordern
- Systeme für elektronisches Geld, die in offenen Netzen einsetzbar sind, sind für Fernzahlungen geeignet

Elektronisches Geld kann es sowohl in Form von Kartengeld als elektronische Geldbörse, als auch in Form von Netzgeld geben. In der sechsten Novelle des Kredit-

wesengesetzes (KWG) wurden diese beiden neuen Formen des Geldes aufgenommen [Bundesregierung 1997].

Es gibt verschiedene Definitionen zu elektronischem Geld, von denen einige kurz vorgestellt werden sollen. Die Bank für internationalen Zahlungsausgleich (BIS) versteht unter elektronischem Geld ein vorausbezahltes Geldprodukt, das auf einem elektronischen Medium gehalten wird, das sich im Besitz des Konsumenten befindet ([BIS 1996], S. 1).

Die Europäische Zentralbank definiert in ihrem Bericht zu elektronischem Geld:

> *„Elektronisches Geld wird allgemein definiert als eine auf einem Medium elektronisch gespeicherte Werteinheit, die allgemein genutzt werden kann, um Zahlungen an Unternehmen zu leisten, die nicht die Emittenten sind. Dabei erfolgt die Transaktion nicht notwendigerweise über Bankkonten, sondern die Werteinheiten auf dem Speichermedium fungieren als vorausbezahltes Inhaberinstrument.“*
> *([Europäische Zentralbank 1998], S. 8)*

In der Begründung zum Richtlinienentwurf der Europäischen Kommission, mit dem die Herausgeberschaft von elektronischem Geld geregelt werden soll, heißt es:

> *„Im Sinne dieses Vorschlags läßt sich elektronisches Geld am besten als digitale Form des Bargelds definieren, mit dem es viele Eigenschaften gemeinsam hat. Es ähnelt ihm in erster Linie darin, daß für die Verwendung elektronischen Geldes keine Genehmigung einer Bank oder eines anderen Dritten notwendig ist. Die Kunden kaufen das elektronische Äquivalent zu Münzen und Banknoten, das heißt sie tauschen Bargeld eins zu eins in Geldwert um. Sie tauschen also Bargeld gegen ein anderes Zahlungsmittel.*
>
> *Anstatt eine Debitkarte (für die ein Bankonto notwendig ist) oder eine Kreditkarte (für die erstens das Einverständnis der Kreditkartengesellschaft oder der Bank und zweitens eine angemessene Vorauszahlung von Kapital erforderlich ist) zu verwenden, hat der Kunde ein bargeldloses Zahlungsmittel erworben, das er genauso wie Bargeld oder andere Arten der Kartenzahlung nutzen kann, ohne daß dafür die Genehmigung eines Dritten notwendig wäre.*
>
> *Eine weitere Gemeinsamkeit von elektronischem Geld und Bargeld liegt in der Anonymität. Man benötigt kein Konto bei einem Finanzinstitut. Die Verbraucher können mit elektronischem Geld - wie mit Bargeld - einkaufen, ohne dem Einzelhandel spezielle Angaben (zum Beispiel Namen, Bankverbindung, ...) zu machen.“*
> *([Europäische Kommission 1998], S. 1ff)*

Diese Definitionen sind schwer vereinbar. Zum einen wird elektronisches Geld als vorausbezahltes Geldprodukt und als Inhaberinstrument, also ein Zahlungsinstrument im Besitz des Inhabers bezeichnet, zum anderen wird es als digitale Form von Bargeld mit all seinen Eigenschaften bezeichnet. Eine Mischform dieser Definitionen wäre wünschenswert, wobei die Realität dem nicht nahe kommt. Heutige elektronische Zahlungssysteme, die nach ihrer eigenen Definition elektronisches Geld repräsentieren, erfüllen nicht alle diese Eigenschaften (siehe Kapitel 2.3).

Die Bedeutung von elektronischen Zahlungssystemen (Electronic Payment) als Bestandteil von Electronic Commerce wird weltweit anerkannt. So hat zum Beispiel die amerikanische Regierung eine eigens eingesetzte Arbeitsgruppe damit beauftragt, die bereits jetzt abzusehenden Veränderungen des Lebens durch die zunehmende Bedeutung des Internets als Alltagsmedium zu beschreiben. Die sogenannte Consumer Electronic Payment Task Force (CEPTF) hat einen Arbeitsplan aufgestellt, in dem die notwendigen Schritte beschrieben sind, um Electronic Commerce für Industrie und den öffentlichen Bereich erfolgreich einzusetzen [CEPTF 1998].

Inwieweit heute bereits eine umfangreiche Nutzung von elektronischen Zahlungssystemen erfolgt, ist schwer zu sagen. Es existieren kaum Studien, die eine Nutzuung solcher Systeme untersuchen. Eine Studie des Internetmagazins FirstSurf beschreibt die heutige Situation des Internetshoppings [FistSurf 1998]. Das Marktforschungsinstitut Fittkau und Maaß wurde beauftragt herauszufinden, inwieweit im Internet eingekauft wird.

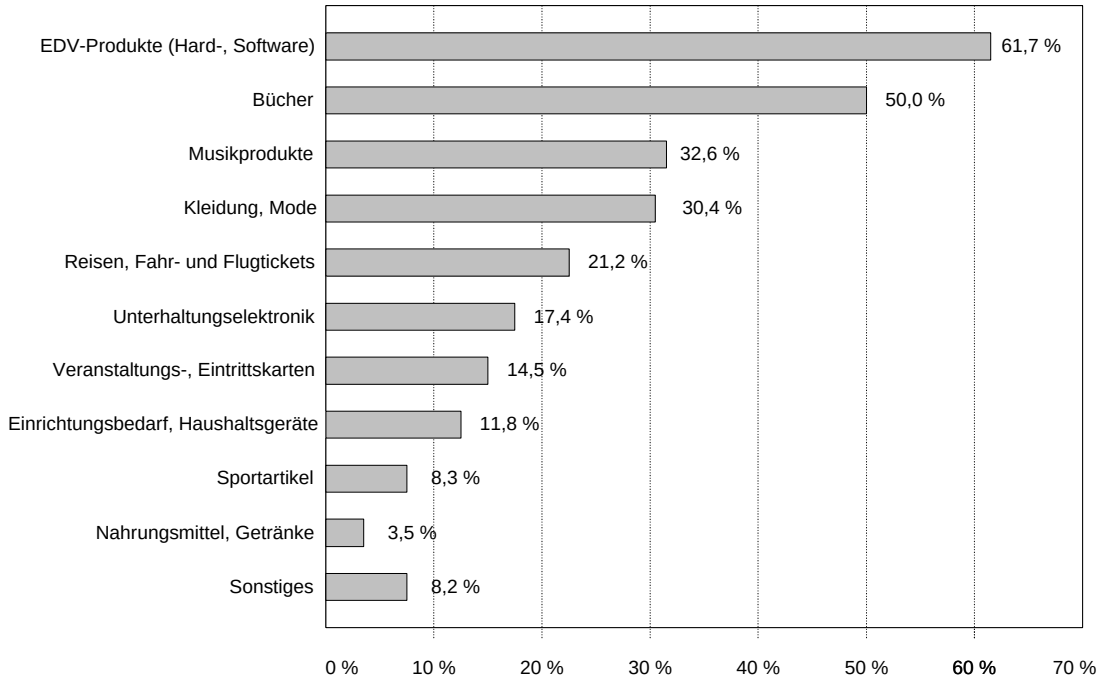


Abbildung 2-1: Produkte, die via Internet gekauft werden

Quelle: [FistSurf 1998]

In Koorperation mit 22 Online-Shops wurde eine Befragung der Internetnutzer durchgeführt. An dieser Befragungsaktion beteiligten sich 13244 Internetnutzer. Das

Ergebnis dieser Studie zeigt, daß in erster Linie EDV-Produkte und Bücher über das Internet gekauft werden (Abbildung 2-1). An dritter Stelle landeten Musikprodukte. Fast jeder dritte Befragte (30%) kauft seine Kleidung mit Hilfe seines vernetzten Rechners. Auf besonderes Interesse stoßen Reisen beim Onlineshopping (21% haben bereits einmal gebucht). Nur einer von zehn Befragten nutzt allerdings das virtuelle Pizzataxi. Bei der Befragung waren Mehrfachnennungen erlaubt.

Insgesamt bestellen Internetnutzer jedoch sehr unregelmäßig. Die größten Schwierigkeiten wurden in der Studie als folgende bezeichnet (Abbildung 2-2):

- Die Unsicherheit des Bezahlvorgangs: Fast zwei Drittel der Befragten halten die Sicherheit des Zahlungsverkehrs für das gravierendste Problem.
- Angst vor Schnüffelei: Deutschlands Internetnutzer sind kritisch. Mehr als die Hälfte (54%) hat Sorgen um die Sicherheit personenbezogener Daten.
- Langwierige Produktsuche: Viele Nutzer (38%) beklagen erhebliche Orientierungsprobleme beim Recherchieren von Produkten.

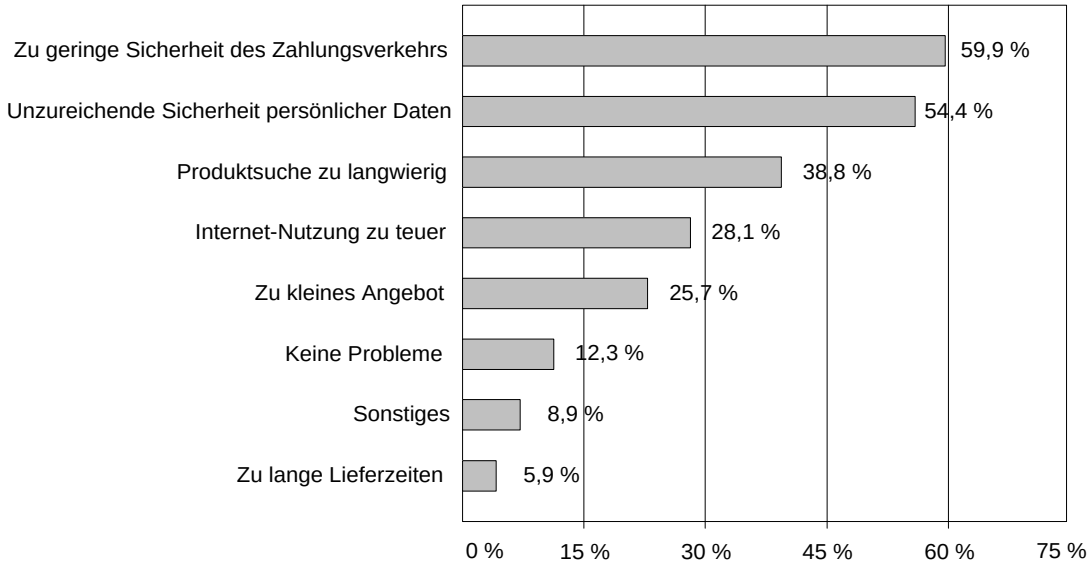


Abbildung 2-2: Probleme beim Einkauf im Internet
Quelle: [FirstSurf 1998]

Die wesentliche Aussage dieser Studie besagt, daß eine zögerliche Nutzung des Internetshoppings vor allem in der fehlenden Sicherheit der Bezahlvorgänge und der Angst vor dem Auswerten von Datenspuren begründet liegt. Es ist also notwendig, dem potentiellen Kunden Modelle und Verfahren an die Hand zu geben, die ein sicheres Bezahlen unter Wahrung des Persönlichkeitsrechtes ermöglichen. Dies ist Ziel der vorliegenden Arbeit.

Um nun die einzelnen Probleme und Gefahren deutlich herausarbeiten zu können, wird ein Aktionsmodell (Abbildung 2-3) eingeführt, das alle beteiligten Beteiligten im elektronischen Zahlungsverkehr beschreibt. Das Aktionsmodell beschreibt den Handel in offenen Netzen. Neben den Beteiligten des früheren klassischen Handlungsdreiecks

von Kunde, Händler und Bank, sind im elektronischen Zahlungsverkehr weitere Parteien beteiligt. Zwischen den einzelnen Banken stehen Evidenzzentralen, die den Geldfluß zwischen den Banken koordinieren.

Einsetzbar sind ebenfalls Makler, die zwischen den Währungen der einzelnen Geldsysteme der unterschiedlichen Banken tauschen.

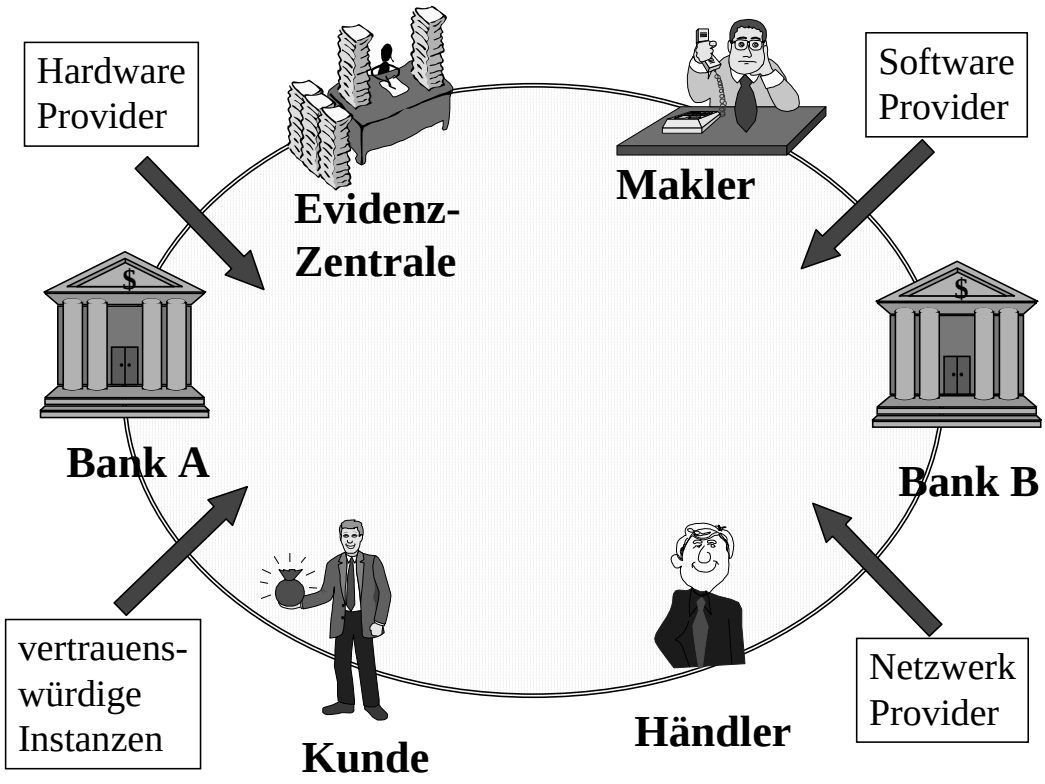


Abbildung 2-3: Aktionsmodell im elektronischen Zahlungsverkehr

Zusätzlich wird es vertrauenswürdige Instanzen geben, die im Bereich Schlüsselverwaltung arbeiten und damit die Vertrauensbasis für die Verwendung von asymmetrischen Verschlüsselungsverfahren darstellen (siehe Kapitel 4).

Neue Beteiligte im elektronischen Zahlungsverkehr sind Hardware- und Software-Provider und Netzwerkprovider. Sie sorgen für einen reibungslosen technischen Ablauf und stellen die dafür notwendige Infrastruktur zur Verfügung.

2.2 Grundlegende Sicherheitsanforderungen an elektronische Zahlungssysteme

Wer Leistungen auf elektronischem Wege verkaufen will, muß auch Verfahren zur Abrechnung anbieten. Bei jedem Handel erfolgt in irgendeiner Form ein Tausch von „Ware gegen Bezahlung“. Für verschiedene Kontexte gibt es verschiedene Regeln, die den Ablauf dieser Verfahren festlegen. Aus dem täglichen Leben sind die in unserer Kultur üblichen Regeln für den Umgang mit Bargeld, mit Schecks, mit EC- und

Kreditkarten geläufig. Diese Regeln sind durch Rechtsverbindlichkeit geschützt. Wenn ein Kunde eine Ware bestellt, verpflichtet er sich, diese bei Erhalt auch zu bezahlen, ebenso wie der Verkäufer sich verpflichtet, dem Kunden die angebotene Ware zukommen zu lassen. Elektronische Zahlungssysteme haben eine Analogie zu nichtelektronischen Zahlungssystemen. Das betrifft sowohl Kriterien der Anonymität oder Nachweisbarkeit, als auch den Zeitpunkt des Geldtransfers. Dieser beschreibt, ob es sich um vorausbezahlte, sofort bezahlte oder später bezahlte Zahlungsverfahren handelt.

Grundsätzlich gibt es eine Reihe von Anforderungen an elektronische Zahlungssysteme (und auch an Electronic Commerce im allgemeinen). Dazu gehören Rechtssicherheit, eine bestimmte Robustheit der Kommunikationsinfrastruktur sowie Begrenztheit und gerechte Verteilung der verbleibenden Risiken. Zu den sicherheitsrelevanten Anforderungen, die durch kryptographische Verfahren gelöst werden können, zählen das Erkennen und Verhindern von Manipulation und von Maskeraden, das Ausspähen von Geheimnissen, das Abstreiten von Handlungen, das Abfangen und Unterdrücken von Nachrichten sowie das Wiedereinspielen von Nachrichten. Dabei sind die technischen Probleme meist mit geringerem Aufwand zu lösen als die rechtlichen und organisatorischen Probleme.

In kommerziellen Systemen, in denen der Nutzen und der Schaden in Geldeinheiten gemessen werden kann, kann es sinnvoll sein, die nachfolgenden Sicherheitsanforderungen nicht in ihrer Gesamtheit durchzusetzen. Wenn finanzieller Aufwand und dadurch verursachter Schaden in keiner Relation zueinander stehen, kann es sinnvoll sein, folgende Definition von Sicherheit benutzen:

> *„Relative kommerzielle Sicherheit: Ein System ist im kommerziellen Sinn sicher, wenn die finanziellen Aufwendungen für eine Attacke zu jedem Zeitpunkt den maximal möglichen Gesamtschaden, den diese Attacke zufügt, deutlich überschreiten.“ [Posch 1999]*

Der Begriff der relativen kommerziellen Sicherheit kann dazu benutzt werden, um Systeme im elektronischen Zahlungsverkehr in der Praxis damit zu bewerten.

Bevor die einzelnen Sicherheitsanforderungen an elektronische Zahlungssysteme herausgearbeitet werden können, ist es notwendig, grundsätzlich die verwendeten Begriffe zu erläutern. Es wird zwischen folgenden Sicherheitsanforderungen unterschieden:

- Traditionelle Sicherheitsanforderungen
- Holistische Sicherheitsanforderungen
- Anwendungsspezifische Sicherheitsanforderungen
- Übergeordnete Sicherheitsanforderungen

2.2.1 Traditionelle Sicherheitsanforderungen

Unter Sicherheit versteht man im allgemeinen Sprachgebrauch die Abwesenheit von Gefahr [Grimm 1994]. Der Begriff der Sicherheit und Begriffe im Bedeutungsumfeld davon können sehr allgemein verstanden werden und nicht alle Autoren meinen immer dasselbe mit diesen Begriffen.

In der deutschen Sprache wird der Begriff Sicherheit mit einer subjektiven und einer objektiven Dimension definiert:

> *„Zustand des Unbedrohtseins, der sich objektiv im Vorhandensein von Schutz(einrichtungen) beziehungsweise im Fehlen von Gefahr(enquellen) darstellt und subjektiv als Gewißheit über die Zuverlässigkeit von Sicherungseinrichtungen empfunden wird.“ ([Meyer 1992], S. 162)*

Der subjektive Aspekt der Sicherheit beschreibt ein Gefühl, das als Vertrauen in Personen oder Systeme bezeichnet werden kann, während der objektive Aspekt eine Eigenschaft einer Person oder eines Systems (Verläßlichkeit) beschreibt. Vertrauen ist also ein Gefühl des vertrauenden Subjektes, während Verläßlichkeit eine Eigenschaft des Objektes ist. Vertrauen ist ein positives Gefühl, das unterstellt, daß der vertrauenswürdige Mensch oder das vertrauenswürdige System so handelt, wie es der vertrauende Mensch erwartet. Vertrauen basiert auf Erfahrung und kann sich stärken oder verschwinden. Es kann begründet oder unbegründet sein und basiert auf psychischen Prozessen ([Grimm 1994], S. 20ff).

Unter der Verläßlichkeit eines Menschen oder einer funktionalen Sache wird die nachweisbare funktionale Zuverlässigkeit als Eigenschaft des Menschen oder der Sache verstanden. Die Verläßlichkeit gehört in einen gesellschaftlichen Kontext, in dem sich die Frage stellt, wie ein Verläßlichkeitsnachweis zu führen ist. Die Idee ist eine Rekursion von verläßlichen Verfahren, die ihren Anfang in einem psychologisch begründeten Vertrauen hat. Die Basis ist das Vertrauen, dem vor allem Erfahrung zugrunde liegt. So beziehen sich Vertrauen und Verläßlichkeit auf eine gemeinsame Funktion, wobei Vertrauen ein Gefühl eines vertrauenden Subjektes darstellt und Verläßlichkeit eine Eigenschaft des vertrauenswürdigen Objekts ist.

Der Begriff der Sicherheit ist nicht scharf eingrenzbar. Er bezeichnet einen wenig faßbaren Zustand und eher eine Zielbestimmung. Psychologische Faktoren spielen ebenso eine Rolle wie äußere Maßnahmen. Die Bedeutung dieses Begriffs veränderte sich im Laufe der Jahre und spiegelt dabei die Entwicklungen von Technik und Anwendungen wider [Kossakowski 1999]. Um zu verstehen, was Sicherheit konkret bedeutet, müssen die wertvollen Güter und die möglichen Gefahren, die diese bedrohen, aufgezeigt werden. Dabei können die Begriffe Gefahr und Bedrohung direkt wie in der Umgangssprache verstanden werden.

Dieser sehr allgemeine Begriff der Sicherheit kann nun auf die Informationstechnik abgebildet werden. Dabei bezieht sich Sicherheit in der Informationstechnik hauptsächlich auf Maßnahmen, die Sicherheit herstellen, und weniger auf Eintrittswahrscheinlichkeit und Größe von möglichen Schäden, die wegen der Anwendungsvielfalt

nicht allgemein festgestellt werden können. Störfallanalysen und Risikobetrachtungen sind meist kein Thema der architekturellen Gestaltung, sondern des Betriebs in einer konkreten Anwendungsumgebung, was deutlich macht, daß in der Gestaltung des IT-Systems wesentliche Sicherheitsaspekte vernachlässigt wurden. Eine sinnvolle Risikoanalyse muß bereits im Designprozeß eines IT-Systems erfolgen, damit den möglichen Schwachstellen und Bedrohungen schon zu Beginn der Entwicklung entgegengewirkt werden kann. Die Praxis zeigt jedoch häufig den Fall, daß Risikoanalysen erst nach Beendigung eines Systems durchgeführt werden. Maßnahmen, die zu diesem Zeitpunkt noch ergriffen werden können, sind in der Regel sehr zeit- und kostenaufwendig und werden nicht selten aus eben diesen Gründen nicht ergriffen. Werden die Maßnahmen zu diesem späten Zeitpunkt doch noch ergriffen, können sie sogar negative Auswirkungen auf die Anwendung haben, da nur eine lückenhafte Sicherheit erreicht werden kann.

Sicherheit in der Informationstechnik beruht auf Maßnahmen, die Angriffen entgegenwirken, durch die wertvolle Güter bedroht sind. Sicherheit von IT-Systemen kann folgendermaßen definiert werden:

> *„Unter Sicherheit von IT-Systemen versteht man eine Eigenschaft eines IT-Systems, bei der Maßnahmen gegen die im jeweiligen Einsatzumfeld als bedeutsam angesehenen Bedrohungen in dem Maße wirksam sind, daß die verbleibenden Risiken tragbar sind.“ ([Grimm 1994], S. 27)*

Risiko ist ein Maß dafür, daß die bestehende Sicherheit eines Systems gebrochen wird. Da die subjektive Komponente der Sicherheit in der Informationstechnik sich auf Vertrauen stützt, hat Risiko also Einfluß auf Vertrauen. Als notwendige Bedingung für Vertrauen in eine Sache muß das Risiko so klein sein, daß man das Schadensausmaß vernachlässigen kann oder das Eintreten des Schadens „so gut wie“ ausschließen kann. Gegen einige Bedrohungen können keine Maßnahmen getroffen werden, also verbleibt ein sogenanntes „Restrisiko“. Es ist damit das Risiko gemeint, das trotz getroffener Sicherheitsmaßnahmen immer noch besteht.

Um dieses Risiko so klein wie möglich zu halten, muß ein gewisser „Grundschutz“ existieren. Damit ist gemeint, daß in einem IT-System betroffene Güter und Grundbedrohungen definiert werden müssen. Dieser Grundschutz wird im Grundschutzhandbuch des Bundesamtes für Sicherheit in der Informationstechnik ausführlich beschrieben [BSI 1998]. Die traditionelle Herangehensweise, einen Grundschutz zu beschreiben, erfolgt in vier Schritten ([Grimm 1994], S. 27):

1) Es werden abstrakte Werte identifiziert, die sich auf Informationen (oder Betriebsmittel) ganz allgemein beziehen, aber nicht auf die (ökonomisch oder gesellschaftlich relevanten) Werte der Informationszwecke. Am häufigsten werden Integrität und Vertraulichkeit der Daten und Verfügbarkeit der IT-Systeme als abstrakte Werte genannt. Es gibt aber noch viele andere Werte wie Echtheit, Angemessenheit, Verbindlichkeit oder Nachvollziehbarkeit.

2) Es werden Grundbedrohungen identifiziert, die erfahrungsgemäß für abstrakte Werte bestehen. Unter ihnen sind unautorisierter Datenzugriff, Identitätstäuschung, Zerstörung von Daten, Ausforschen fremder Daten, Dienstverweigerung, nachträgliche Ableugnung.

3) Es werden Grundmechanismen definiert, die zu komplexen Maßnahmen zusammengesetzt werden können. Die Mechanismen wirken gegen Bedrohungen, die erfahrungsgemäß für abstrakte Werte von Informationen bestehen. Unter ihnen sind Zugriffskontrolle, Authentisierung, Verschlüsselung, Beweissicherung.

4) Die konkreten Sicherheitsbelange einer IT-Anwendung werden in einer sogenannten Sicherheitspolitik (Security Policy) beschrieben. Die Vielfalt von Anwendungen und ihrer Sicherheitsbelange drückt sich in einer Vielfalt von Security Policies aus.

Historisch bedeutete IT-Sicherheit vor allem den Schutz der **Vertraulichkeit** im militärischen Bereich. Unter Vertraulichkeit versteht man die Gewährleistung des Schutzes vor Kenntnisnahme von Informationen durch Dritte. Dabei geht es nicht nur um den Schutz vor Kenntnisnahme der Dateninhalte, sondern auch schon das Wissen über die Existenz von Daten soll - soweit möglich - verhindert werden.

Die hohen Anforderungen an dieses Kriterium verdeutlichten den Bedarf der Evaluation von Kriterien. Dies führte zur Definition verschiedener Evaluationskriterien, wie zum Beispiel der „Trusted Computer Systems Evaluation Criteria“ (TCSEC) [DoD 1985] oder der „Information Technology Security Evaluation Criteria“ (ITSEC) [ITSEC 1991] und später der „Common Criteria for Information Technology Security Evaluation“ (CC) [CC 1993]. Alle diese Kriterienkataloge definieren Kriterien, die zur Evaluierung von informationstechnischen Systemen (IT-Systemen) herangezogen werden können. IT-Systeme werden entsprechend der Kriterien evaluiert, das heißt mit einem Zertifikat ausgestattet, das Aussagen über den sicheren Herstellungsprozeß (Assurance) und die Vertraulichkeit und Integrität der schützenswerten Daten (Functionality) trifft.

Folgende Sicherheitsklassen existieren in den oben genannten Katalogen, sie sind in aufsteigender Reihenfolge sortiert. Die zuerst genannte Klasse entspricht dem niedrigsten Sicherheitsniveau, die zuletzt genannte dem höchsten.

- TCSEC: D, C1, C2, B1, B2, B3, A1
- ITSEC: E0, E1, E2, E3, E4, E5, E6
- CC: EAL0, EAL1, EAL2, EAL3, EAL4, EAL5, EAL6, EAL7

Auf die Bedeutung der einzelnen Sicherheitsklassen soll an dieser Stelle nicht eingegangen werden. Es wird auf die Originalliteratur verwiesen.

Im Unterschied zu den TCSEC und den ITSEC enthalten die Common Criteria, zusätzlich zu funktionalen Sicherheitsanforderungen und Sicherheitsanforderungen zur Vertrauenswürdigkeit, vordefinierte Schutzprofile (Protection Profiles), die dazu gedacht sind, anerkannte Lösungen für Standardsicherheitsprobleme anzubieten. Ein

Schutzprofil wird dabei durch die daraus abgeleiteten Sicherheitsvorgaben (Security Target) auf den konkreten Fall eines Produktes zugeschnitten.

Nachdem der Einfluß der Rechner auch im wirtschaftlichen Bereich immer größer wurde, wuchs neben der Bedeutung der Vertraulichkeit, auch die Bedeutung der **Integrität** von Informationen, also der Unverfälschtheit von Informationen. Neben dem Aspekt der **Verfügbarkeit**, der die Situation beschreibt, daß Dienste oder Systeme einem Benutzer zur Verfügung stehen, wenn und wann immer er diese benötigt, sind diese drei Aspekte auch heute noch die Grundpfeiler der IT-Sicherheit [Pfleeger 1997].

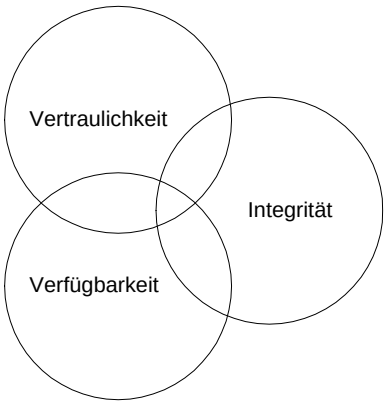


Abbildung 2-4: Vertraulichkeit, Integrität und Verfügbarkeit
Quelle: [Pfleeger 1997], S. 5

Als Bestandteil der Informationstechnik wird Sicherheit wie folgt definiert [ITSEC 1991]:

- **Vertraulichkeit** (Confidentiality) - Schutz vor unbefugter und unbeabsichtigter Kenntnisnahme von Informationen
- **Integrität** (Integrity) - Schutz vor unbefugter und unbeabsichtigter Veränderung von Informationen
- **Verfügbarkeit** (Availability) - Schutz vor unbefugter und unbeabsichtigter Vorenthaltung von Informationen oder Betriebsmitteln

Im folgenden werden alle Aspekte von IT-Sicherheit auf den elektronischen Zahlungsverkehr übertragen. Dies wird dadurch besonders kenntlich gemacht, daß die Absätze, die sich auf den elektronischen Zahlungsverkehr beziehen, eingerückt sind. Neben den bereits erwähnten Aspekten wie Vertraulichkeit, Integrität und Verfügbarkeit, werden auf den folgenden Seiten außerdem die Aspekte Verläßlichkeit, Zurechenbarkeit, informationelle Selbstbestimmung und einfache Handhabung erläutert.

> **Vertraulichkeit** im elektronischen Zahlungsverkehr soll Schutz vor unbefugter Kenntnisnahme von Informationen gewähren [Engel, Lessig 1997]. Für elektronische Zahlungssysteme bedeutet das, daß die Vertraulichkeit eines Transaktionsinhaltes geschützt werden muß.

Alle Beteiligten einer Transaktion haben ein berechtigtes Interesse daran, daß ihre Transaktionsdaten geheim bleiben. Deswegen müssen geeignete Maßnahmen getroffen werden, um das Ausspähen des Transaktionsinhaltes (Sniffing) zu verhindern.

Weiterhin muß die Vertraulichkeit der Daten in den Endgeräten gesichert sein. Die Gefahr des Belauschens durch einen eventuellen Angreifer beschränkt sich nicht nur auf den Verkehr auf den Netzen. Der Rechner, auf dem Transaktionsdaten gespeichert sind, kann ebenso einem Lauschangriff zum Opfer fallen. Deshalb müssen die Daten in den Endgeräten ebenso geschützt werden.

Von besonderer Bedeutung ist die Vertraulichkeit der Kundenidentität. Im Gegensatz zur Bank oder dem Händler, deren Identitäten normalerweise bekannt sind, hat der Kunde ein Interesse daran, in bestimmten Situationen anonym zu bleiben. Der Kunde hat zu Recht den Wunsch, daß seine Transaktionen nicht zu einem Kauf- oder Persönlichkeitsprofil verarbeitet werden. Die Vertraulichkeit der Transaktionsbeziehung ist ebenso von Bedeutung.

Integrität im elektronischen Zahlungsverkehr bedeutet Schutz vor unbefugter und unbeabsichtigter Veränderung von Information [Engel, Lessig 1997]. „Keine Buchung ohne Gegenbuchung“ ist ein elementares Gesetz jeder ordnungsgemäßen Buchführung. Dies bedeutet, daß keine Gutschrift ohne gleichzeitige Belastung erfolgen darf. Gelänge es einem Käufer, einem Verkäufer Geld zu überweisen, ohne daß dieses von seinem Konto abgebucht wird, hätte er Falschgeld erzeugt. Eine sichere Implementierung elektronischen Geldes muß also folgendes verhindern:

- Das Erzeugen mehrerer Kopien derselben elektronischen Geldeinheit
- Das Verändern des Wertes einer zu übermittelnden Geldeinheit
- Das Manipulieren am eigenen Kontostand (zum Beispiel auf einer Chipkarte)

Einem Händler sollte es nur erlaubt sein, von einem Kunden Geld anzunehmen oder abzubuchen, wenn die Bestätigung vom Kunden vorliegt, daß dafür ein Gegenwert erhalten wurde. Prinzipiell muß der Schutz vor Veränderung einer Transaktion gewährleistet sein.

Der Schutz vor Umleitung und vor Manipulation von Zahlungsströmen muß ebenfalls gewährleistet sein. Ist die Übertragung nicht genügend gesichert, so ist es möglich, zum Beispiel die Angabe des Zahlungsempfängers zu verändern. Daher muß auch nach einer gegenseitigen Authentisierung sichergestellt werden, daß kein Zwischenknoten sich in die Transaktion einschalten und sich als einer der Beteiligten maskieren kann. Die Absicherung der Übertragung kann zum Beispiel durch Verschlüsselung oder Signierung des Datenstroms geschehen. Die Umleitung des Zahlungsstroms ist jedoch ein Problem der Verfügbarkeit, da die Transaktionsdaten verloren gehen können.

Verfügbarkeit bei elektronischen Zahlungssystemen bedeutet, daß diese Systeme jederzeit nutzbar sein müssen [Engel, Lessig 1997]. Wird das Internet als Basis für die Übertragung von Zahlungsvorgängen benutzt, kann die Verfügbarkeit nicht gewährleistet werden. Es ist bekannt, daß ein Server durch das Senden einer ausreichenden Menge von Datenpaketen oder durch das Öffnen einer ausreichenden Menge von Verbindungen praktisch lahmgelegt werden kann.

Erfolgt keine Übertragung über das Internet, bedeutet Verfügbarkeit, daß alle sonstigen Medien und Dienste immer zur Verfügung stehen müssen. Wird eine Chipkarte als Speicher- und Zugangsmedium verwendet, muß diese immer benutzt werden können. Ebenso ist die Verfügbarkeit von Bankautomaten ein wesentlicher Faktor, genauso wie die Zahlungsterminals bei den Einzelhändlern. Der Zahlungsvorgang - wie auch immer er abgewickelt werden soll - muß zu jeder Zeit durchgeführt werden können. Die Verfügbarkeit ist jedoch nicht nur bei elektronischen Verfahren ein Problem. Auch bei herkömmlichen Zahlungsverfahren kann nicht immer eine Verfügbarkeit gewährleistet werden. Dies ist jedoch nicht ein Kriterium für die einzelnen Zahlungssysteme, sondern für die dahinter liegende Infrastruktur. Aus diesem Grund wird dieses Kriterium nicht zur Bewertung der einzelnen Zahlungssysteme verwendet.

2.2.2 Holistische Sicherheitsanforderungen

Es gibt jedoch Kritik an dem Begriff der Sicherheit, der nur die Aspekte Vertraulichkeit, Integrität und Verfügbarkeit umfaßt, da wesentliche Aspekte vernachlässigt werden. Für den deutschen Begriff Sicherheit, stellt die englische Sprache zwei verschiedene Begriffe zur Verfügung: **Security** und **Safety**. Beide stellen unterschiedliche Aspekte der „deutschen“ Sicherheit dar. Das Langenscheidt Handwörterbuch hat folgende Übersetzungen:

> *„Security: Sicherheit (Zustand oder Schutz), Sorglosigkeit, Gewißheit.“ ([Langenscheidt 1997], S. 573)*

> *„Safety: Sicherheit, Gefahrlosigkeit, Zuverlässigkeit, Verläßlichkeit.“ ([Langenscheidt 1997], S. 560)*

Beide Begriffe haben eine lange Tradition und kommen aus unterschiedlichen Disziplinen. Während der Begriff Safety aus dem Ingenieursbereich abgeleitet wird, kommt Security aus der Informationstechnik. Beide Anwendungsbereiche hatten bisher wenig Überschneidungspunkte, so daß eine getrennte Betrachtung von Security und Safety stattfand. Inzwischen ist jedoch allgemein anerkannt, daß der deutsche Sicherheitsbegriff beide Aspekte, sowohl Security als auch Safety, in einem ganzheitlichen Ansatz umfassen muß ([Brunnstein 1997] und [Yngström 1996]).

Aus diesem Grund müssen zusätzliche Aspekte von „Safety“ wie **Verläßlichkeit** (Reliability) beziehungsweise **Zuverlässigkeit** mit aufgenommen werden.

Verläßlichkeit für elektronische Zahlungssysteme bedeutet, wie sicher ein System vor Ausfällen und Störungen ist [Engel, Lessig 1997].

Dabei spielt die Transportverläßlichkeit eine Rolle. Wenn der Übertragungsweg selbst keinen Schutz vor Störungen bietet, wie das beim Internet der Fall ist, muß ein zusätzliches Protokoll verhindern, daß Inkonsistenzen auftreten.

Über den Transport hinaus, muß jede einzelne Transaktion verläßlich ausgeführt werden. Dazu bieten sich die ACID-Eigenschaften einer Transaktion an, die erfüllt sein müssen. ACID ist ein Akronym für Atomocity, Consistency, Isolation und Durability. Atomocity bedeutet, daß eine Transaktion entweder ganz oder gar nicht ausgeführt werden soll. Consistency meint, daß ein konsistenter Zustand immer in einen ebensolchen überführt wird. Isolation sagt aus, daß gleichzeitig ablaufende Transaktionen sich nicht beeinflussen dürfen und Durability besagt, daß wenn ein beteiligter Rechner abstürzt, es möglich sein muß, auf einen konsistenten Zustand zurückzufallen [Camp, Sirbu, Tygar 1995].

Gibt es nach der Beschädigung zum Beispiel einer Chipkarte oder einer Festplatte eine Möglichkeit, verlorenes Geld wiederzubekommen? Werden Beweise für abgeschlossene Transaktionen zur Verfügung gestellt?

Ebenso müssen Aspekte der **Zurechenbarkeit** (Accountability) beziehungsweise der **Verantwortlichkeit** betrachtet werden. Ist eine Transaktion oder allgemein eine Handlung einer bestimmten Person oder Institution zurechenbar. Übernimmt diese Partei die Verantwortung für eine Handlung oder kann sie zur Rechenschaft gezogen werden? Eng verwoben mit dieser Thematik ist der Aspekt der Verbindlichkeit [Kossakowski 1999]. In immer größerem Maße werden Rechtsgeschäfte über elektronische Wege abgewickelt. Die Einhaltung von Absprachen, Verträgen und Versprechen muß verbindlich gewährleistet sein. Die Rechtsverbindlichkeit einer Absprache hängt eng mit der Beweissicherung zusammen [Hammer 1994]. Die Anerkennung von Digitalen Signaturen (siehe Kapitel 3) zur Beweissicherung muß über das deutsche Signaturgesetz [BMBF 1997] hinaus geregelt werden.

Die **Zurechenbarkeit**, ist im Bereich des elektronischen Zahlungsverkehrs von erheblicher Bedeutung [Engel, Lessig 1997]. Dazu gehört die Entlarvung von Fälschern. Die Bank und der Händler haben ein Interesse, den Schuldigen einer Fälschung zu entlarven. Fälschungen von Geldeinheiten oder Zahlungsvorgängen sollen durch Zurechenbarkeit verhindert werden, Fälscher erkannt werden.

Weiterhin ist der Beweis einer erfolgten Zahlung von Bedeutung. Ist es möglich, daß der Händler leugnet, eine Zahlung erhalten zu haben oder daß der Kunde vorgibt, eine Zahlung getätigt zu haben? Generell darf ein Kommunikationspartner im Rahmen von elektronischem Informationsaustausch nicht leugnen können, eine Nachricht erhalten oder gesendet zu haben. Die **Nichtabstreitbarkeit** (Non-repudiation) ist von wesentlicher Bedeutung für jeglichen Informationsaustausch.

Muß der Kunde eine Zahlung autorisieren oder reicht der Besitz oder die Kenntnis von identifizierenden Daten aus? Erfolgt eine gegenseitige Identifizierung und Authentisierung? Die Identifizierung und Authentisierung eines Benutzers gegenüber einem System ist eine der ältesten Sicherungsfunktionen, auch wenn zumeist nur schwache, paßwortbasierte Verfahren eingesetzt werden. Oft muß der Benutzer jedoch einfach annehmen, daß er einem authentischen System gegenübersteht. Um das Vertrauen des Benutzers in das gegenüberstehende System zu gewährleisten, gewinnt die gegenseitige Identifizierung und Authentisierung in weltweiten Netzwerken immer größere Bedeutung.

2.2.3 Spezielle Sicherheitsanforderungen

Die Anforderung der Vertraulichkeit der Daten reicht heute nicht mehr aus, weil gerade in offenen Systemen dem Schutz der Privatsphäre durch anonymes oder pseudonymes Handeln eine wachsende Bedeutung zukommt. Dadurch entsteht der Anspruch auf **Anonymität** (Anonymity), beziehungsweise **Pseudonymität** (Pseudonymity). Sowohl Anonymität als auch Pseudonymität gewährleisten die Vertraulichkeit der Benutzeridentität und des Benutzerverhaltens. Während Anonymität die Identifizierung des Benutzers verhindert, ist es bei Pseudonymität möglich, den Zusammenhang zwischen Pseudonym und wahrer Identität von einer vertrauenswürdigen Instanz auf Verlangen des Benutzers in bestimmten Fällen aufdecken zu lassen.

Anonymität ist für elektronische Zahlungssysteme von großer Bedeutung. Solange es keine anonyme elektronische Alternative zum herkömmlichen Bargeld gibt, muß Bargeld für alle Menschen verfügbar sein, weil sonst die Möglichkeit verloren geht, anonyme Zahlungen zu tätigen [Schier 1997]. Diese Alternative ermöglicht ein anonymes Bezahlen im täglichen Leben, wie es seit Jahrhunderten üblich ist. Eine Einschränkung auf ausschließlich elektronisches Geld würde eine starke Beeinträchtigung der freien Entfaltung der Persönlichkeit darstellen.

Die freie Entscheidung zur Auswahl eines Zahlungsmediums muß gewahrt bleiben und darf auch nicht durch gesellschaftliche Zwänge beeinträchtigt werden. Als gesellschaftlicher Zwang ist zum Beispiel die Situation anzusehen, in der nur noch wenige Händler herkömmliches Geld, aber alle Händler elektronisches Geld, akzeptieren würden. Damit wäre ein Kunde gezwungen, elektronisches Geld zu verwenden, da er sonst seine gewünschten Güter nicht (mit Bargeld) anonym bezahlen könnte. Noch schlimmer wäre es, wenn ein Benutzer von herkömmlichem Geld sogar geächtet würde, weil „man herkömmliches Geld nicht mehr nutzt“.

2.2.4 Übergeordnete Sicherheitsanforderungen

Eng verbunden mit den bisher genannten Anforderungen existiert die übergeordnete Anforderung des Rechts auf **informationelle Selbstbestimmung** (Right of Informational Self Determination). Dieses deutsche Recht wurde aus dem Volkszählungsurteil aus dem Jahre 1986 abgeleitet ([Fischer-Hübner, Schier 1996], S. 4ff).

Nach dem Volkszählungsurteil hat jeder das Recht auf informationelle Selbstbestimmung, das sich aus Artikel 1, Abs. 1 (Grundsatz der Menschenwürde) sowie Artikel 2, Abs. 2 (Grundrecht der freien Entfaltung der Persönlichkeit) des Grundgesetzes ableitet. Da die Würde der Person, welche sich in freier Selbstbestimmung als Teil einer freien Gesellschaft entfalten kann, als höchster Rechtswert der verfassungsgemäßen Ordnung gilt, gehört das informationelle Selbstbestimmungsrecht somit zu den höchsten vom Grundgesetz geschützten Werten.

Dieses Grundrecht auf informationelle Selbstbestimmung gewährleistet die Befugnis des einzelnen, grundsätzlich selbst über die Preisgabe und Verwendung seiner persönlichen Daten zu bestimmen. Daß die Garantie dieses Grundrechtes nicht nur für die individuellen Entfaltungschancen des einzelnen, sondern auch für die freiheitlich demokratische Grundordnung von Bedeutung ist, wird vom Bundesverfassungsgericht folgendermaßen begründet:

> *"Wer nicht mit hinreichender Sicherheit überschauen kann, welche ihn betreffende Informationen in bestimmten Bereichen seiner sozialen Umwelt bekannt sind, und wer das Wissen möglicher Kommunikationspartner nicht einigermaßen abzuschätzen vermag, kann in seiner Freiheit wesentlich gehemmt werden, aus eigener Selbstbestimmung zu planen oder zu entscheiden. Mit dem Recht auf informationelle Selbstbestimmung wären eine Gesellschaftsordnung und eine diese ermöglichende Rechtsordnung nicht vereinbar, in der Bürger nicht mehr wissen können, wer was wann und bei welcher Gelegenheit über sie weiß.*
>
> *Wer unsicher ist, ob abweichende Verhaltensweisen jederzeit notiert und als Information dauerhaft gespeichert, verwendet oder weitergegeben werden, wird versuchen, nicht durch solche Verhaltensweisen aufzufallen. Dies würde nicht nur die individuellen Entfaltungschancen des einzelnen beeinträchtigen, sondern auch das Gemeinwohl, weil Selbstbestimmung eine elementare Funktionsbedingung eines auf Handlungs- und Mitwirkungsfähigkeit seiner Bürger freiheitlichen demokratischen Gemeinwesens ist." ([BVerfGE 1986], S. 187ff)*

Das Menschenbild des Grundgesetzes geht jedoch auch von einer Gemeinschaftsbezogenheit und Gemeinschaftsgebundenheit des Menschen aus. Der moderne Rechts- und Sozialstaat benötigt in großem Umfang personenbezogene Daten, um seine vielfältigen Aufgaben fachlich richtig und gerecht erledigen zu können. Zusätzlich fallen in IT-Systemen immer mehr personenbezogene Daten an.

Einschränkungen des informationellen Selbstbestimmungsrechts sind nur im überwiegenden Allgemeininteresse hinzunehmen. Sie bedürfen jedoch einer verfassungsgemäßen gesetzlichen Grundlage, die insbesondere dem Rechtsstaatsgebot der Normenklarheit und dem Verhältnismäßigkeitsgebot genügen muß.

Informationelles Selbstbestimmungsrecht bedeutet für den elektronischen Zahlungsverkehr, daß der Benutzer selbst entscheiden können muß, wem und wann er welche Daten von sich im Rahmen von Zahlungsvorgängen oder Bankgeschäften zur Verfügung stellt, wenn keine gesetzliche Grundlage für die Erhebung vorliegt.

Der Informationsfluß im Rahmen von Bankgeschäften ist weitestgehend geregelt. Bei der Eröffnung eines Kontos bei einer Bank muß der Kunde die sogenannte SCHUFA-Klausel unterschreiben, die besagt, daß die Daten des Kunden an eine zentrale Stelle, die SCHUFA (Schutzgemeinschaft für allgemeine Kreditsicherung) übermittelt werden.

Diese Zustimmung ist theoretisch freiwillig, praktisch gesehen eröffnet jedoch keine Bank ein Konto mit den üblichen Konditionen, wenn diese Zustimmung vom Kunden nicht gegeben wurde. Das Recht auf informationelle Selbstbestimmung ist damit im Bankgeschäftsbereich nicht immer gewährleistet. Der Kunde verzichtet in der Regel auf sein Recht, um die banküblichen Konditionen (Dispositionskredit, Schecks, Scheckkarte, ...) für sein Girokonto zu erhalten. Er gibt damit gezwungenermaßen seine Einwilligung zur Übermittlung seiner Daten an die SCHUFA, weil er sonst keine Chance hätte, ein Konto unter den genannten Konditionen zu eröffnen. Wenn er es vorzieht, unter den Bedingungen kein Konto zu eröffnen, wird er von der Gesellschaft ausgegrenzt.

Der Benutzer hat ein prinzipielles berechtigtes Interesse daran, zu entscheiden, wem er welche Informationen über sich preisgibt. Dies gilt auch bei elektronischen Zahlungsvorgängen oder Bankgeschäften. In vielen Fällen ist jedoch entweder gesetzlich vorgeschrieben, welche Daten benötigt werden oder durch Einwilligung des Benutzers geregelt.

Die freiwillige Entscheidung zur Weitergabe seiner Daten ist meist darauf beschränkt, daß der Benutzer bei Nichtzustimmung der Verarbeitung auf das Zahlungsverfahren oder den entsprechenden Bankdienst verzichten muß. Deshalb ist eine Forderung, die sich aus dem informationellen Selbstbestimmungsrecht ergibt, die Möglichkeit, alternative Zahlungsverfahren zu nutzen, die mit weniger Informationen über den Benutzer auskommen.

Ein weiteres Kriterium, das nicht nur speziell im Sicherheitsumfeld gilt, ist das Kriterium der **einfachen Handhabung** (Ease of Use). Die Benutzung von Anwendungssystemen muß so gestaltet sein, daß sie dem Benutzer leicht fällt und seiner Intuition entspricht. Komplizierte Vorgänge müssen dem Benutzer verborgen bleiben und nach außen hin transparent und durchschaubar wirken. Sicherheitsmaßnahmen dürfen keinen zusätzlichen Aufwand in der Benutzung darstellen, wenn sie einen tatsächlichen Sicherheitsvorteil bieten sollen.

Diese Eigenschaft läßt sich jedoch nicht vollständig durch „richtige“ Sicherheitsmaßnahmen erfüllen. Vielmehr ist das eine Aufgabe der Anwendungsgestaltung. Trotzdem sollte bei der Entwicklung von Sicherheitsmaßnahmen darauf geachtet werden, daß die Vorgänge und Handlungen eines Systems oder einer Organisation so realitätsnah wie

möglich modelliert werden können. Einfache Handhabung wird oft als Widerspruch zu sicheren Systemen angesehen. Dabei schließen sich diese beiden Eigenschaften nicht von vornherein aus. Es ist jedoch richtig, daß Sicherheitsmaßnahmen oft komplexe Vorgänge erfordern. Dies muß jedoch nicht unbedingt bedeuten, daß damit auch umständliche Handlungen verbunden sind. Es zeichnet ein Anwendungssystem aus, wenn komplexe Vorgänge für den Benutzer transparent und durchschaubar bleiben und trotzdem die geforderten Sicherheitsansprüche erfüllt werden.

> **Einfache Handhabung** bedeutet für elektronische Zahlungssysteme, daß die verschiedenen Transaktionen einfach zu handhaben sein müssen, sowie in ihrer Funktionalität durchschaubar und transparent. Es muß für den Benutzer einfach sein, aus einer Vielzahl von elektronischen Zahlungsverfahren ein geeignetes Verfahren auszuwählen und zu nutzen.
>
> So wie es heutzutage keine Schwierigkeit darstellt, bestehende herkömmliche Zahlungsmedien auszuwählen und zu nutzen, darf die Wahl und Nutzung elektronischer Zahlungsmedien kein Problem sein. Heutzutage entscheidet der Benutzer durch den Griff ins Portemonnaie, welches Zahlungsmedium er nutzen möchte. Entweder greift er ins Kleingeldfach oder zieht einen Geldschein heraus oder er wählt seine EC-Karte oder Kreditkarte zur Zahlung, je nachdem wie er bezahlen möchte und welches Medium ihm zur Verfügung steht. Der Griff ins Portemonnaie ist nicht nur einfach, er gewährleistet auch die freie Entscheidung über die Zahlungsmodalität, vorausgesetzt der Benutzer verfügt über alle Zahlungsmedien und der Händler akzeptiert diese.

2.3 Existierende elektronische Zahlungssysteme

Elektronische Zahlungssysteme repräsentieren oder verarbeiten elektronisches Geld (siehe Kapitel 2.1). Sie können nach unterschiedlichen Kriterien aufgeteilt werden. Beispielsweise kann, wie bei herkömmlichen Zahlungssystemen auch, zwischen den verschiedenen Zahlungsinstrumenten unterschieden werden. Neben der Möglichkeit des Bezahlens mit Bargeld, kann man per Scheck, per Kreditkarte oder mit vorausbezahlten Koupons eine Ware oder eine Information bezahlen. Diese Zahlungsmöglichkeiten gibt es auch bei elektronischen Zahlungsverfahren.

Für diese Arbeit wird jedoch eine weitere Unterscheidungsmöglichkeit gewählt, die Einteilung von elektronischen Zahlungssystemen in:

- Kartenbasierte Zahlungssysteme
- Netzbasierte Zahlungssysteme

Prinzipiell muß man zwischen multifunktionalen und nicht-multifunktionalen Anwendungen in Zahlungssystemen unterscheiden. Die Art der Bedrohungen und der Sicherheitsanforderungen unterscheiden sich erheblich zwischen diesen beiden Varianten, deshalb muß auch zwischen Anforderungen an einfache und an multi-funktionale Anwendungen unterschieden werden. Da es in dieser Arbeit um ein Sicherheitsmodell für multifunktionale Chipkartenanwendungen im elektronischen

Zahlungsverkehr geht, wird auf die Betrachtung der Risiken bei multifunktionalen Anwendungen hingearbeitet (siehe Kapitel 2.4). Es ist jedoch notwendig, auch die Risiken einzelner Anwendungen von Zahlungssystemen zu untersuchen, da die Risiken einzelner Systeme in multifunktionalen Systemen kumulativ sind [Schier 1998a/1998b].

Im folgenden werden einzelne Zahlungssysteme exemplarisch vorgestellt und anhand der obigen Anforderungen (Vertraulichkeit / Anonymität, Integrität, Verläßlichkeit, Zurechenbarkeit, informationelle Selbstbestimmung und einfache Handhabung) bewertet.

Im Anschluß werden Anforderungen an multifunktionale Systeme im elektronischen Zahlungsverkehr gestellt.

2.3.1 Kartenbasierte Zahlungssysteme

Um über kartenbasierte Zahlungssysteme zu reden, muß strikt zwischen elektronischer Geldbörse einerseits und den darin verwahrten elektronischen Geldeinheiten andererseits unterschieden werden. Ähnlich wie die traditionelle Geldbörse ist auch die elektronische Geldbörse selbst kein Geld. Die elektronische Geldbörse, entweder in Form einer Chipkarte (siehe Kapitel 3) oder einer speziellen Software, ist Medium zum Verwahren. Dieses Medium dient ebenso als technische Schnittstelle zum Übertragen elektronischer Geldeinheiten im elektronischen Zahlungsverkehr [Bibow, Wichmann 1998].

Bei Chipkarten, die elektronische Geldbörsen enthalten, muß man zwischen Chipkarten unterscheiden, die zählerbasiert arbeiten, und solchen, die auf kryptographisch abgesicherten Werten (elektronische Münzen) arbeiten. Bei zählerbasierten Wertkarten ist das Angriffspotential sehr viel größer als bei Chipkarten, die elektronische Münzen enthalten, der zu erwartende Schaden ist jedoch begrenzt. Das Angriffspotential ist umso höher, je niedriger der finanzielle und organisatorische Aufwand zum Brechen des Systems ist. Bei Chipkarten mit kryptographisch abgesicherten elektronischen Münzen ist zwar das Angriffspotential geringer, jedoch kann der zu erwartende Schaden große Ausmaße annehmen. Der finanzielle Aufwand zum Brechen des Systems ist für beide Chipkarten zwar unterschiedlich, jedoch übersteigt er jeweils den zu erwartenden Schaden, deshalb ist die relative kommerzielle Sicherheit in beiden Fällen ähnlich hoch.

Ein weiteres entscheidendes Merkmal der elektronischen Geldbörse ist, daß sie sich in unmittelbarem Besitz des Verwenders befindet. Dieser muß beim Bezahlen nicht auf sein Konto einer Bank zugreifen, sondern nur auf die elektronische Geldbörse selbst. Beim Bezahlen mit Bargeld ist im herkömmlichen Zahlungsverkehr auch keine Bank beteiligt, die den Zugriff auf das Bargeld in der traditionellen Geldbörse kontrolliert.

Im folgenden werden zwei bekannte Verfahren für elektronische Geldbörsen auf der Basis von Chipkarten diskutiert, zum einen die GeldKarte von der deutschen Kreditwirtschaft und zum anderen Mondex von Mondex International. Diese beiden Verfahren wurden für die Beschreibung im Rahmen dieser Arbeit ausgewählt, da sie am bekanntesten sind und jeweils in ihrem Land (Deutschland beziehungsweise England) die größte Verbreitung haben. Außerdem repräsentieren sie den Stand der Technik in Bezug auf chipkartenbasierte Zahlungssysteme.

2.3.1.1 GeldKarte

Die GeldKarte ist eine elektronische Geldbörse mit Verrechnungseinheiten, die in die EC-Karte der deutschen Kreditwirtschaft integriert ist. Das GeldKarten-Verfahren wird von der gesamten deutschen Kreditwirtschaft getragen. Ende 1997 waren 50 Millionen Karten mit dieser Zusatzfunktionalität ausgestattet [Bibow, Wichmann 1998].

Beschreibung:

Die GeldKarte ist eine Erweiterung der EC- oder Kundenkarte um die Funktionalität der elektronischen Geldbörse. Diese Erweiterung geschieht mit Hilfe eines zusätzlichen Mikrochips auf den Karten, die bisher nur mit einem Magnetstreifen ausgestattet waren. Für Kunden ohne EC-Karte gibt es die sogenannte „White Card“, eine kontoungebundene Chipkarte. In der Praxis wird diese kontoungebundene Chipkarte jedoch kaum ausgegeben [HmbDSB 1995].

Die Beteiligten im GeldKarte-System sind:

- Der Kunde
- Der Händler
- Die Bank des Kunden
- Die Bank des Händlers
- Die Kartenevidenzzentrale
- Die Händlerevidenzzentale

Die Chipkarte enthält eine Karten-ID, das Guthaben, einTransaktionslog über die letzten 15 Bezahlvorgänge und die letzten drei Aufladevorgänge. Das Aufladen der elektronischen Geldbörse erfolgt von seinem Inhaber bei seiner Bank, aus Sicherheitsgründen jedoch nur bis zu einem Betrag von DM 400,-. Dies geschieht entweder an entsprechend ausgerüsteten Geldautomaten unter Eingabe der PIN und mit Online-Autorisierung oder gegen Bargeld in den Bankfilialen. Technisch gesehen ist das Aufladen lediglich das Erhöhen eines im Chip enthaltenen Zählers um den aufgeladenen Betrag. Die Geldeinheiten auf der Chipkarte werden als Kartengeld bezeichnet.

Erfolgt das Aufladen an einem Geldautomaten mit Online-Autorisierung, wird sofort das Kundenkonto belastet und eine Gutschrift über den geladenen Betrag auf ein eigens für diese Zwecke von der Bank eingerichtetes Verrechnungskonto getätigt. Der Saldo dieses Verrechungskontos entspricht dem Gegenwert des Guthabens auf der GeldKarte. Der aufgeladene Betrag wird von der Bank an die Kartenevidenzzentrale weitergeleitet, die den Ladevorgang vermerkt, und der Saldo des Schattenkontos wird erhöht. Die Kartenevidenzzentrale führt für jede GeldKarte ein Schattenkonto und fungiert als Kontrollstelle innerhalb des GeldKarten-Zahlungssystems.

Das Bezahlen (Abbildung 2-5) erfolgt im Gegensatz zum Aufladen immer „offline“ und ohne PIN-Überprüfung oder Leistung einer Unterschrift. Beim Zahlungsvorgang steckt der Kunde seine Chipkarte in das Kartenlesegerät des Händlers. Der zu zahlende Betrag

wird angezeigt und es wird geprüft, ob auf der Chipkarte ausreichend Guthaben vorhanden ist. Ist dies der Fall, wird der Einheitenzähler auf der Chipkarte um den zu zahlenden Betrag reduziert, das Transaktionslog auf der Chipkarte aktualisiert und ein von der Chipkarte zertifizierter Datensatz an das Händlerterminal übergeben und dort gespeichert. Auf der GeldKarte werden die Salden der letzten 15 Transaktionen, sowie die letzten drei Aufbuchungen gespeichert und können mit einem beliebigen Taschenleser ausgelesen werden.

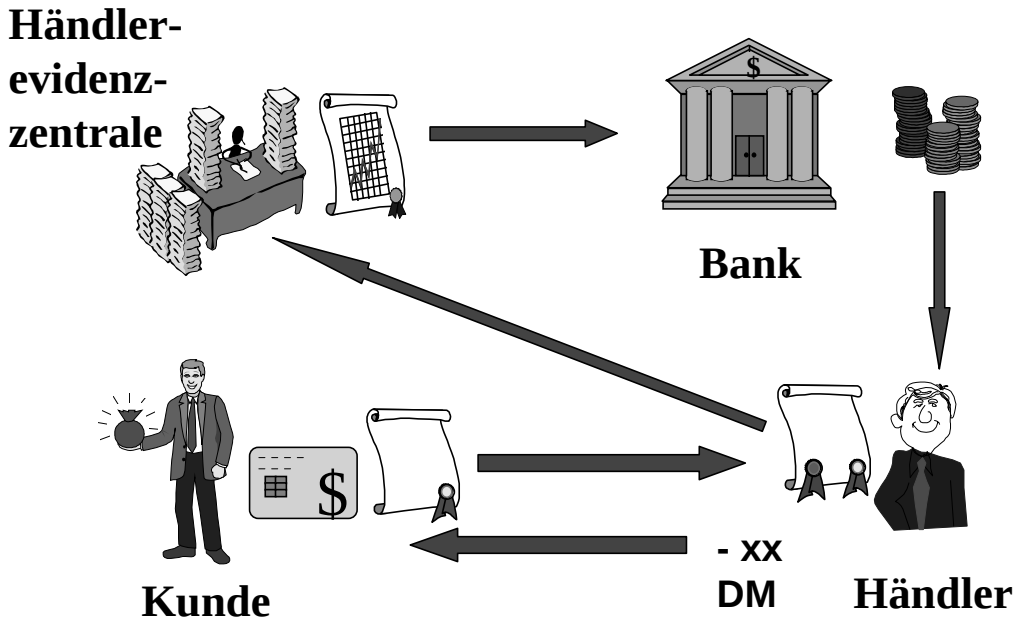


Abbildung 2-5: Bezahlen mit der GeldKarte

Quelle: [Engel, Lessig 1997], S. 107

Das Händlerterminal speichert alle Umsätze eines Tages. Am Ende des Tages werden die Umsätze auf eine spezielle Händlerkarte übertragen, mit deren Hilfe die Umsätze an die jeweilige Händlerevidenzzentrale weitergeleitet werden. Die Händlerevidenzzentrale veranlaßt am Ende eines Tages die Überweisung der Tageseinnahmen des Händlers auf sein Girokonto. Die Transaktionsdaten werden an die Kartenevidenzzentrale weitergeleitet, die den Schattensaldo der zu den GeldKarten gehörenden Schattenkonten um den Zahlungsbetrag vermindert. Dieser Schritt ist in Abbildung 2-5 nicht dargestellt.

Bei den Kartenevidenzzentralen werden alle Einzeltransaktionen gesammelt. Dort werden pro Schattenkonto die Tagessalden gebildet und an die entsprechenden Banken weitergeleitet. Die Kartenevidenzzentrale führt für jede sich im Umlauf befindliche GeldKarte ein Schattenkonto. Die GeldKarten ausgebenden Banken führen, wie oben bereits erwähnt, für jede Chipkarte ein Verrechnungskonto. Der Saldo dieses Verrechnungskontos reduziert sich um die ausgegebenen Beträge, sobald der Händler das empfangene Kartengeld in gewöhnliches Geld umtauscht.

In der Kartenevidenzzentrale werden alle Transaktionsdaten gespeichert. Durch die Führung des Schattensaldos ist ein Ersatz des Kartenguthabens möglich, falls die

GeldKarte defekt sein sollte. Der bis dahin nicht ausgegebene Betrag kann dem Kunden zurückerstattet werden, falls der Defekt der Chipkarte nicht sein Verschulden ist. Im Falle des Verlusts der Chipkarte ist auch das darin enthaltene Guthaben verloren.

In Frankfurt am Main kann seit 1998 mit der GeldKarte in allen städtischen Parkhäusern bezahlt werden. Die zu zahlende Parkgebühr wird aus der Ankunfts- und der Abfahrtszeit berechnet. Die Ankunftszeit wird auf der GeldKarte gespeichert, um bei Verlassen des Parkhauses mit der Abfahrtzeit verrechnet zu werden.

In Ravensburg, Weingarten startete im März 1996 ein Feldversuch mit 30.000 aktiven Chipkarten [Martin 1997]. Seit 1997 statten die Sparkassen bundesweit die EC-Karte mit der Funktionalität der GeldKarte aus. Andere Banken ziehen langsam nach, von einer flächendeckenden Einführung der GeldKarte kann jedoch Ende des Jahres 1998 nicht die Rede sein.

Bewertung:

Die Verwendung von elektronischem Geld sollte es prinzipiell erlauben, die Identität des Kunden vor jedem zu verbergen, solange er sie nicht von sich aus preisgibt. Diese Anforderung steht nicht im Widerspruch zu Verfahren, die es erlauben, einen Kunden zu entlarven, der dasselbe elektronische Geld mehrfach ausgibt.

Möchte man die Vorteile des Bargeldes auf elektronisches Geld übertragen, muß man auch die damit verbundenen Nachteile in Kauf nehmen. So kann Bargeld verloren gehen oder gestohlen werden. Dies muß auch bei elektronischem Geld berücksichtigt werden.

Vertraulichkeit/Anonymität: Laut Aussage der Banken, ist der Kunde anonym gegenüber dem Händler [Heiring 1998]. In der Tat wird beim Zahlungsvorgang nicht der Name des Kunden gespeichert, jedoch die ID seiner Chipkarte. Dem Händler ist es zumindest möglich, die mit derselben Chipkarte getätigten Zahlungen zusammen zu sortieren [Bibow, Wichmann 1998]. Der Händler könnte auch das Transaktionslog aus der Chipkarte auslesen. Da dort jedoch nur der Betrag und das Datum, nicht jedoch der beteiligte Händler gespeichert sind, eignen sich diese Informationen wenig zur Profilerstellung.

Die Anonymität des Kunden gegenüber den Evidenzzentralen ist nicht gegeben [AgV 1998]. Dort werden alle Transaktionsdaten gespeichert, insbesondere die Kontonummer des Karteninhabers und die Kontonummer des Händlers, um die einzelnen Salden an die Banken der jeweiligen Verrechnungskonten zu senden. Der Bezug zu Kontonummer und Name des Kunden kann auf diese Weise leicht ermittelt werden und ist für Reklamationsfälle auch vorgesehen.

Den Banken gegenüber ist der Kunde auf keinen Fall anonym, da dort die Karten-ID in einem direkten Zusammenhang mit der Kontonummer und damit mit dem Karteninhaber steht.

Die GeldKarte erfüllt also nicht die Anforderungen an traditionelles Bargeld, sondern stellt eine Einschränkung bezüglich der Vertraulichkeit und Anonymität für den Kunden dar. Eine weitere Einschränkung ist die Tatsache, daß

kontoungebundene Chipkarten, sogenannte White Cards, zwar prinzipiell vorgesehen sind, jedoch in der Praxis nicht eingesetzt werden. Die Nutzung dieser Chipkarten könnte Anonymität oder zumindest Pseudonymität gewährleisten und wird von den Datenschutzbehörden energisch gefordert [HmbDSB 1995/1998].

Integrität: Das mehrfache Ausgeben von Geldeinheiten wird bei der GeldKarte durch relativ sichere Hardware und Plausibilitätsprüfungen der Kartendaten bei jeder Transaktion erschwert. Die eigentliche Geldtransaktion findet bei der Bank im Nachhinein und nicht während des Zahlungsvorgangs statt. Somit ist eine Buchung ohne Gegenbuchung praktisch ausgeschlossen.

Technisch gesehen ist es prinzipiell möglich, das Händlerterminal so zu manipulieren, daß es einen anderen Betrag anzeigt, als tatsächlich von der Chipkarte abgebucht wird. Da der Betrag nicht explizit freigegeben wird und die Anzeige des Betrages auf dem Händlerterminal erfolgt, hat der Kunde keine Kontrolle über die abgebuchten Beträge. Eine Möglichkeit, dieses Problem zu umgehen, wäre ein vertrauenswürdiges Gerät unter der Kontrolle des Benutzers, das garantiert, daß nur die korrekten Beträge und nicht mehr abgebucht wird [Dethloff 1997]. Diesem Problem gegenüber zu halten ist jedoch die Tatsache, daß alle Transaktionsdaten bei den Evidenzzentralen gespeichert werden und dazu verwendet könnten, einen Mißbrauch aufzudecken.

Verläßlichkeit: Die Kreditinstitute versprechen bei unverschuldeter Beschädigung der GeldKarte die Erstattung des verlorengegangenen Betrages. Da alle bereits ausgegebenen Beträge gespeichert werden, sollte dies praktisch kein Problem darstellen. Die Definition von „unverschuldet“ ist jedoch in diesem Zusammenhang nicht zu finden.

Zurechenbarkeit: Da die GeldKarte der Bank gegenüber nicht anonym ist, sollte eine Entlarvung von Fälschern kein Problem darstellen. Gelänge es einem Kunden, seine GeldKarte so zu manipulieren, daß sie sich zum Beispiel nach jeder Transaktion wieder auflädt, so kann dies in den Evidenzzentralen festgestellt werden.

Der Beweis einer erfolgten Zahlung kann bei der GeldKarte entweder durch Erhalt einer physischen Quittung oder Reidentifikation einer Transaktion durch die Evidenzzentralen und die Banken erfolgen. Eine Autorisierung der Zahlung erfolgt nicht, da der Kunde seine Chipkarte nur in das Händlerterminal steckt und der zu zahlende Betrag ohne aktive Einwilligung des Kunden abgebucht wird.

Bevor eine Zahlung erfolgt, authentisieren sich Chipkarte und Händlerterminal gegenseitig. Es wird jedoch nicht geprüft, ob der Karteninhaber auch der rechtmäßige Besitzer der Chipkarte ist. Theoretisch ist es möglich, mit einer gestohlenen oder gefundenen Chipkarte einzukaufen, da keine Authentisierung der Personen stattfindet.

Informationelle Selbstbestimmung: Theoretisch hat der Kunde die Wahl zwischen kontogebundenen und kontoungebundenen Chipkarten. Hätte ein Kunde beide Chipkarten zur Verfügung, könnte er entscheiden, wann er wem seine Identität preisgibt. Durch die Tatsache, daß die kontoungebundenen Chipkarten in der Praxis nicht verfügbar sind, wird sein informationelles Selbstbestimmungsrecht stark eingeschränkt. Ein weiteres Problem ist die Tatsache, daß er einer Zahlung nicht aktiv einwilligt, sondern durch das Einführen der Chipkarte in das Händlerterminal, die Zahlung automatisch durch ihn autorisiert ist.

Einfache Handhabung: Die Benutzung der Chipkarte ist sehr anwendungsfreundlich konzipiert. Der Bezahlvorgang läuft für den Kunden sehr einfach ab, da keine Authentisierung und Autorisierung erfolgt. Die Zahl der Kartenakzeptanzstellen in Deutschland ist jedoch noch nicht so hoch, daß man von einer breiten Nutzung sprechen kann. Über die Akzeptanz der GeldKarte von Seiten der Kunden kann zur Zeit noch keine Aussage gemacht werden.

2.3.1.2 Mondex

Im Juli 1996 wurde Mondex International, ein Konsortium von 17 Großbanken, gegründet [Mondex 1996], die in 20 Ländern zusammenarbeiten. Dieses Zahlungsverfahren ist sowohl auf der Basis von Chipkarten als auch im Internet einsetzbar. Es wird an dieser Stelle in seiner Rolle als Kartensystem erläutert, wobei keine wesentlichen Änderungen für Mondex als Internetzahlungsverfahren gelten.

Beschreibung:

Geld kann in bis zu fünf Währungen auf einer Chipkarte gespeichert werden. Dabei ist für jede Währung ein Höchstlimit vorgesehen, bis zu dem die Chipkarte aufgeladen werden kann. Das Währungskonzept ist so aufgebaut, daß jederzeit neue Währungen (zum Beispiel der Euro) definiert werden können. Die Chipkarte wird an Geldautomaten mit einer Online-Autorisierung aufgeladen.

Verbindet man zwei Chipkarten mit einem geeigneten Gerät, ist es möglich, Geld von einer Chipkarte auf die andere zu transferieren. Die beiden Chipkarten authentisieren sich gegenseitig (mit Hilfe kryptographischer Verfahren) und aktualisieren anschließend ihren neuen Guthabenstand.

Die Beteiligten bei Mondex sind:

- Der Kunde
- Der Händler
- Die Bank des Kunden
- Die Bank des Händlers

Die Chipkarte enthält neben den Guthaben der verschiedenen Währungen eine Karten-ID, eine Bezeichnung des Kunden, sowie ein Transaktionslog über die letzten zehn

Bezahlvorgänge. Eine Transaktion wird mit Kaufdatum, Betrag und Händleridentifikation auf der Chipkarte gespeichert. Wie bei der GeldKarte existiert ein kleines Lesegerät, mit dem der Guthabenstand und alle gespeicherten Transaktionen (die letzten zehn) ausgelesen werden können. Jede Transaktion wird auf der Händlerkarte zur späteren Abrechnung gespeichert, zusätzlich speichert der Händler in seinem Händlerterminal die letzten 300 Transaktionen mit Kartennummer, Betrag und Datum. Der Händler kann die Kartennummer jedoch nicht dem Karteninhaber zuordnen, außer er notiert sich den auf der Chipkarte aufgedruckten Namen des Kundens.

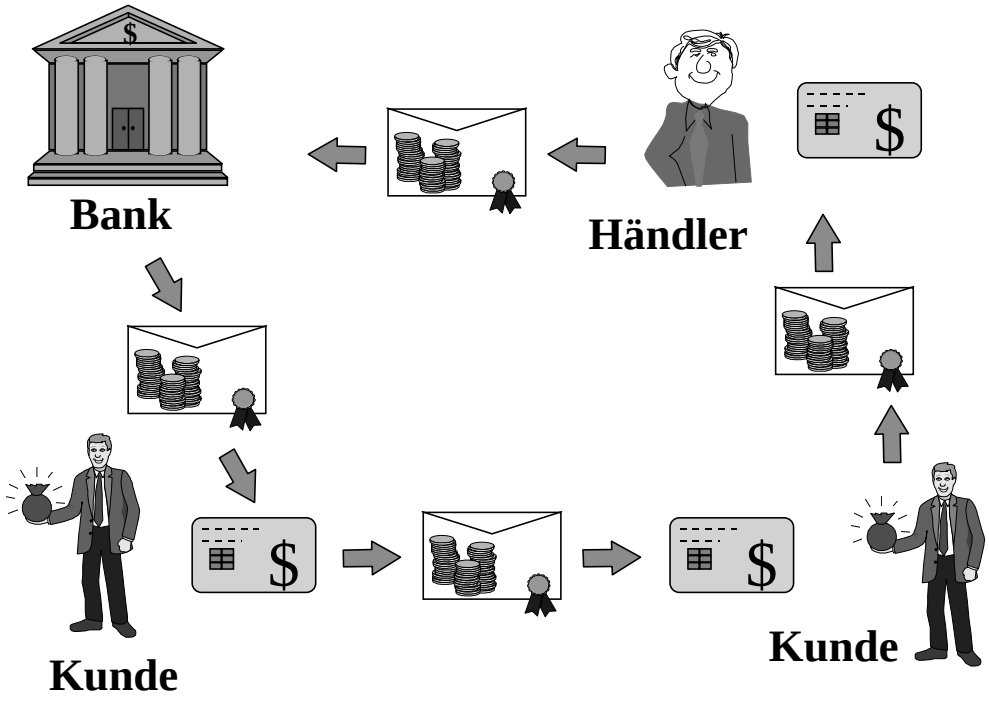


Abbildung 2-6: Mondex
Quelle: [Engel, Lessig 1997], S. 112

Eine PIN kann vom Karteninhaber dazu verwendet werden, die Chipkarte gegen unbefugte Nutzung zu sperren, sozusagen „abzuschließen“. Ist die Chipkarte geöffnet, findet keine weitere Authentisierung statt. Geht dem Karteninhaber die Chipkarte im „geöffneten“ Zustand verloren, kann jeder andere das auf der Chipkarte gespeicherte Geld ausgeben. Wird jedoch eine verlorene Chipkarte bei der Bank abgegeben, kann die Bank aufgrund der Karten-ID den Besitzer ausfindig machen und ihm sein Geld zurückerstatten.

Um Betrug zu verhindern und Angriffe zu erkennen, werden Transaktionsdaten und typische Verhaltensmuster erhoben. Da jeder Transaktion eine eindeutige Nummer zugeordnet ist, soll das doppelte Auftreten einer Nummer leicht erkannt werden können. Aufgrund des gespeicherten Verhaltensmusters können Abweichungen leicht erkannt werden. Auf diese Weise erregen einmalige Abhebungen an ungewöhnlichen Orten, regelmäßige Abhebungen aus „ungeklärten“ Quellen, sowie ein hoher Durchschnitt von Abhebungen nahe am Limit der Chipkarte Verdacht.

Pilotversuche gibt es in Swindon und Devon in Großbritannien, San Francisco in den USA, Hong Kong in China und Guelph in Kanada [Engel, Lessig 1997]. Der aktuelle Stand der Projekte wird auf der Homepage von Mondex International laufend aktualisiert [Mondex 1998].

Bewertung:

Ebenso wie die GeldKarte, soll die Mondex-Karte Bargeld ersetzen. Auch hier gilt es zu prüfen, ob die Eigenschaften von Bargeld bei dem elektronisch nachgebildeten Geld erhalten bleiben und die Sicherheitsanforderungen (siehe Kapitel 2.2) erfüllt werden.

Vertraulichkeit/Anonymität: Der Kunde ist dem Händler nur durch seine Karten-ID bekannt, kann also pseudonym einkaufen. Der Händler kann jedoch theoretisch die auf der Chipkarte aufgedruckten Informationen, wie den Namen, für eigene Zwecke notieren. Der Bank gegenüber ist der Kunde nicht anonym, da dort ein direkter Bezug zwischen Karten-ID und dem Karteninhaber besteht.

Die Vertraulichkeit der Transaktionsdaten ist nicht gegeben, da eine „geöffnete" Chipkarte es prinzipiell erlaubt, die letzten zehn Transaktionen auszulesen. Dies ist ohne großen technischen Aufwand auch für den Händler möglich, der auf diese Weise Kaufprofile für die einzelnen Chipkarten erstellen kann. Im Gegensatz zur GeldKarte ist in dem Transaktionsdatensatz neben Kaufdatum und Betrag auch der Händler vermerkt, bei dem die Ware gekauft wurde. So kann ein Händler feststellen, ob vorher bei der Konkurrenz gekauft wurde.

Integrität: Mondex erlaubt Transaktionen zwischen Privatpersonen, wobei die Bank nicht involviert ist. Dies würde theoretisch eine Buchung ohne Gegenbuchung unterstützen, was zur Folge hätte, die Geldmenge im System zu erhöhen.

Da sich bei Mondex das Geld tatsächlich auf der Chipkarte befindet, wäre es theoretisch möglich, Falschgeld zu erzeugen und in Umlauf zu bringen. Dazu müßten jedoch die verwendeten kryptographischen Verfahren gebrochen und die physischen Schutzmaßnahmen der Chipkarte umgangen werden.

Ähnlich wie bei der GeldKarte, ist es auch bei Mondex prinzipiell möglich, die Anzeige des Händlerterminals so zu manipulieren, daß ein anderer Betrag angezeigt wird, als tatsächlich abgebucht wird.

Verläßlichkeit: Ebenso wie bei der GeldKarte, verspricht Mondex, das Geld auf einer beschädigten Chipkarte zu erstatten, wenn sich der Wert feststellen läßt. Wenn dies nicht der Fall ist, gibt es von Mondex keine eindeutige Aussage darüber, ob das Geld erstattet wird. Da keine Schattenkonten geführt werden und der Geldtransfer zwischen Privatpersonen (ohne Einmischung der Bank) erlaubt ist, ist nicht deutlich, wie die versprochene Erstattung ablaufen soll.

Zurechenbarkeit: Eine manipulierte Chipkarte, die sich immer wieder auflädt, kann bei Zahlungen zwischen Privatpersonen nicht erkannt werden, da die Bank als einzige die Manipulation erkennen kann, aber nicht in diesem Zahlungsstrom

beteiligt ist. Wird diese Chipkarte jedoch an einem mit der Bank verbundenen Terminal benutzt, kann der Mißbrauch entdeckt werden.

Der Kunde kann eine Zahlung nur mit einer physischen Quittung nachweisen oder mit dem Transaktionslog der letzten zehn gespeicherten Zahlungen auf der Chipkarte.

Ähnlich wie bei der GeldKarte authentisieren sich bei Mondex die beteiligten Chipkarten gegenseitig, jedoch nicht die beteiligten Personen.

Informationelle Selbstbestimmung: Da es bei Mondex nur eine Art der Chipkarten gibt, kann der Kunde sich nur entscheiden, ob er die Chipkarte nutzt oder nicht. Er kann nicht zwischen einer anonymen / pseudonymen oder identifizierenden Chipkarte wählen. Da die Mondex-Karte ebenso wie die GeldKarte Bargeld ersetzen soll, ist das informationelle Selbstbestimmungsrecht des Kunden durch die prinzipiell fehlende Anonymität gefährdet.

Einfache Handhabung: Aus den Berichten über verschiedene Pilotprojekte geht hervor, daß die Handhabung einfach und schnell ist und auf keinen Widerstand in der Bevölkerung stößt [Mondex 1998].

2.3.2 Netzbasierte Zahlungssysteme

Neben den kartenbasierten Zahlungssystemen werden nun netzbasierte Zahlungssysteme beschrieben. Das Medium Chipkarte steht hier nicht im Vordergrund, da die Geldeinheiten entweder bei der Bank liegen oder auf dem Rechner des Kunden. Die Chipkarte wird nicht als Geldaufbewahrungsmittel, sondern als Zugangsmedium zu den netzbasierten Zahlungssystemen genutzt werden. Im folgenden werden das Verfahren Millicent von Digital Equipment Corporation, SET von Mastercard und VISA und Ecash von der Firma DigiCash vorgestellt. Diese drei Verfahren repräsentieren das gesamte Spektrum von netzgestützten Zahlungsverfahren. Dies reicht von Verfahren für geringe Beträge (Millicent), über Verfahren zur Kreditkartenzahlung (SET) bis hin zu digitalem Bargeld (Ecash), das über Netze ausgegeben werden kann. Millicent ist ebenfalls ein Verfahren zur Zahlung mit Koupons, also vorbezahlten Geldeinheiten.

2.3.2.1 Millicent

Das Forschungslabor der Digital Equipment Corporation (DEC) hat 1995 ein Verfahren für Transaktionen geringen Betrages entwickelt [Millicent 1995]. Millicent ist ausschließlich für Summen zwischen 0,1 US-Cent und fünf US-Dollar konzipiert.

Beschreibung:

Die Idee von Millicent beruht auf der Einführung einer neuen Währung namens Scrip. Diese Währung gilt nur für einen bestimmten Händler und seine Produkte. Um zu verhindern, daß der Kunde bei allen Händlern, bei denen er etwas erwerben möchte, ein Konto in der Währung des jeweiligen Händler-Scrip einrichten muß, werden Makler eingerichtet. Jeder Makler hat seine eigene Währung, das sogenannte Makler-Scrip. Der

Makler hat eine Vertrag mit einem oder mehreren Händlern und verkauft dem Kunden das jeweilige Händler-Scrip.

In dem Verfahren sind demnach vier beteiligte Parteien vertreten:

- Der Kunde
- Der Händler
- Der Makler des Kunden
- Der Makler des Händlers

Bevor der Kunde bei einem Händler etwas kaufen kann, besorgt er sich in ausreichender Menge Makler-Scrip seines Hausmaklers (Abbildung 2-7). Bei diesem Makler hat der Kunde ein Konto in der Währung des Makler-Scrip angelegt. Steht nun eine Kauftransaktion bevor, wechselt der Kunde einen Teil seines Makler-Scrip in das jeweilige Händler-Scrip. Damit kann der Kunde nun bei dem Händler einkaufen. Der Händler reicht die eingenommenen Händler-Scrip bei dem entsprechenden Makler ein und bekommt so sein Geld auf sein Konto überwiesen.

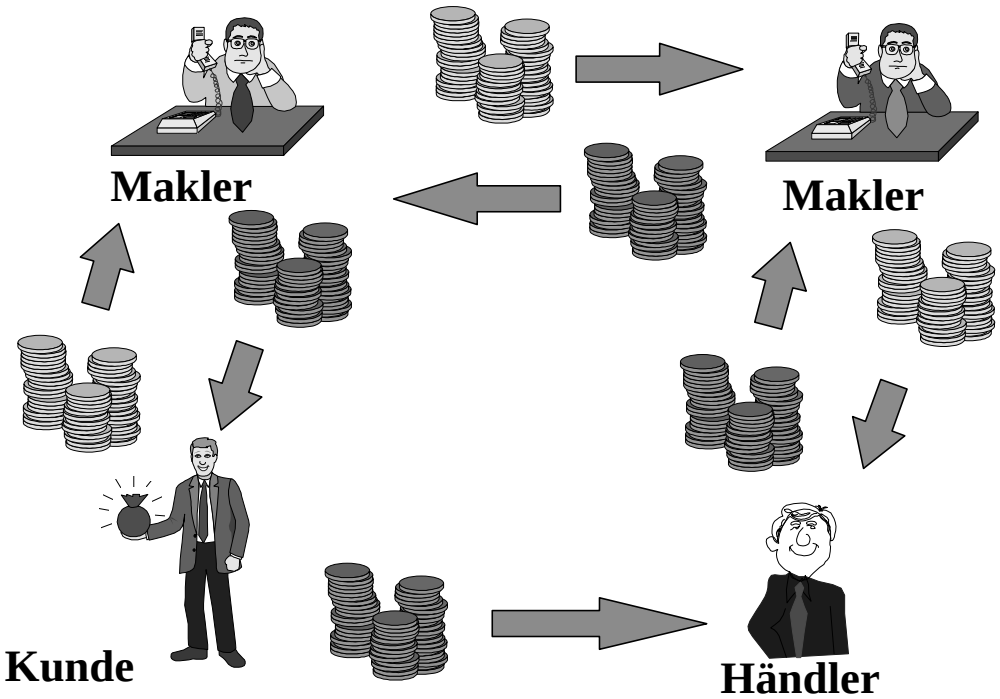


Abbildung 2-7: Millicent

Quelle: [Engel, Lessig 1997], S. 29

Die Sicherheitsüberlegungen für Millicent basieren auf der Annahme, daß es sich nur um Transaktionen geringen Wertes handelt, so daß ein Gewinn durch Betrug sehr beschränkt wäre. Millicent verwendet als Kryptoalgorithmen die schnell berechenbaren Einweg-Hash-Funktionen wie zum Beispiel MD5 (siehe Kapitel 4.3). Das hat den

Vorteil, daß Transaktionen bei gleicher Rechenleistung schneller verarbeitet werden können als mit Kryptoalgorithmen wie DEA oder RSA (siehe Kapitel 4.1 und 4.2).

Jeder Händler hat seine eigene Währung, sein eigenes Scrip. Dieses Scrip enthält eine Seriennummer, um zu verhindern, daß es mehrmals ausgegeben werden kann. Es existiert eine digitale Signatur (siehe Kapitel 4.4), um Manipulationen und Fälschungen zu verhindern. Weiterhin enthält Scrip personenbezogene Daten über den Kunden, so daß ein Betrug seitens des Kunden und des Händlers sofort auffallen würde. Ein Betrug seitens des Maklers würde sofort auffallen, da die Kunden- und Händlersoftware unabhängig voneinander das Scrip prüfen und den Kontostand verwalten.

Bei Benutzung signiert der Kunde jedes Scrip mit einem auf das jeweilige Scrip bezogenen geheimen Schlüssel. Sobald der Händler das Scrip vom Kunden erhält, führt er zwei Schritte durch. Zunächst berechnet er selbst noch mal die Signatur und vergleicht sie mit der mitgeschickten Signatur. Sind die beiden Signaturen identisch, handelt es sich um gültiges Scrip. Als nächstes überprüft er die Seriennummer des Scrip. Falls diese Nummer in einer von ihm geführten Liste über bereits eingelöstes Scrip auftaucht, verweigert der Händler die Annahme.

Um die Verwaltung der Seriennummern für den Händlern etwas zu erleichtern, ist jedes Scrip mit einem Verfallsdatum versehen. Ist das Verfallsdatum überschritten, ist das Scrip ungültig und der Händler kann die Seriennummer aus seiner Liste streichen. Für den Kunden bedeutet das, daß er nie zuviel Scrip auf Vorrat haben darf, weil sonst für ihn die Gefahr besteht, daß das Scrip seine Gültigkeit verliert, wenn er es nicht rechtzeitig ausgibt. Der Wert des abgelaufenen Scrip wird dem Kunden nicht ersetzt.

Das Verfahren Millicent beinhaltet drei verschiedene Varianten bezüglich Sicherheit, Geheimhaltung und Komplexität.

Die einfachste Variante nennt sich **Scrip in the Clear**. Sie ist sehr effizient bezüglich der Übertragungsgeschwindigkeit, da keinerlei Verschlüsselung verwendet wird. Der Kunde schickt sein Scrip unverschlüsselt an den Händler, dieser schickt sein Wechselgeld ebenso unverschlüsselt zurück. Ein Angreifer könnte das Scrip auf dem Übertragungsweg abfangen und ausgeben. Versucht der eigentliche Besitzer später, dieses Scrip selbst auszugeben, ist es bei dem Händler schon gespeichert und wird nicht mehr angenommen. In dieser Variante werden keine digitalen Signaturen verwendet, um die Authentizität (siehe Kapitel 4.3) des Scrip zu gewährleisten.

Die Variante **Secure without Encryption** verzichtet auf den Einsatz von Verschlüsselungsalgorithmen, so daß die Geheimhaltung der übertragenen Nachrichten nicht gewährleistet ist. Wie weiter oben bereits beschrieben, signiert jedoch der Kunde sein Scrip bei der Benutzung und schickt dem Händler die Signatur. Dieser überprüft diese Signatur durch Berechnung und Vergleich derselben und kann somit feststellen, ob das Scrip wirklich zu dem Kunden gehört.

Die Variante **Private and Secure** verwendet einen geheimen, nur Kunde und Verkäufer bekannten, Schlüssel. Dieser Schlüssel und die Verwendung eines schnellen symmetrischen Algorithmus lassen einen sicheren Übertragungskanal entstehen. Sicher

bedeutet in diesem Zusammenhang, daß Vertraulichkeit, Integrität und Authentizität der übertragenen Nachricht gesichert sind.

Im Forschungslabor der Digital Equipment Corporation wurde bereits eine Millicent-Anwendung für Netzwerk-Firewalldienste implementiert ([Engel, Lessig 1997], S. 26).

Bewertung:

Da Millicent ein Verfahren für das Bezahlen von geringen Beträgen ist, sind die Sicherheitsanforderungen nicht so hoch wie bei Verfahren für höhere Beträge. Die Anforderung der Anonymität ist bei diesem Verfahren jedoch sehr wichtig, da gerade die Transaktionen vieler geringer Beträge ein detailliertes Kaufprofil eines Kunden ergeben.

Vertraulichkeit/Anonymität: Der Kunde ist gegenüber dem Makler nicht anonym. Aufgrund der Tatsache, daß sich im Scrip die ID des Kunden befindet, ist der Kunde auch gegenüber dem Händler nicht anonym. Von Millicent wird zur Wahrung der Privatsphäre empfohlen, als Kunden-ID ein Pseudonym zu verwenden. Der Händler kann jedoch trotzdem die unter diesem Pseudonym getätigten Transaktionen auswerten. Es besteht auch zwischen Makler und Händler keine Anonymität, da beide eine Geschäftsbeziehung pflegen.

Der Transaktionsinhalt kann bei Millicent durch die Wahl des Protokolls **Private and Secure** geschützt sein. Bei den beiden anderen Protokollen wird keine Verschlüsselung eingesetzt, so daß der Schutz der Vertraulichkeit nicht gewährleistet ist.

Integrität: Der Makler hätte im Millicent-Verfahren prinzipiell die Möglichkeit, Falschgeld zu erzeugen, das heißt ein und dasselbe Händler-Scrip mehrfach auszugeben. Da der Händler jedoch Listen über bereits eingelöstes Scrip führt, würde eine Mehrfachausgabe schnell auffallen. Der Makler würde seinen „guten Ruf“ verlieren.

Es ist nachteilig, daß der Händler die Möglichkeit hat, Geld anzunehmen, ohne Waren zu liefern oder das empfangene Geld für ungültig zu erklären, da er die Listen über bereits eingelöstes Geld verwaltet.

Eine Umleitung oder Manipulation von Zahlungsströmen ist kaum denkbar, da das Scrip zum einen händlerbezogen ist und zum anderen vom Kunden signiert wird. Nur in der **Scrip in the Clear** Variante wird auf sämtliche Verschlüsselung verzichtet, damit wäre prinzipiell die Möglichkeit zur Manipulation gegeben.

Verläßlichkeit: Bei Millicent werden keine Maßnahmen zum Schutz der Verläßlichkeit erwähnt. Es ist zu vermuten, daß bei einer Beschädigung der Endgeräte oder des Übertragungsweges die betroffenen Geldeinheiten verloren sind.

Zurechenbarkeit: Der Beweis einer Zahlung ist nicht immer möglich. Da prinzipiell Geld auch gestohlen sein kann, kann ein Nachweis einer Zahlung nicht immer erfolgen. Lediglich die beiden Varianten, die signierte Geld-

einheiten verwenden, können mit Hilfe der protokollierten Transaktionen eine Zahlung nachweisen. Wobei der Nachweis nicht als rechtlich anerkanntes Beweismittel zu verstehen ist.

Die Autorisierung einer Zahlung durch den Kunden erfolgt durch das Erzeugen seiner Signatur, jedoch nicht bei **Scrip in the Clear**. Eine gegenseitige Authentisierung findet nicht statt, höchstens eine indirekte Identifizierung, indem der Händler bei Erhalt des Scrip die Signatur neu berechnet und somit feststellen kann, ob es sich um gültiges Geld eines bestimmten Kunden handelt. Der Händler seinerseits kann sein Wechselgeld ebenfalls signieren, um sich dem Kunden gegenüber zu identifizieren.

Informationelle Selbstbestimmung: Der Ansatz im Millicent-Verfahren, drei verschiedene Protokolle bezüglich Geheimhaltung, Sicherheit und Komplexität anzubieten, stellt dem Kunden eine Wahlmöglichkeit zur Verfügung. Er hat jedoch keine Wahl bezüglich der Anonymität, da alle drei Varianten nicht anonym sind. Steht dem Kunden also kein weiteres anonymes Zahlungsverfahren zur Verfügung, hinterläßt er bei jeder Transaktion personenbezogene Datenspuren.

Einfache Handhabung: Aus den zur Verfügung stehenden Unterlagen geht nicht hervor, in welcher Weise die Benutzung erfolgt und welche Komplexität damit verbunden ist.

2.3.2.2 SET-Anwendung

Von den Kreditkartenfirmen Mastercard und VISA wurde im Februar 1996 ein gemeinsames Verfahren, SET (Secure Electronic Transactions) zum Bezahlen mit Kreditkarten im Internet entwickelt [SET 1996].

Beschreibung:

Dieses Verfahren hat zum Ziel, erstens daß sich Kunde und Händler gegenseitig authentisieren können, zweitens soll eine Zahlung nicht durch eine der Parteien oder einen Außenstehenden manipuliert werden können und drittens soll der Inhalt einer Transaktion vertraulich bleiben.

Die Beteiligten einer Transaktion bei SET sind:

- Der Kunde
- Der Händler
- Die Bank des Kunden
- Die Zertifizierungsinstanz
- Das Payment Gateway, Teil der Bank des Händlers

Alle Beteiligten an einer Transaktion müssen einander vertrauen. Um sich gegenseitig authentisieren und damit vertrauen zu können, haben Kunde und Händler jeweils ein Zertifikat von einer vertrauenswürdigen Instanz, der Zertifizierungsinstanz bekommen

(siehe Kapitel 4.5). Diese Zertifizierungsinstanz ist in eine Zertifizierungshierarchie eingebettet, in der jeweils eine Instanz der nächst niedrigen ihre Authentizität durch ein Zertifikat bescheinigt. Die Zertifikate hängen wie eine Kette zusammen.

Um ein Zertifikat zu überprüfen, ist nötig, nachzuprüfen, ob es mit dem Schlüssel der nächst höheren Instanz unterschrieben wurde. Das Zertifikat der obersten Instanz kann allerdings nicht geprüft werden, da es mit dem Schlüssel dieser obersten Instanz unterschrieben wurde. Dieser Instanz müssen alle Beteiligte vertrauen. Vor Beginn einer Transaktion tauschen Kunde und Händler ihre Zertifikate aus und überprüfen sie gegenseitig, um sicherzustellen, daß sie den jeweils gewünschten Kommunikationspartner vor sich haben.

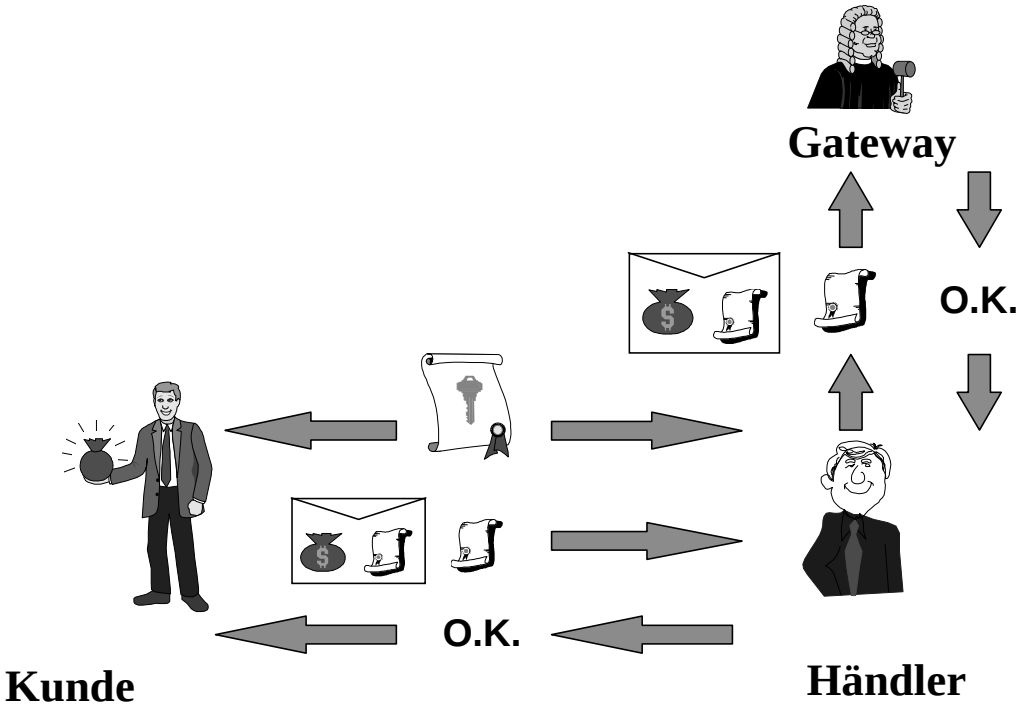


Abbildung 2-8: Secure Electronic Transactions (SET)

Quelle: [Engel, Lessig 1997], S. 75

Nach dieser Überprüfung schickt der Kunde dem Händler die Bestellinformation und die Zahlungsinformation. Die Bestellinformation enthält Angaben über die gewünschte Ware mit entsprechendem Preis, die der Kunde zu kaufen wünscht, wenngleich die eigentliche Verhandlung des Transaktionsinhaltes aus dem Protokoll ausgelagert wurde. Die Zahlungsinformation enthält Angaben über die Kreditkarte des Kunden, wie Kreditkartennummer und Ablaufdatum.

Der Kunde hat nun einerseits das Interesse, daß der Händler seine Kreditkarteninformation nicht erhält, und andererseits, daß die Bank keine Information über die gekaufte Ware und sonstige Vereinbarungen mit dem Händler erhält. Allerdings möchte er, daß die Bank nur dann das Geld auszahlt, wenn der Händler mit den Bedingungen einverstanden ist. Dies wird durch das Prinzip der doppelten digitalen Signatur gelöst [Schneier 1996].

Bei einer einfachen digitalen Signatur wird ein Hashwert über die Nachricht gebildet und dieser wird mit dem geheimen Schlüssel des Unterzeichners verschlüsselt (siehe Kapitel 4.4). Bei doppelten Signaturen werden die Hashwerte der Einzelnachrichten (Bestell- und Zahlungsinformation) gebildet, die Kombination dieser beiden Werte wird erneut gehasht und danach verschlüsselt. Der Kunde schickt nun dieses Ergebnis, sowie zusätzlich die Bestellinformationen an den Händler.

Der Händler schickt die vom Kunden erhaltene Signatur an das Payment Gateway weiter und berechnet selbst einen Hash über die Bestellinformation, die er auch an das Payment Gateway schickt. Das Payment Gateway kann überprüfen, ob die beiden Hashwerte über die Bestellinformation gleich sind, ob also der Kunde und der Händler mit den gleichen Bedingungen einverstanden sind.

Die Zahlungsinformation hat der Kunde mit dem öffentlichen Schlüssel des Payment Gateways verschlüsselt. Somit kann das Payment Gateway die Zahlungsinformation entschlüsseln und prüfen, ob der Kunde liquide ist. Ist das der Fall, erhält der Händler vom Payment Gateway diese Information und kann dem Kunden die Bestellung bestätigen. Die Zahlung erfolgt durch eine Aufforderung des Payment Gateways an die Bank des Kunden, die Zahlung an die Bank des Händlers einzuleiten.

Prinzipiell verläuft jegliche Kommunikation zwischen den beteiligten Parteien verschlüsselt und signiert ab und wird von jedem geprüft.

SET ist inzwischen zu einem Standard für Kreditkartenübertragung im Internet geworden und viele Organisationen benutzen dieses Verfahren. Zum Beispiel wird es von Cybercash verwendet [Cybercash 1998], einem ursprünglich eigenen Verfahren zur Kreditkartenübertragung. In einem Pilotversuch der Dresdner Bank und der Sachsen LB wird Cybercash auf der Basis von SET seit März 1997 eingesetzt [Dresdner Bank 1998].

Im Mai 1998 hat die SGZ-Bank die Markteinführung von SET begonnen [SGZ 1998]. Als Zentralbank und Technologiezentrum für 400 Volks- und Raiffeisenbanken in Baden, Hessen, der Pfalz und dem Saarland bietet sie ihren Partnerbanken eine komplette SET-Lösung an. Die gewählte Lösung basiert auf dem SET-Standard 1.0. Die Zertifizierung erfolgt über den Pfad Bankfiliale, Kreditkartenorganisation, Zertifizierungsstelle und wird von VeriSign in den USA ausgestellt. Das Payment Gateway befindet sich in Dänemark.

Bewertung:

Im Gegensatz zu Debitverfahren, die bei kartengestützten Verfahren verwendet werden, ist es SET nicht möglich, die Identität des Kunden völlig geheimzuhalten. Da die Bank dem Kunden einen Kredit gewährt, ist es notwendig, daß sie seine Identität kennt. Es ist jedoch nicht notwendig, daß der Händler die Identität des Kunden kennt. Er braucht lediglich eine Information, ob der Kunde kreditwürdig ist und wohin er seine Ware liefern soll. Ebensowenig ist es notwendig, daß die Bank erfährt, welche Produkte der Kunde kauft.

Vertraulichkeit/Anonymität: Die Vertraulichkeit der gesendeten Transaktionsdaten ist durch kryptographische Verfahren gesichert. Über die Stärke der verwendeten Verfahren kann jedoch keine Aussage gemacht werden. Falls zur Verschlüsselung ausreichend sichere Kryptoverfahren eingesetzt werden, ist die Vertraulichkeit gewährleistet. Dies ist jedoch ein prinzipielles Problem aller Zahlungssysteme, die kryptologische Verfahren zur Sicherung verwenden.

Die Kundenidentität ist dem Händler und allen anderen beteiligten Parteien zu jedem Zeitpunkt bekannt, da sie aus den Zertifikaten hervorgeht, die vor einem Bezahlvorgang überprüft werden. Somit könnten Profile über die Kauftransaktionen eines Kunden von den jeweiligen Händlern erstellt werden, da der Händler die Information über die gekaufte Ware hat. Die Bank jedoch hat außer der Identität des Kunden und des Gesamtbetrages, keine Information über die gekaufte Ware.

Integrität: SET verwendet digitale Signaturen, die gewährleisten sollen, daß die Zahlungstransaktion weder manipuliert noch der Zahlungsstrom umgeleitet werden kann. Wie auch bei dem Aspekt der Vertraulichkeit, stellt sich hier die Frage, ob die Stärke der verwendeten kryptographischen Verfahren ausreicht, die Integrität zu gewährleisten. Ein anderes Problem ist die rechtliche Anerkennung von digitalen Signaturen als Beweismittel vor Gericht. Das deutsche Signaturgesetz [BMBF 1997] regelt zwar die Handhabung von digitalen Signaturen, erkennt sie jedoch nicht als Beweismittel vor Gericht an. Weiterhin ist die Äquivalenz zu handschriftlichen Unterschriften nicht gegeben.

Verläßlichkeit: Auf dem Übertragungsweg können keine Geldeinheiten verlorengehen oder beschädigt werden, da nur Kreditinformationen übertragen werden. Da die eigentliche Geldübertragung zwischen den Banken stattfindet, kann man davon ausgehen, daß diese Kreditinstitute eine ausreichende Sicherung ihrer Rechner vornehmen, so daß diese Betrachtung an dieser Stelle für die Bewertung des Zahlungsverfahrens nicht von Bedeutung ist.

Zurechenbarkeit: SET ist nicht anonym. Jede Transaktion hinterläßt deutliche Spuren. Dadurch ist ein Schutz vor Fälschungen, Geldwäsche und Verleugnung des Erhalts einer Zahlung gegeben, soweit die kryptographischen Verfahren nicht überwunden werden können. Vor jedem Zahlungsvorgang authentisieren sich die beiden beteiligten Parteien gegenseitig durch Überprüfen der Zertifikate. Durch die Verwendung von digitalen Signaturen wird jede Zahlung autorisiert.

Informationelle Selbstbestimmung: Wie schon erwähnt, ist SET nicht anonym. Das bedeutet, daß der Kunde nicht selbst entscheiden kann, ob er dem Händler seine Identität mitteilt oder nicht. Dabei wäre es rein technisch gesehen nicht notwendig, dem Händler die Kenntnis über die Kundenidentität mitzuteilen. Der Bank gegenüber kann der Kunde auf keinen Fall anonym bleiben, da das Abrechnungsverfahren die Identität des Kunden benötigt. Es bleibt dem Kunden nur die Wahl, SET als personenbezogenes Kreditzahlungsmittel zu nutzen oder ganz darauf zu verzichten.

Einfache Handhabung: Die Benutzung von SET stellt einige Anforderungen an die beteiligten Parteien. Der Zahlungsvorgang wird durch die Überprüfung der Zertifikate und das Bilden von Hashwerten und Signaturen aufwendiger als es bei bisherigen Kreditzahlungen ist. Der Aufwand garantiert jedoch ein gewisses Maß an Sicherheit, das bei klassischer Kreditkartenzahlung über das Internet, also unverschlüsselt und damit ungeschützt, nicht gegeben ist. Sicherheitsmaßnahmen, wie Verschlüsselung und digitale Signaturen, stellen in diesem Fall einen Nachteil für einfache Handhabung dar.

2.3.2.3 Ecash

David Chaum gründete die Firma DigiCash und entwickelte ein Verfahren für elektronisches Geld, das die Anonymität des Kunden bewahrt [Chaum 1992]. Er entwickelte eine Methode, mit der es möglich ist, elektronisches Geld von einer Bank abzuheben und es bei einem Händler auszugeben, ohne daß dem Kunden der Kauf zugeordnet werden kann [DigiCash 1996]. Dabei stand in erster Linie, neben der Fälschungssicherheit der elektronischen Zahlungsmittel, die Privatsphäre des Kunden im Vordergrund.

Beschreibung:

Der Kunde generiert auf seinem heimischen Rechner eine gewisse Anzahl von elektronischen Geldeinheiten mit zugehörigen Seriennummern und schickt sie an die Bank. Die Bank bucht den Gegenwert vom Konto des Kunden ab und bestätigt mit ihrer digitalen Unterschrift die Gültigkeit der eingereichten Geldeinheiten. Prinzipiell besteht die Gefahr, daß zwei Kunden Geldeinheiten mit derselben Seriennummer erzeugen und diese von der Bank signiert werden. Die zuerst eingelöste Geldeinheit ist gültig, während danach eingelöste Einheiten zurückgewiesen werden, obwohl diese ordnungsgemäß erzeugt wurden. Die signierten Geldeinheiten speichert der Kunde auf seinem Rechner und kann sie nun bei einem Händler ausgeben und an Privatpersonen weitergeben.

Die Beteiligten bei Ecash sind:

- Der Kunde
- Der Händler
- Die Bank

Der Händler erhält die Geldeinheiten vom Kunden als Bezahlung seiner Ware, reicht diese bei der Bank ein und läßt anhand der Seriennummer prüfen, ob diese Geldeinheit bereits einmal eingelöst wurde. Ist das nicht der Fall, ist der Zahlungsvorgang erfolgreich abgeschlossen und der Kunde erhält seine Ware. Der Händler bekommt von der Bank den erhaltenen Zahlungsbetrag auf sein Konto gutgeschrieben.

Damit die Anonymität des Kunden gewahrt bleibt, arbeitet das Ecash-Verfahren mit blinden Signaturen [Schneier 1996]. Der Kunde erzeugt seine Geldeinheit mit einem Zufallszahlengenerator und versieht sie mit einer Seriennummer. Diese Geldeinheit

schickt er nun nicht im Klartext an die Bank, sondern steckt sie in einen digitalen „Briefumschlag“. Die Bank sieht den Wert der Geldeinheit und bucht diesen Betrag von dem Konto des Kunden ab. Die Bank kann die Seriennummer nicht sehen, da sie von dem digitalen Umschlag quasi „verdeckt“ ist. Dann erzeugt die Bank ihre digitale Signatur, macht die Geldeinheit dadurch gültig und schickt sie an den Kunden zurück. Dieser entfernt den „Umschlag“ und erhält auf diese Weise eine gültige Geldeinheit, die nicht zu ihm zurückverfolgbar ist. Auf eigenen Wunsch kann der Kunde jedoch selbst seine Anonymität aufheben und den Bezug zu einer Zahlung im Streitfall herstellen.

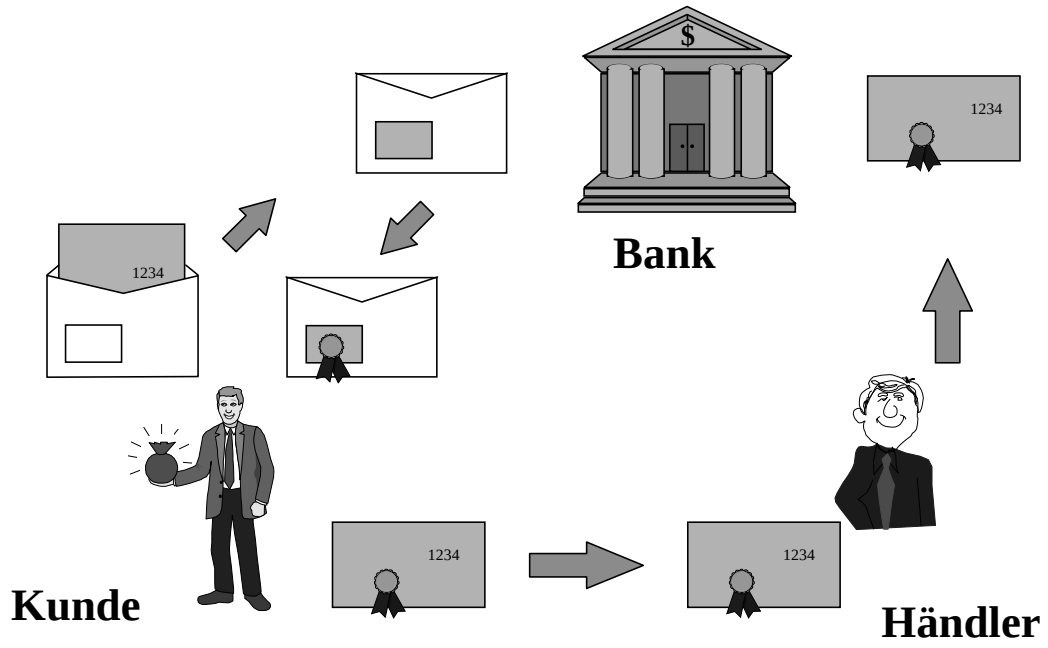


Abbildung 2-9: Ecash

Quelle: [Engel, Lessig 1997], S. 120

Nachteil des Ecash-Verfahrens ist, daß zwischen Währungen verschiedener Banken nicht elektronisch konvertiert werden kann. Jede Bank gibt damit ihr eigenes Ecash heraus und die Kunden können es nur bei dieser Bank wieder einlösen.

Das Ecash-Verfahren wurde seit 1996 in verschiedenen Pilotversuchen in Amerika (Mark Twain Bank), Finnland (EUNet), Deutschland (Deutsche Bank) und Australien (Advance Bank) getestet. Der größte und erfolgversprechendste Versuch war in den USA. Die Mark Twain Bank führte Ecash auf Dollar-Basis ein und stellte eine umfangreiche Infrastruktur zur Verfügung. Leider wurde dieser Versuch im September 1998 aufgrund der geringen Umsätze beendet. Die Firma DigiCash hat sich für zahlungsunfähig erklärt, und bereits im Februar 1998 hat das US-Unternehmen 44 seiner 50 Mitarbeiter entlassen. Die niederländische Niederlassung wurde im September desselben Jahres geschlossen [CZ46 1998].

In der Schweiz hingegen hat die Swiss NetPay AG einen bis Ende 1998 angekündigten Pilotversuch mit Ecash durchgeführt [Swiss NetPay 1999]. Es nehmen etwa 3000 Kunden und circa 30 Händler daran teil. Ein Unterschied zu allen anderen

Pilotversuchen ist die Tatsache, daß Ecash nicht an eine Bank gebunden ist, sondern von allen schweizerischen Banken ausgegeben und eingelöst werden kann. Im Januar 1999 läuft der Pilotversuch noch immer und soll um eine unbestimmte Zeit verlängert werden. Einen Ergebnisbericht gibt es zu diesem Zeitpunkt noch nicht.

Seit April 1997 hat die Bank Austria Ecash in einem Pilotversuch lizensiert und getestet [Bank Austria 1998]. Seit Mai 1998 ist Ecash auch für die Kunden der Bank Austria zugänglich und wird in großem Rahmen genutzt. Die Laufzeit des Projekts ist nicht bestimmt. Die Kunden können mit Ecash bei verschiedenen Händlern bezahlen und zum Beispiel Tickets für Veranstaltungen kaufen. Vom heimischen Computer aus können in Zukunft Vorschauen auf neue Kinofilme angesehen und bezahlt werden. Eine der größten Tageszeitungen bietet den Verkauf über das Internet von einzelnen Artikeln und Sparten an. Insgesamt soll das Angebot in der nächsten Zeit stark erweitert werden. [Digicash 1998].

Bewertung:

Eine Bewertung dieses Verfahren erfolgt unter ähnlichen Gesichtspunkten wie bei der GeldKarte und Mondex, da es sich bei Ecash ebenso um ein Debitverfahren zur Repräsentation von elektronischem Geld handelt.

Vertraulichkeit/Anonymität: Bei Ecash handelt der Kunde anonym. Dadurch wird die Vertraulichkeit der Kundenidentität gewährleistet. Die Übertragung der Zahlungsinformation erfolgt in der Regel unverschlüsselt. Dies stellt jedoch kein Vertraulichkeitsproblem dar, da kein Bezug zwischen der Transaktion und dem Kunden hergestellt werden kann. Die Erstellung von Kundenprofilen ist daher unmöglich. Weder der Händler noch die Bank kann einen Zusammenhang zwischen Zahlungsvorgängen und Kunden herstellen.

Integrität: Das Erzeugen von Falschgeld wird durch kryptographische Verfahren und Seriennummern verhindert, solange die Sicherheit der verwendeten Schlüssel gewährleistet ist. Da es sich um ein Online-System handelt, kann der Händler sicher sein, daß er sein Geld erhält, bevor er seine Ware liefert. Ecash verwendet signierte Geldeinheiten und macht dadurch eine Manipulation der Geldeinheiten unmöglich.

Verläßlichkeit: Aufgrund eines Netzwerkfehlers könnte die Situation entstehen, daß nicht klar ist, wo sich eine Geldeinheit im Augenblick befindet. Glauben beide Parteien im Besitz einer Geldeinheit zu sein, kommt es darauf an, wer sie zuerst einlöst. Der andere hat dann das Nachsehen. Der Kunde kann jedoch den Bezug zu einer Geldeinheit herstellen, wenn Unklarheiten bestehen.

Ein weiteres Problem besteht in dem möglichen Erzeugen von zwei Ecash-Geldeinheiten mit derselben Seriennummer. Die zweite, obwohl gültige Einheit wird von der Bank zurückgewiesen.

Zurechenbarkeit: Prinzipiell ist der Kunde völlig anonym, so daß er auch bei Fehlverhalten nicht zur Rechenschaft gezogen werden kann. Die Software des Kunden führt Buch über die getätigten Transaktionen. Der Kunde kann im

Streitfall dem Händler den Erhalt des Geldes nachweisen, in dem er die Faktoren für die blinde Signatur preisgibt. Da der Kunde anonym ist, kann keine gegenseitige Authentisierung stattfinden.

Informationelle Selbstbestimmung: Bei Ecash wird das informationelle Selbstbestimmungsrecht bestens gewährleistet. Da der Kunde vollständig anonym handeln kann, kann er selbst entscheiden, wann er wem seine Identität mitteilt. Er kann ebenso selbst entscheiden, ob er in einem Streitfall seine Identität preisgibt, um beispielsweise eine Zahlung nachzuweisen.

Einfache Handhabung: Ecash wurde für eine einfache Handhabung konzipiert. Dem Benutzer wird eine einfache „point and click" Benutzungsoberfläche zur Verfügung gestellt, mit der er alle Transaktionen leicht durchführen kann [DigiCash 1996].

2.4 Bewertung anhand der grundlegenden Sicherheitsanforderungen

Alle hier vorgestellten Verfahren für elektronische Zahlungssysteme wurden bereits bei ihrer Vorstellung anhand der aufgelisteten Sicherheitsanforderungen bewertet. An dieser Stelle soll noch einmal zusammenfassend ein Überblick über die einzelnen Bewertungen gegeben werden (Tabelle 2-1). Ein Haken als Eintrag in der Tabelle bedeutet, daß die entsprechende Sicherheitsanforderung in der Spalte des jeweiligen Zahlungssystems erfüllt ist.

	GeldKarte	Mondex	Millicent	SET	Ecash
Vertraulichkeit	✓	(✓)	✓	✓	
Anonymität					✓
Integrität	✓	✓	✓	✓	✓
Verläßlichkeit	✓	✓		✓	
Zurechenbarkeit	✓	✓	✓	✓	✓
Inf. Selbstbestimmung					✓
Einfache Handhabung		✓			✓

Tabelle 2-1: Bewertung der beschriebenen Zahlungssysteme

Die einzelnen Zahlungssysteme sind jedoch nicht ohne weiteres miteinander vergleichbar. Nicht nur, daß sie unterschiedliche Zahlungsmöglichkeiten, wie Zahlung per Kreditkarte, per Koupon oder per Bargeld, zur Verfügung stellen, sondern sie erlauben das Transferieren von ganz unterschiedlich hohen Beträgen. Wie bereits in diesem Kapitel herausgearbeitet wurde, führen diese nicht unwesentlichen Unterschiede zu unterschiedlichen Bewertungen der einzelnen Sicherheitsanforderungen.

Aus diesem Grund ist es problematisch, die unterschiedlichen Zahlungssysteme in einer Tabelle gegenüberzustellen. Trotzdem soll der Versuch an dieser Stelle gewagt werden.

Vertraulichkeit der vorgestellten Verfahren wird sehr unterschiedlich unterstützt. Prinzipiell stellt sich bei allen Verfahren die Frage nach der Stärke der kryptographischen Algorithmen.

Bei der GeldKarte und bei Mondex erfolgt die Übertragung der Transaktionsdaten zu den Verrechnungsstellen prinzipiell verschlüsselt. Der Transaktionsdatensatz (die letzten zehn beziehungsweise 15 Transaktionen) ist zusätzlich auf der Chipkarte gespeichert. Diese Information kann mit jedem üblichen Taschenlesegerät ausgelesen werden. Da bei Mondex neben Kaufdatum und Betrag auch die Information über den Händler vermerkt ist, können davon leicht Kaufprofile des Kunden erstellt werden. Aus diesem Grund wird der Aspekt der Vertraulichkeit bei Mondex eingeschränkt unterstützt (Tabelleneintrag in Klammern), während er bei der GeldKarte stärker unterstützt wird.

Millicent stellt keinen hohen Anspruch an Übertragungssicherheit, da nur niedrige Beträge transferiert werden. Jedoch ist die Gefahr der Profilbildung bei vielen Käufen mit geringen Beträgen sehr hoch. Millicent stellt daher in seinen unterschiedlichen Protokollvarianten die Möglichkeit zur Verfügung, verschlüsselt zu übertragen.

SET sichert die Vertraulichkeit der Transaktionsdaten durch kryptographische Algorithmen.

Ecash benutzt keine Verschlüsselung zur Wahrung der Vertraulichkeit. Da dieses Verfahren jedoch als einziges anonym (für den Kunden) ist, besteht keine Möglichkeit, Kaufprofile des Kunden zu erstellen, da kein Bezug zwischen Transaktion und Benutzer herstellbar ist.

Anonymität wird nur von Ecash gewährt. Bei SET ist vom Prinzip her keine anonyme Benutzung möglich, da es sich um ein Verfahren zur Übertragung von Kreditkartendaten handelt. Alle anderen Verfahren (GeldKarte, Mondex und Millicent) hätten jedoch auch als anoyme Zahlungsverfahren konzipiert werden können. Aus verschiedenen Gründen wurde dies jedoch nicht getan. Damit ist Ecash das einzige anonyme Zahlungsverfahren, wobei jedoch die Anonymität nur im Nichtbestreitungsfalle gegeben ist. Sobald der Kunde eine Zahlung bestreiten oder nachweisen will, hebt er seine Anonymität auf.

Integrität wird bei allen vorgestellten Verfahren unterstützt. Alle Verfahren nutzen entweder kryptographische Prüfsummen oder digitale Signaturen zur Sicherstellung der Unverfälschtheit und Authentizität der Nachrichten.

Verläßlichkeit ist bei der GeldKarte und bei Mondex am ehesten gegeben. Beide Systeme versprechen die Erstattung verlorengegangenen Geldes, soweit nicht Fahrlässigkeit des Kunden zugrunde liegt.

Bei Millicent gibt es keine Aussagen zu diesem Punkt. Es ist davon auszugehen, daß bei einer Beschädigung der Endgeräte oder auf dem Übertragungsweg die betroffenen Geldeinheiten verloren sind.

SET hat keine eingebaute Maßnahme, um Verläßlichkeit sicherzustellen. Da es sich aber um ein Kreditsystem handelt, bei dem die eigentlichen Geldbeträge über das eigene Netz der Banken (SWIFT) transferiert werden, kann man davon ausgehen, daß dies ausreichend von Seiten der Banken abgesichert wurde. Aus diesem Grund fällt dieser Aspekt nicht in diese Betrachtung und wird als erfüllt angesehen.

Ecash kann nicht verhindern, daß bei einer Netzwerkstörung inkonsistente Zustände auftreten und digitale Geldeinheiten verloren gehen. Obwohl Geldeinheiten, die sich auf dem privaten Rechner des Kunden befinden und noch nicht ausgegeben wurden, nach einem Rechnerausfall rekonstruiert werden können, wird dieser Aspekt in der Gesamtwertung als nicht ausreichend und damit als nicht erfüllt angesehen.

Zurechenbarkeit ist bei der GeldKarte und bei Mondex durch die Aufzeichnungen der Transaktionen gegeben. Dadurch können Zahlungen nachgewiesen werden.

Bei Millicent kann lediglich in den Varianten, die signierte Geldeinheiten verwenden, eine Zahlung nachgewiesen werden. In der anderen Variante ist eine Zurechenbarkeit von Zahlungen unmöglich. Insgesamt wird dieser Aspekt jedoch als erfüllt angesehen.

Bei SET hinterläßt jede Transaktion deutliche Spuren. So kann jederzeit eine Zahlung einem Kunden zugeordnet werden.

Ecash verhindert durch sein anonymes Konzept gerade die Zurechenbarkeit von Transaktionen. Im Streitfall kann der Kunde auf eigene Veranlassung die Zuordnung zu einer erfolgten Transaktion herstellen, um eine Zahlung nachzuweisen. Damit ist auch hier die Zurechenbarkeit möglich.

Informationelle Selbstbestimmung ist nur bei dem Verfahren Ecash gegeben, da hier der Kunde absolut anonym handeln kann und nur im Streitfall und auf eigenen Wunsch seine Identität preisgibt. Dadurch ist die Entscheidung hundertprozentig auf seiner Seite, wem er seine personenbezogenen Daten übermittelt.

Einfache Handhabung ist am schwersten zu beurteilen, da dieses Kriterium eine praktische Benutzung aller vorgestellten Verfahren zugrunde legt. Da dies nicht möglich war, bezieht sich die Bewertung nur auf Aussagen, die in der Literatur über die Handhabung der vorgestellten Verfahren gemacht wurden. Daraus geht hervor, daß nur Mondex und Ecash eine einfache Benutzung ermöglichen, während die anderen Systeme recht kompliziert sind.

2.5 Multifunktionale Chipkarten als Vision

Die einzelnen Zahlungssysteme, die im obigen Teil erläutert wurden, erfüllen die genannten Sicherheitsanforderungen auf unterschiedliche Weise. Jedes Zahlungsverfahren für sich stellt unterschiedliche Anforderungen an Sicherheit, Anonymität und einfache Handhabung. Verfahren zum Bezahlen geringer Beträge haben beispielsweise eine geringere Anforderung an Übertragungssicherheit (Integrität und Zurechenbarkeit), dafür aber eine besonders hohe Anforderung an Vertraulichkeit und Anonymität. Dagegen ist beim Bezahlen hoher Beträge die Anforderung an Anonymität zugunsten der Integrität und Zurechenbarkeit kaum ausgeprägt.

Zur Zeit gibt es viele verschiedene Zahlungsverfahren, sowohl auf Chipkartenbasis, als auch auf Netzbasis. Im Augenblick ist noch nicht absehbar, welche Verfahren sich in Zukunft durchsetzen und welche Verfahren die größte Akzeptanz der Benutzer genießen werden.

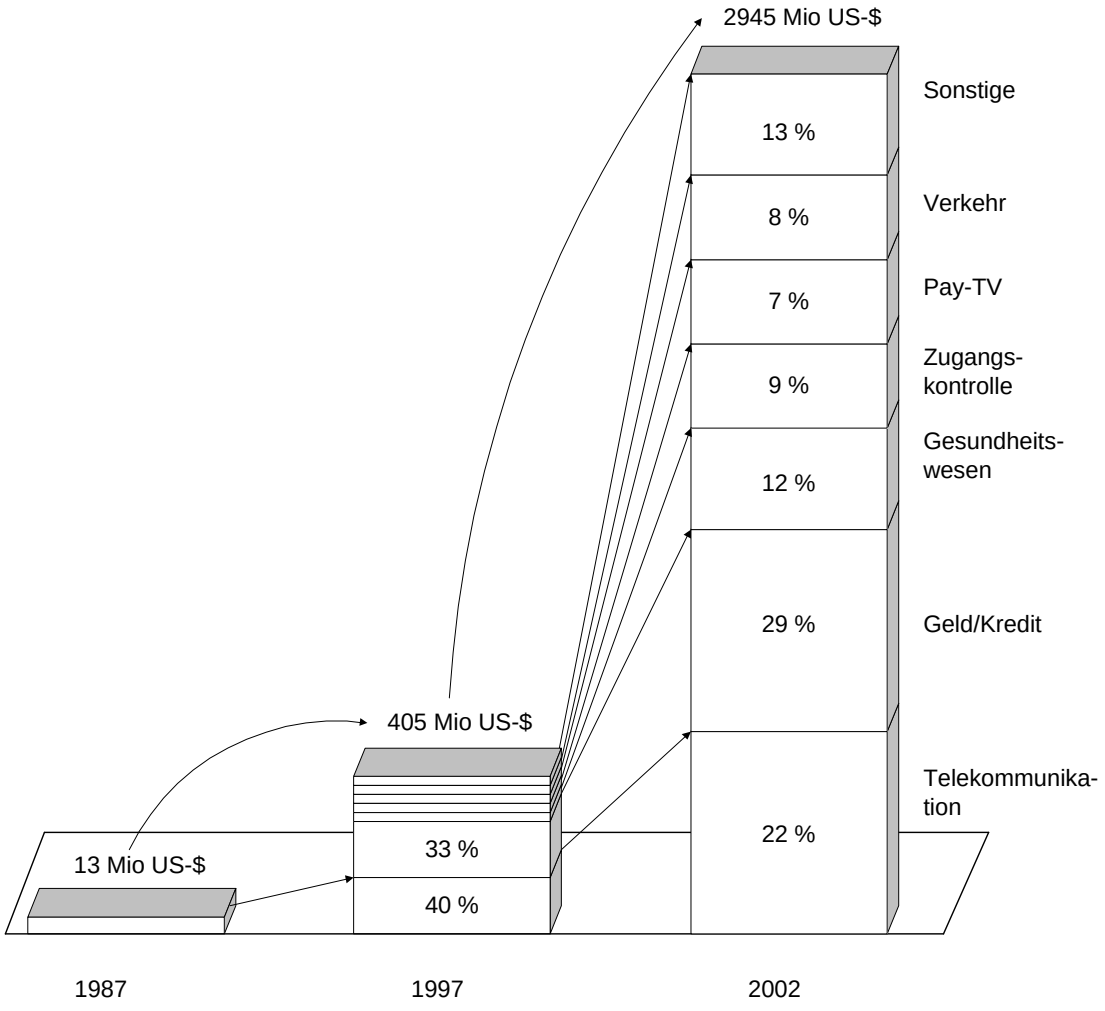


Abbildung 2-10: Schätzung der Entwicklung des Weltmarktes für Chipkarten
Quelle: [Eberl 1998], S. 10

Die Bedeutung der Chipkarte für elektronische Zahlungssysteme ist schon heute unübersehbar. Für das Jahr 2002 wird ein Boom für die Chipkartenindustrie vorhergesagt, der besonders den Bereich des elektronischen Zahlungsverkehrs betreffen soll [Eberl 1998].

Nach einer Schätzung der Siemens AG wird der Geld- / Kreditbereich im Jahr 2002 knapp 30% des gesamten Weltmarktes für Chipkarten einnehmen (Abbildung 2-10).

Nimmt man diese Prognose ernst, kann man davon ausgehen, daß Chipkarten auch weiterhin eine wichtige Rolle für den elektronischen Zahlungsverkehr spielen werden. Elektronische Zahlungssysteme werden entweder auf Chipkarten direkt basieren oder Chipkarten als Zugangsmedium zu netzgestützten Verfahren verwenden. Dabei soll im Augenblick nicht in der Verwendung zwischen Chipkarten und Magnetstreifenkarten unterschieden werden. In naher Zukunft werden jedoch vermutlich Magnetstreifenkarten durch die leistungsfähigeren Chipkarten ersetzt werden, so daß in dieser Arbeit im wesentlichen auf die Chipkarten- und weniger auf die Magnetstreifentechnologie eingegangen wird (siehe Kapitel 3).

Prinzipiell gibt es zwei mögliche Entwicklungsvisionen:

- Viele Chipkarten mit einzelnen differenzierten und spezialisierten Zahlungsverfahren
- Eine multifunktionale Chipkarte mit unterschiedlichen Zahlungsverfahren

Die erste Vision ist eine mögliche Fortführung der heutigen Situation. Dieser Trend zwingt sich auf, wenn man die Literatur zu elektronischen Zahlungssystemen studiert, in der die Listen solcher Systeme immer länger werden. Zur Zeit existieren viele unterschiedliche inhomogene Zahlungssysteme, die jeweils auf einer eigenen Karte basieren, wie zum Beispiel die unterschiedlichen Kreditkarten oder Chipkarten zur elektronischen Geldbörse. Im weiteren Sinne des Electronic Commerce gibt es Bankkarten, Telefonkarten, Kundenkarten und viele andere mehr. Über den Electronic Commerce hinaus gibt es zusätzlich die Krankenversichertenkarte. Denkbar sind noch viele weitere Einzelkarten, die nicht zueinander kompatibel sind. Nachteil dieser Differenzierung von Zahlungssystemen ist die Tatsache, daß jedes System eine gewisse Marktdurchdringung erreichen muß, um wirtschaftlich überleben zu können. Kunden und Händler werden es als unzumutbare Last empfinden, mit zu vielen Verfahren umgehen zu müssen. Aufwand und Kosten der Übertragung des Geldes von einem System in ein anderes können unverhältnismäßig hoch werden.

Aus diesen Überlegungen heraus ist die zweite genannte Vision für den elektronischen Zahlungsverkehr sehr wahrscheinlich. Vermutlich werden unterschiedliche Zahlungsfunktionen auf einer Chipkarte vereint werden. Die Vorteile einer Homogenisierung der verschiedenen Zahlungssysteme auf einer multifunktionalen Chipkarte liegen auf der Hand: Die Benutzer müssen nur ein System verstehen und bedienen können. Die Systemkosten je Transaktion werden deutlich günstiger sein und Kompatibilitätsprobleme treten nicht auf. Erste Bestätigungen dieser Vision ist die deutsche GeldKarte, die als Erweiterung der EC- oder Kundenkarte realisiert wurde. In der

Slowakei wurde bereits eine multifunktionale Chipkarte auf den Markt gebracht, die verschiedene Zahlungssysteme wie elektronische Geldbörse, Kreditkarte und Bankkarte vereint [Thompson 1998].

Die zweite Entwicklungsvision, eine multifunktionale Chipkarte im elektronischen Zahlungsverkehr wird in dieser Arbeit als wahrscheinlich angenommen und als Basis den weiteren Betrachtungen zugrunde gelegt. Im nachfolgenden Kapitel wird daher auf die grundlegende Technologie der Chipkarten eingegangen.

Welche Bedrohungen und daraus resultierende Anforderungen ergeben sich nun für multifunktionale Chipkarten? Als Basis für diese Überlegungen wird eine multifunktionale Chipkarte betrachtet, die folgende Anwendungen kombiniert:

- Elektronische Geldbörse
- Kreditkarte
- Bankkarte für Überweisungen u.ä.

Jede dieser drei Anwendungen hat spezielle Anforderungen bezüglich Sicherheit, Anonymität und einfacher Handhabung, wie in den Bewertungen der einzelnen Zahlungssysteme gezeigt wurde (siehe Kapitel 2.3.1 und 2.3.2).

Zusammengefaßt läßt sich für multifunktionale Chipkarten folgendes festhalten:

- Unterschiedliche Anwendungen haben unterschiedliche Sicherheitsanforderungen
- Der Benutzer soll selbst das gewünschte Sicherheitsniveau der Anwendung bestimmen können
- Es muß die Möglichkeit der anonymen Benutzung von Anwendungen geben (eventuell mit eingeschränkter Funktionalität)
- Eine freie Gestaltung der Chipkarte bezüglich der vorhandenen Anwendungen muß möglich sein
- Gleichzeitige Ausführung von konfliktfreien Anwendungen muß möglich sein
- Akzeptanz und Vertrauen in das neue System kann durch vollständige Kontrolle des Benutzers erreicht werden
- Eine einfache Handhabung der unterschiedlichen Anwendungen muß ermöglicht werden

Damit verschiedene Anwendungen auf einer Chipkarte konfliktfrei nebeneinander existieren können, muß garantiert werden, daß sich diese Anwendungen nicht gegenseitig beeinflussen. Daten, die auf der Chipkarte gespeichert sind, dürfen nicht von unautorisierten Anwendungen benutzt werden. Der Benutzer muß selbst entscheiden können, wem er welche Daten über sich zur Verfügung stellt.

Dies kann ein Konzept erreichen, das dem Benutzer ermöglicht, in verschiedenen Sicherheitsniveaus zu handeln. Für jede Anwendung, also zum Beispiel für jedes Zahlungsverfahren, kann ein Sicherheitsniveau definiert werden. Diese Sicherheitsniveaus können als Rollen definiert werden, in denen der Benutzer handeln kann. Weiterhin kann es in diesem Konzept verschiedene Aufgaben (zum Beispiel unterschiedliche Zahlungsvorgänge) geben, die ein Benutzer erledigen kann. Durch die Wahl einer Aufgabe und einer entsprechenden Rolle, kann der Benutzer bestimmte Zahlungsvorgänge auf unterschiedlichen Sicherheitsniveaus ausführen.

Um ein solches Konzept sinnvoll auf der Basis von Chipkarten einzusetzen, ist es notwendig, dieses Konzept als Sicherheitsmodell zu beschreiben. Sicherheitsmodelle haben die Eigenschaft, daß sie formale und informelle Bestandteile haben und exakt spezifizierbar sind.

Es existiert bereits eine große Anzahl von Sicherheitsmodellen in der Literatur und der Praxis. In einem nachfolgenden Kapitel wird eine Auswahl an bestehenden Sicherheitsmodellen vorgestellt und bezüglich der Benutzung in multifunktionalen Chipkarten bewertet (siehe Kapitel 5). Das Ergebnis dieser Bewertung führt zu einem neuen, spezifischen Sicherheitsmodell, das speziell auf multifunktionale Chipkartenanwendungen zugeschnitten ist. Das neue Sicherheitsmodell (R&A-Modell) basiert auf Rollen und Aufgaben. Rollen, in denen ein Benutzer handeln kann, und Aufgaben, die er erledigen kann (siehe Kapitel 6). Das R&A-Modell soll dazu beitragen, multifunktionale Chipkartenanwendungen im elektronischen Zahlungsverkehr sicher und benutzungsfreundlich zu konzipieren und damit eine breite Akzeptanz zu ermöglichen.

3 Chipkarten

In einem weiteren grundlegenden Kapitel dieser Arbeit wird die Technologie der Chipkarten und die dort verwendete Kryptographie beschrieben.

Chipkarten finden immer weitere Verbreitung im täglichen Leben. Besonders im elektronischen Zahlungsverkehr werden immer mehr Chipkarten als Zahlungsmedium oder Zugangsmedium für Bankdienste eingesetzt. Wie im vorhergehenden Kapitel deutlich gemacht wurde, ist es vorstellbar, mehrere unterschiedliche Zahlungsverfahren auf einer Chipkarte zu integrieren. Da diese Zahlungsverfahren vielschichtige Sicherheitsanforderungen haben können, ist es notwendig, die zugrundeliegende Chipkarte so sicher wie möglich zu gestalten. Dazu gehören neben den technischen Voraussetzungen der Chipkarte ebenso organisatorische und sicherheitsmodelltechnische Überlegungen.

Die technischen Voraussetzungen der Chipkarte sind weitgehend vorhanden und müssen nach entsprechenden Anforderungen eingesetzt werden. Es besteht jedoch ein Bedarf an Sicherheitsmodellen, die mit Chipkarten kombiniert werden können. Bereits in Kapitel 2.5 wurde ein Sicherheitsmodell angedeutet, das durch Rollen und Aufgaben eine sichere Benutzung von unterschiedlichen Zahlungsfunktionen auf einer Chipkarte unterstützen soll. Das in Kapitel 6 beschriebene R&A-Modell wird für multifunktionale Chipkarten im elektronischen Zahlungsverkehr benutzt. Deshalb werden in diesem Kapitel die Grundlagen für ein technisches Verständnis für Chipkarten gelegt.

Nach einem kurzen historischen Überblick (siehe Kapitel 3.1) werden die verschiedenen Kartenarten (siehe Kapitel 3.2), wie Speicher- und Mikroprozessorkarten, vorgestellt. Anschließend werden physikalische und elektrische Eigenschaften erläutert (siehe Kapitel 3.3). Nach dem Beschreiben des Lebenszyklus einer Chipkarte (siehe Kapitel 3.4), liefert das nachfolgende Unterkapitel einen Überblick über die Funktionsweise von Chipkarten-Betriebssystemen (siehe Kapitel 3.5). Im nachfolgenden Kapitel werden die Grundlagen der Kryptographie erläutert (siehe Kapitel 4), da die Prinzipien der Authentisierung, Verschlüsselung und digitale Signaturen für Chipkarten von großer Bedeutung sind.

3.1 Historie und Entwicklung

Die Geschichte der Plastikkarten begann Anfang der 50er Jahre in den USA. Die bis dahin gebräuchlichen Karten aus Papier oder Karton, die mechanischen Belastungen und Klimaeinwirkungen nur unzureichend standhielten, wurden von langlebigen Karten abgelöst, die aus dem preisgünstigen Kunststoff PVC hergestellt wurden. Eine erste solche Karte wurde 1950 von Diners Club herausgegeben und diente als Statussymbol, weil sie nur einem exklusiven Personenkreis zugänglich war. Die Akzeptanz dieser Karte, die vorerst nur in Hotels und Restaurants genutzt werden konnte, erhöhte sich durch den Eintritt von VISA und Mastercard in die Kartenszene, zunächst in Amerika, dann auch Europa und dem Rest der Welt [Rankl, Effing 1996].

Heutzutage werden viele hundert Millionen Karten produziert und ausgegeben, um weltweit ohne Bargeld einzukaufen. Der Karteninhaber ist jederzeit - Bonität vorausgesetzt - zahlungsfähig, ohne das Risiko des Geldverlustes durch Diebstahl tragen oder im Ausland Geld tauschen zu müssen.

Zu Beginn war die Funktion der Karten recht einfach. Die Karten dienten zunächst als geschützte Datenträger, wobei die allgemeinen Informationen, wie der Name des Kartenausgebers, auf der Karte aufgedruckt waren, und individuelle Daten, wie der Name des Karteninhabers oder die Kartennummer, durch Hochprägung aufgebracht wurden. Zusätzlich hatten viele Karten ein Unterschriftsfeld, auf dem der Karteninhaber seine Referenzunterschrift leisten konnte. Die Sicherheit dieses Systems hing in erster Linie von der Sorgfalt des Personals in den Kartenakzeptanzstellen ab, was mit der zunehmenden Verbreitung zum Problem wurde.

Zunehmender Betrug und steigender Kostendruck führten zu einer ersten Verbesserung der Karte durch Aufbringen eines Magnetstreifens auf der Kartenrückseite. Dort waren zusätzlich zum Aufdruck und der Hochprägung digitalisierte Daten in maschinenlesbarer Form gespeichert. Dieser Kartentyp hat auch heute noch als Kreditkarte oder als EC-Karte die größte Verbreitung im Zahlungsverkehr. Die Magnetstreifentechnik hat jedoch einen entscheidenden Nachteil. Die Daten auf dem Magnetstreifen können von jedem, der ein Lese-/Schreibgerät für Magentstreifenkarten besitzt, gelesen, gelöscht und geschrieben werden. Diese Technik eignet sich also nicht zur Speicherung sensitiver Daten, wird jedoch bis heute im Zahlungsverkehr eingesetzt. Zur Sicherstellung der Vertraulichkeit und Integrität sind alternative oder zusätzliche technische Maßnahmen erforderlich.

Durch rasante Fortschritte der Mikroelektronik geht die Entwicklung der Chipkarte voran. Bereits in den 70er Jahren wurde es möglich, Datenspeicher und Rechnerlogik auf einem einzigen kleinen Siliziumplättchen zu integrieren. In Deutschland wurde 1968 das erste Patent einer Identifikationskarte mit integriertem Schaltkreis von Jürgen Dethloff und Helmut Grötrupp angemeldet ([Dethloff, Grötrupp 1968] und [Dethloff 1968]). 1970 folgte eine ähnliche Anmeldung in Japan und 1974 gelang der Durchbruch durch die Anmeldung der Chipkartenpatente in Frankreich von Roland Moreno. Die französische Telefongesellschaft PTT konnte 1984 einen Feldversuch mit Telefonkarten erfolgreich durchführen, der ein Jahr später auch in Deutschland folgte. Dort waren verschiedene Kartentechnologien im Vergleich und die Chipkarte versprach mit ihrer großen Flexibilität die Zukunftsorientierteste zu sein. Im Jahr 1986 gab es in Frankreich bereits mehrere Millionen Chipkarten zum Telefonieren, 1990 waren es fast 60 Millionen und 1995 über 200 Millionen. Deutschland erlebte den Boom mit einer Verzögerung von etwa drei Jahren.

Im Bankenbereich verläuft die Entwicklung etwas langsamer, da die dort verwendeten Karten eine höhere Komplexität als Telefonkarten aufweisen und dort auf bereits vorhandene Systeme Rücksicht genommen werden muß, was eine Innovation und Einführung neuer Technologien stark behindert.

Die Entwicklung der modernen Kryptographie bedeutete einen erheblichen Fortschritt. Die Implementierung von komplexen und mathematisch anspruchsvollen Algorithmen

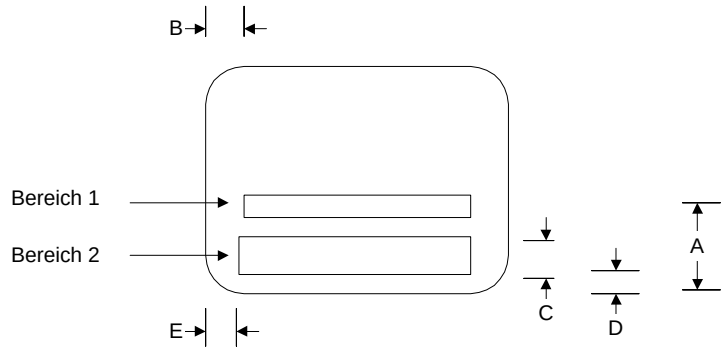
machte den Einsatz von Kryptographie als Sicherheitswerkzeug für nahezu jeden verfügbar. Die Chipkarte ist das ideale Medium, das hohe Sicherheit auf der Basis der Kryptographie für jedermann zugänglich macht, weil sie geheime Schlüssel auf relativ sichere Weise speichern und gleichzeitig Kryptoalgorithmen ausführen kann.

3.2 Identifikationskarten

In der ISO-Norm 7810 „Identification Cards - Physical Characteristics“ [ISO 7810] sind Identifikationskarten im ID-1 Format definiert. Dieses Format entspricht - neben dem ID-2 und ID-3 Format - dem gängigen Format von heute genutzten Karten, wie sie zum Beispiel im Zahlungsverkehr eingesetzt werden. Die erwähnte ISO-Norm spezifiziert die physikalischen Eigenschaften einschließlich der Materialeigenschaften wie Flexibilität, Temperaturbeständigkeit und Abmessungen für die verschiedenen Größen. Die ISO-Norm 7816 bildet die Basis für die ID-1 Karte [ISO 7816]. Die wichtigsten Normen, die für Identifikationskarten Anwendung finden, sind im Anhang Normen aufgelistet.

3.2.1 Hochgeprägte Karten

Karten mit Hochprägung stellen die älteste Möglichkeit von Identifikationskarten in maschinenlesbarer Form dar. Die Art und Lage der hochgeprägten Zeichen sind in der ISO-Norm 7811 spezifiziert [ISO 7811]. Im Teil Drei dieser Norm wird die genaue Lage der Zeichen auf der Karte festgelegt, wobei zwei Bereiche unterschieden werden. Der eine Bereich ist für die Kartenidentifikationsnummer reserviert und der andere Bereich für persönliche Daten des Karteninhabers, wie zum Beispiel Name und Adresse.



Bereich 1 für die Identifikationsnummer (19 Zeichen)
Bereich 2 für Name und Adresse (4 * 27 Zeichen)
A: 21,42 mm +/- 0,12 mm
B: 10,18 mm +/- 0,25 mm
C: 14,53 mm
D: 2,41 mm .. 3,30 mm
E: 7,65 mm +/- 0,25 mm

Abbildung 3-1: Lage der Hochprägung auf der ID-1 Karte

Quelle: [Rankl, Effing 1996] S. 29

3.2.2 Magnetstreifenkarten

Da ein wesentlicher Nachteil von hochgeprägten Karten in der Masse von Papierbelegen besteht, deren Handhabung und Auswertung hohe Kosten verursacht, ist ein Magnetstreifen auf der Karte als erste Verbesserung anzusehen. Ein Magnetstreifen auf der Rückseite der Karte enthält die digitale Kodierung der Kartendaten. Zum Lesen wird die Karte von Hand oder maschinell an einem Lesekopf vorbeigezogen, der die Daten ausliest und speichert. Eine Weiterverarbeitung der Daten erfordert weniger Papier.

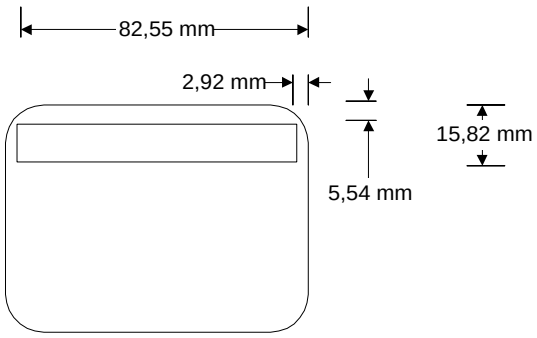


Abbildung 3-2: Lage des Magnetstreifens auf der ID-1 Karte
Quelle [Rankl, Effing 1996], S. 29

Die Teile Zwei, Vier und Fünf der ISO-Norm 7811 beschreiben die Eigenschaften des Magnetstreifens, die Kodiertechnik und die Lage der Magnetspuren. Insgesamt können sich drei Spuren auf dem Magnetstreifen befinden, wobei die ersten beiden Spuren nur für den Lesebetrieb spezifiziert sind und die dritte Spur auch beschrieben werden kann. Die Speicherkapazität ist jedoch mit ca. einem kBit sehr gering und zusätzlich lassen sich die Daten auf dem Magnetstreifen sehr leicht manipulieren, so daß eine Veränderung der Magnetstreifendaten oder Kopieren der Daten auf eine Blankokarte mit einem handelsüblichen Schreib-/Lesegerät problemlos möglich und im Nachhinein nur schwer nachweisbar ist. Die Hersteller von Magnetstreifenkarten haben zwar verschiedene Mechanismen zur Echtheitsprüfung der Karte entwickelt, wie zum Beispiel Hologramme. Dies erfordert jedoch einen speziellen Sensor im Kartenterminal, der sich aufgrund der hohen Kosten und Störungsanfälligkeit noch nicht international durchgesetzt hat. Weiterhin gibt es mehrere Verfahren zur Echtheitsprüfung, die jeweils einen speziellen Sensor benötigen, da sich ein Standard noch nicht durchgesetzt hat.

3.2.3 Chipkarten

Die Chipkarte ist die jüngste Entwicklung der Identifikationskarten im ID-1 Format. Sie zeichnet sich durch eine im Kartenkörper integrierte Schaltung aus, die Elemente zur Datenübertragung, zum Speichern und zum Verarbeiten von Daten enthält. Die Übertragung von Daten erfolgt hierbei entweder über Kontakte (siehe Kapitel 3.3.3) oder kontaktlos (siehe Kapitel 3.3.2).

Die Speicherkapazität von Chipkarten ist um ein Vielfaches größer als die von Magnetstreifen. Während heute Chipkarten mit einer Speicherkapazität von ca. 20 kByte keine Seltenheit mehr sind, wird die Speicherkapazität der nächsten

Chipgenerationen durch die immer weiter fortschreitende Technologie erheblich erhöht werden.

Ein wesentlicher Vorteil gegenüber Magnetstreifenkarten ist die Tatsache, daß gespeicherte Daten vor unerwünschtem Zugriff und Manipulation weitestgehend geschützt sind. Bekannt sind zwar Angriffe auf die Daten und Kommunikationsleitungen des Chips, wie sie von Anderson und Kuhn vorgestellt wurden [Anderson, Kuhn 1997], dafür ist jedoch ein relativ hoher Aufwand zu betreiben, der nur in seltenen Fällen eingesetzt werden wird. Im allgemeinen erfolgt der Zugriff auf die Daten über eine serielle Schnittstelle, die vom Betriebssystem und/oder einer Sicherheitslogik gesteuert wird. So ist es möglich, geheime Daten auf die Chipkarte zu laden, die von außen nicht ohne weiteres gelesen, sondern nur intern vom Rechenwerk des Chips verarbeitet werden können. Die Datenübertragung erfolgt für Speicherkarten synchron, für Mikroprozessorkarten asynchron. Die zusätzliche Fähigkeit, Kryptoalgorithmen zu verwenden, läßt die Chipkarte zu einem handlichen Sicherheitsmodul werden, wobei hohe Anforderungen an die Benutzungsfreundlichkeit gestellt werden müssen.

3.2.3.1 Speicherkarten

Speicherkarten bestehen in der Regel aus einem oder mehreren Speichern. Sie können zusätzlich mit einer Sicherheitslogik ausgestattet sein, die den Zugriff auf die Daten im Speicher regelt, welche im einfachsten Fall nur aus einem Schreib- oder Löschschutz für den Speicher oder einzelne Teile des Speichers besteht. Die integrierte Sicherheitslogik kann einfache Verschlüsselungen durchführen.

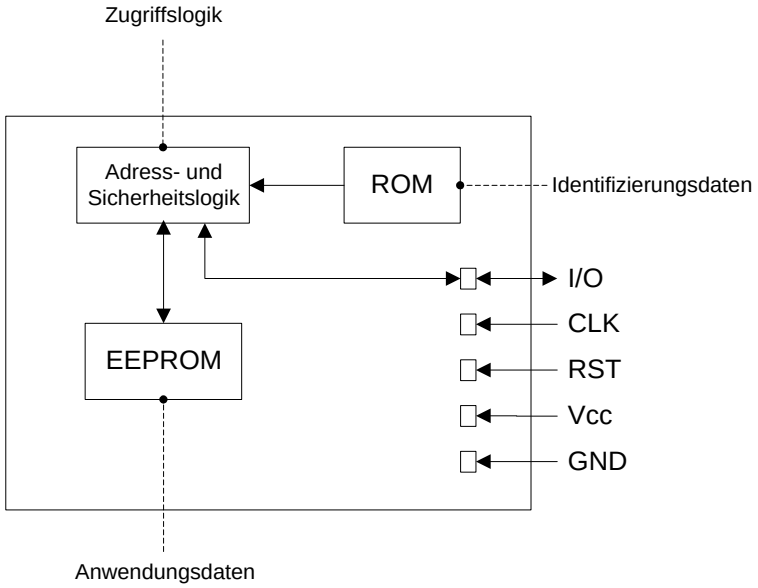


Abbildung 3-3: Speicherkarte mit Sicherheitslogik

Quelle: [Rankl, Effing 1996], S. 32

Für die Datenübertragung zur und von der Chipkarte ist in der ISO-Norm 7816-3 ein spezielles synchrones Übertragungsprotokoll definiert. Aufgrund ihrer anwendungsspezifischen Optimierung sind Speicherkarten sehr preisgünstig, jedoch in ihrer

Funktionalität eingeschränkt. Als Beispiel für Speicherkarten ohne Sicherheitslogik ist die Krankenversichertenkarte zu nennen, wohingegen die Telefonkarte eine Speicherkarte mit Sicherheitslogik ist.

3.2.3.2 Mikroprozessorkarten

Neben den Speicherkarten gibt es Mikroprozessorkarten, die zusätzlich zum ROM und EEPROM einen Prozessor und einen Arbeitsspeicher (RAM) haben.

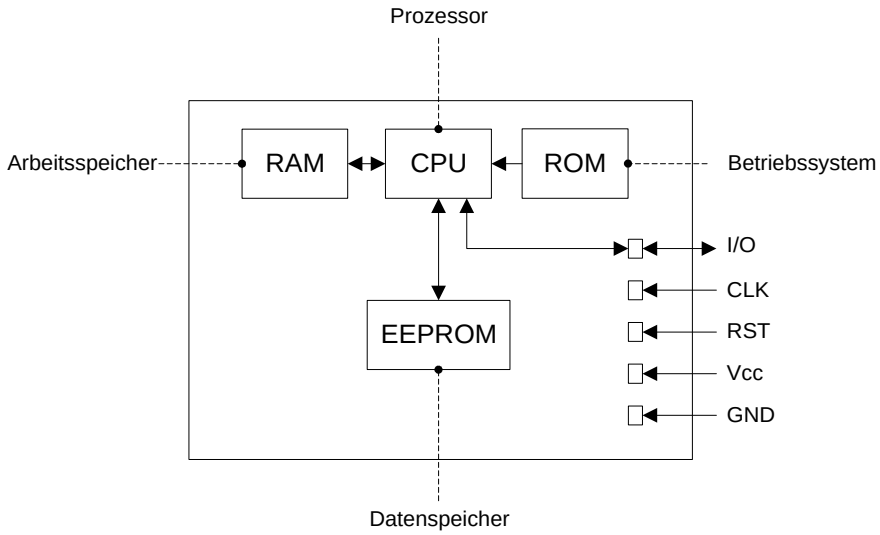


Abbildung 3-4: Mikroprozessorkarte
Quelle: [Rankl, Effing 1996], S. 32

Das ROM enthält das Betriebssystem oder zumindest Basisbefehle davon und wird während der Herstellung eingebrannt. Der Inhalt des ROM ist während der Lebensdauer des Chips nicht veränderbar. Das EEPROM ist ein nichtflüchtiger Speicher, in dem jederzeit Daten und auch Programmcode unter der Kontrolle des Betriebssystems gelesen und geschrieben werden können. Das RAM ist der Arbeitsspeicher. Dieser Speicher ist flüchtig. Alle darin gespeicherten Daten gehen verloren, wenn die Spannungsversorgung der Chipkarte abgeschaltet wird. Die serielle I/O-Schnittstelle besteht meist nur aus einem einzigen Register, über welches die Daten Bit für Bit übertragen werden. In der ISO-Norm 7816-3 sind verschiedene Übertragungsprotokolle beschrieben, wobei neben dem bitorientiertem „T = 0“ auch ein blockorientiertes „T = 1“ Übertragungsprotokoll spezifiziert ist.

Mikroprozessorkarten sind sehr flexibel bezüglich ihrer Anwendungsmöglichkeiten. Meist enthalten sie jedoch ein einziges Programm, das genau auf eine Anwendung zugeschnitten und optimiert ist. Moderne Chipkarten-Betriebssysteme ermöglichen hingegen, mehrere Anwendungen in einer einzigen multifunktionalen Chipkarte zu integrieren. Das ROM enthält in diesem Fall nur die Basisbefehle des Betriebssystems, und die anwendungsspezifischen Teile des Programms werden im EEPROM gespeichert (siehe Kapitel 3.5).

Der wichtigste Bestandteil einer Chipkarte ist der Microcontroller. Er besteht aus dem Prozessor, dem Adress- und Datenbus und den drei verschiedenen Speicherarten (RAM, ROM, EEPROM). Zusätzlich befindet sich noch eine einfache Schnittstellenbaugruppe im Controller, die die serielle Kommunikation übernimmt. Der Microcontroller steuert und überwacht alle Aktivitäten und ist ein vollständiger Computer.

Die in Chipkarten eingesetzten Prozessoren sind keine Spezialentwicklungen, sondern in anderen Bereichen bereits bewährte Bauelemente. Deshalb können von den Herstellern von Betriebssystemen bereits existierende Funktionsbibliotheken verwendet werden. Seit dem Jahre 1971, in dem die Firma Intel den ersten Microcontroller (4004) mit 2.300 Transistoren auf den Markt brachte, hat es eine große Weiterentwicklung gegeben. Produkte wie der Pentium Prozessor mit 3,1 Millionen Transistoren zeigen dies deutlich.

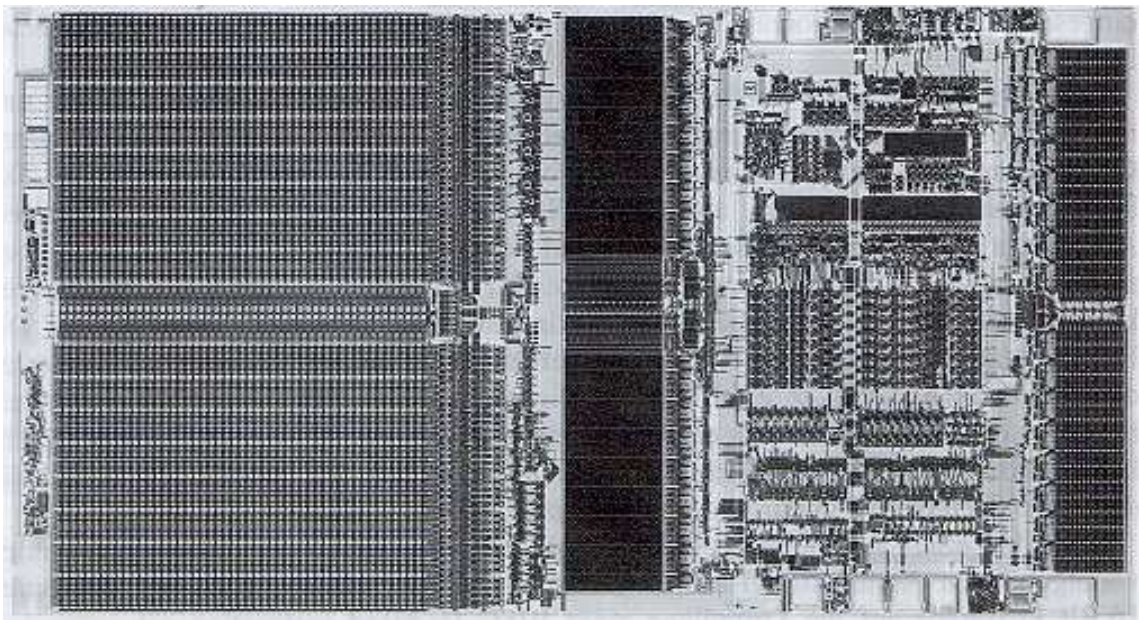


Abbildung 3-5: Chipkarten-Microcontroller

Quelle: [Rankl, Effing 1996], S. 67

Im Bereich der Chipkarten werden jedoch aus Platzgründen nicht die fortschrittlichsten Prozessoren verwendet. 200.000 Transistoren gelten hier bereits als ausreichend. Die meisten Prozessoren basieren auf einer CISC-Architektur (complete instruction set computer) und verwenden eine Befehlsbreite von acht Bit. Die Adreßbreite der acht Bit Prozessoren ist durchweg 16 Bit, mit der sich maximal 65.536 Byte adressieren lassen. Die Befehlssätze der Prozessoren orientieren sich entweder an der Motorola 6805 oder an der Intel 8051 Architektur. Zum Teil sind die vorhandenen Befehle je nach Halbleiterhersteller durch Erweiterungen ergänzt. Seit neuester Zeit gibt es 16 Bit oder 32 Bit Prozessoren, die auf der Grundlage RISC-ähnlicher Architekturen (reduced instruction set computer) beruhen.

Neben dem Prozessor sind die Speicher die wichtigsten Bestandteile. Sie dienen zur Ablage von Daten und Programmcode und haben die charakteristische Aufteilung in

RAM, ROM und EEPROM. Folgende Speicherarten werden für Chipkarten verwendet oder sind in der Entwicklung für zukünftige Anwendungen:

ROM (Read Only Memory)

Aus diesem Speicher kann nur gelesen werden. Er ist ein nichtflüchtiger Speicher, da seine Informationen durch eine feste Verdrahtung repräsentiert werden. Im ROM befindet sich das Betriebssystem oder zumindest Teile davon. Bereits bei der Produktion des Chips werden diese Informationen vom Halbleiterhersteller durch Belichten und Ätzen der Siliziumscheiben in den Chip eingebracht.

EEPROM (Electrical Erasable Programmable Read Only Memory)

Im EEPROM werden Daten und Programme gespeichert, die später wieder geändert oder gelöscht werden sollen. Wie bei der Festplatte eines PCs bleiben die Daten auch ohne Energiezufuhr erhalten. Das EEPROM ist einer der wenigen Halbleiterspeicher, die eine begrenzte Anzahl von Programmierzugriffen haben, wohingegen die Lesezugriffe unbegrenzt sind. Die Anzahl der möglichen Schreib- beziehungsweise Löschzugriffe kann in Abhängigkeit des halbleitertechnischen Aufbaues sehr unterschiedlich sein. Übliche Werte liegen zwischen 10.000 und 1.000.000 Zyklen.

Flash-EEPROM (Flash Electrical Erasable Programmable Read Only Memory)

Flash-EEPROMs haben einen ähnlichen Aufbau wie EEPROMs. Der wesentliche Unterschied liegt in der Art des Schreibvorgangs. Durch einen speziellen Effekt reduziert sich die Zeit eines Schreibvorgangs von 3 bis 10ms bei bisherigen EEPROMS auf 10µs. Die dafür notwendige Programmierspannung beträgt nur etwa 12V im Vergleich zu 17V. Die Integration von Flash-EEPROM-Zellen in Microcontrollern ist von den Halbleiterherstellern für die nahe Zukunft angekündigt.

FRAM (Ferroelectric Random Access Memory)

Eine weitere Neuentwicklung ist das FRAM, dessen Inhalt auch ohne Versorgungsspannung erhalten bleibt. Für diese Speicherart werden die Eigenschaften von ferroelektrischen Substanzen verwendet. Zur Programmierung werden nur 5V benötigt, die Zeit für einen Programmiervorgang beträgt 100ns und die Zugriffszyklen bewegen sich im Bereich von einer Billion. Jedoch ist damit auch die Anzahl der Lesezyklen begrenzt, was einen wesentlichen Nachteil dieser Technologie ausmacht. In heute gängigen Chipkarten wird diese Speicherart noch nicht verwendet, eine Anwendung wurde jedoch auch hier von Seiten der Halbleiterhersteller in naher Zukunft angedeutet.

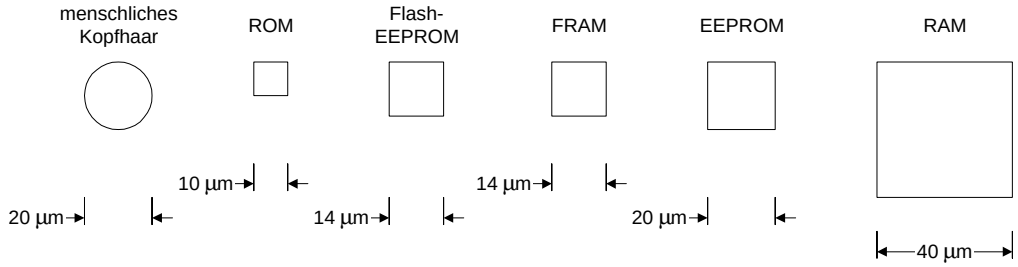
RAM (Random Access Memory)

Das RAM ist der flüchtige Speicher einer Chipkarte, in dem die Daten während einer Sitzung gehalten werden. Ohne Spannungsversorgung ist der Inhalt eines RAMs nicht mehr definiert. Die Zugriffe sind unbegrenzt. In Chipkarten wird statisches RAM

(SRAM) - im Gegensatz zu einem RAM (DRAM) - verwendet, da es unabhängig von der Taktversorgung der Chipkarte arbeiten soll. Nachteil eines RAMs ist der enorme Platzbedarf im Vergleich zu anderen Speicherarten.

Die verschiedenen Speicherarten benötigen unterschiedlich viel Platz pro Bit. Dies muß bei der Entwicklung berücksichtigt werden, da die Fläche des gesamten Chips 25mm^2 nicht übersteigen darf, um die Bruchgefahr zu minimieren ([Rankl, Effing 1996], S. 67).

Eine RAM-Zelle benötigt etwa viermal soviel Platz wie eine EEPROM-Zelle und diese wiederum viermal so viel wie eine ROM-Zelle. Deshalb sind Chipkarten so sparsam mit RAM ausgestattet.



Die Größenangaben sind circa Werte und beziehen sich auf 0,8 µm Technologie.

Abbildung 3-6: Vergleich des Platzbedarfes für ein Bit verschiedener Speicherarten
Quelle: [Rankl, Effing 1996], S. 69

Seit neuestem gibt es eine Speichertechnologie, die sogenannten Flash-EEPROM-Zellen, deren Größe etwa der Hälfte der EEPROM-Zellen entspricht und sehr viel schnellere Schreib- und Löschzugriffe erlaubt. Zum Vergleich: Der Durchmesser des ersten Transistors von 1959 war 764µm.

3.2.3.3 Optische Speicherkarten

Optische Speicherkarten verfügen über die Möglichkeit, mehrere Megabyte Daten aufzunehmen. Durch diese hohe Speicherkapazität werden diese Chipkarten dort eingesetzt, wo die Speicherkapazität von Chipkarten nicht mehr ausreicht. Nach dem heutigen Stand der Technik können optische Speicherkarten allerdings nur ein einziges Mal beschrieben und dann nicht mehr gelöscht werden.

Als sogenannte Hybridkarten (Abbildung 3-7) werden Chipkarten bezeichnet, die die Speicherkapazität einer Optischen Speicherkarte mit der Intelligenz einer Chipkarte kombinieren. So können beispielsweise Daten in verschlüsselter Form im optischen Speicherbereich abgelegt werden und der Schlüssel zum Entschlüsseln im geheimen Speicher des Chips.

Anwendung können solche Chipkarten zum Beispiel im Gesundheitswesen zur Speicherung von Patientendaten oder von Röntgenbildern finden. Lese- und Schreibgeräte für optische Speicherkarten sind jedoch zur Zeit noch sehr teuer, so daß die Anwendungen dieser Chipkarten bisher sehr begrenzt sind.

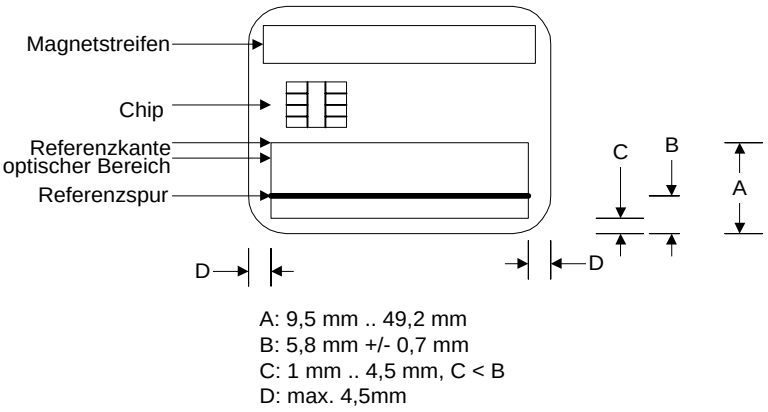


Abbildung 3-7: Lage des optischen Speicherbereiches auf der ID-1 Karte
Quelle: [Rankl, Effing 1996], S. 35

3.3 Physikalische und elektrische Eigenschaften

Viele Normen über die physikalischen Eigenschaften sind nicht speziell für Chipkarten spezifiziert worden, sondern stammen aus der Zeit der Präge- und Magnetstreifenkarten und sind somit für alle Kartentypen gültig. Das unmittelbar auffallende Merkmal ist das Format der Chipkarte. Weitere Teile der physikalischen Eigenschaften sind eine elektrische Schnittstelle (kontaktbehaftete Chipkarte), Magnetstreifen, Hochprägung oder Hologramme auf der Karte. Rein mechanischer Natur sind ein Großteil der physikalischen Eigenschaften wie zum Beispiel Größe, Biege- und Torsionsfestigkeit. Weiterhin sind Licht-, Temperaturempfindlichkeit und Feuchtebeständigkeit von Bedeutung.

3.3.1 Formate, Kartenmaterial

Karten im ID-1 Format gibt es schon sehr lange. In diesem bekannten Format werden fast alle Chipkarten hergestellt. Die ISO-Norm 7810 legt die Größe und Bezeichnung ID-1 fest. Eine Karte im ID-1 Format ist 85,6mm lang und 54mm breit, der Radius der Ecken beträgt 3,18mm. Die Dicke einer Karte beträgt 0,76mm.

Für kleine und vor allem leichte Geräte, wie zum Beispiel Mobiltelefone, wurde ein weiteres Format entwickelt, das ID-000. Diese Karte ist sehr klein, da sie meist als „Einschubkarte“ in Geräte eingesetzt wird. Zur Zeit kommen diese Art Karten im Mobiltelefonbereich (GSM) in Endgeräten zum Einsatz, in denen sehr wenig Platz ist. Ein weiteres Format ist das ID-00, das ein Mittel zwischen ID-1 und ID-000 darstellt. Die Definition von ID-00 ist noch relativ neu und hat sich bisher noch nicht durchgesetzt.

Die drei Formate können von den größeren Kartenkörpern zu den kleineren durch Stanzen überführt werden. Im GSM-Bereich wird zum Teil dem Kunden eine ID-1 Karte ausgehändigt, die so vorgestanzt ist, daß sich die ID-000 Karte heraustrennen läßt.

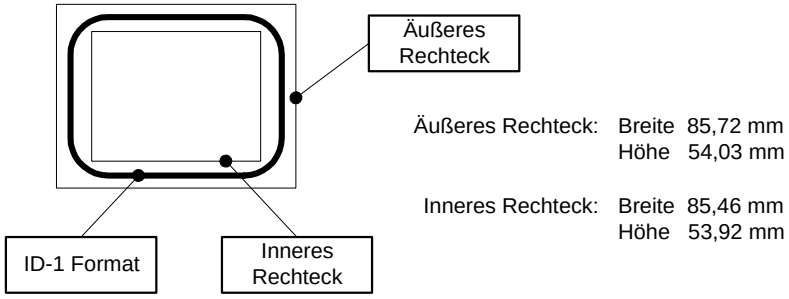


Abbildung 3-8: Definition des ID-1 Formats für Chipkarten

Quelle: [Rankl, Effing 1996], S. 38

Das am meisten verwendete Kartenmaterial ist PVC (Poly Vinyl Chlorid). Nachteile sind die begrenzte Lebensdauer durch Alterung und die geringe Temperaturbeständigkeit. Seit einiger Zeit werden Karten aus ABS (Acrylnitril Butadin Styrol) gefertigt, die sich durch Festigkeit und Temperaturbeständigkeit auszeichnen. Weiterhin wird als Ersatzmaterial für PVC Polyester (PET) eingesetzt. Ein weiteres übliches Kartenmaterial ist Polycarbonat (PC), das dort zum Einsatz kommt, wo hohe Festigkeit und Langlebigkeit gefordert werden.

3.3.2 Kontaktlose Karten

Bei kontaktbehafteten Chipkarten erfolgt die Kommunikation über acht Kontakte (siehe Kapitel 3.3.3). Kontakte können jedoch häufig Fehlerquellen in elektromagnetischen Systemen sein. Störungen können durch Verschmutzung oder Abnutzung der Kontakte entstehen.

Diese technischen Probleme können von kontaktlosen Chipkarten umgangen werden. Vorteile von kontaktlosen Chipkarten können in der Tatsache liegen, daß sie nicht in einen Kartenleser eingesteckt werden müssen. Es gibt Systeme, die über eine Entfernung von bis zu einem Meter funktionieren. Dies ist in Zugangskontrollsystemen sinnvoll, bei denen eine Tür oder ein Drehkreuz geöffnet werden soll. Ein großes Anwendungsgebiet stellt der öffentliche Personennahverkehr dar, wo in möglichst kurzer Zeit möglichst viele Personen erfaßt werden sollen.

Eine kontaktlose Chipkarte muß die Energieübertragung für die Versorgung der integrierten Schaltung über eine gewisse Distanz realisieren. Ebenso muß die Übertragung des Taktsignals sowie die Datenübertragung von und zu der Chipkarte ermöglicht werden. Die am häufigsten verwendeten Verfahren bei kontaktlosen Chipkarten sind die Übertragung mittels Mikrowellen, optische Übertragung oder kapazitive und induktive Kopplung.

Diese Technik hat jedoch nicht den gleichen Entwicklungsstand erreicht, wie er bei den kontaktbehafteten Chipkarten vorliegt. Ein Grund dafür ist die vergleichsweise teure Produktion von kontaktlosen Chipkarten, da zusätzliche Bauteile wie Übertragungsspulen oder Kondensatorplatten in die Chipkarte integriert werden müssen. Zur Zeit gibt es noch keine Massenanwendungen von kontaktlosen Chipkarten, so daß der breite Einsatz dieser Kartenart bisher noch nicht zu verzeichnen ist.

3.3.3 Kontaktbehaftete Karten

Findet die Energieversorgung und die Datenübertragung einer Chipkarte über Kontakte statt, so ist dazu eine galvanische Verbindung notwendig, die aus sechs bis acht vergoldeten Kontakten besteht.

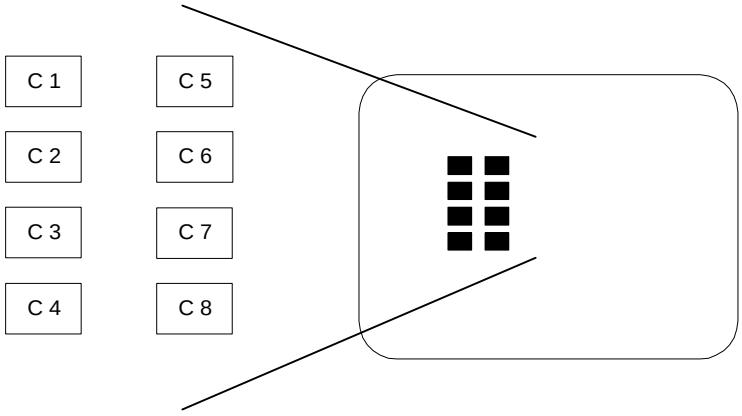


Abbildung 3-9: Kontakte bei Chipkarten

Quelle: [Gentz 1997], S. 6

Die Kontake sind von C 1 bis C 8 durchnummeriert. Die Lage der Kontakte bezüglich des Kartenkörpers, ihre Größe und die Signalbelegung (Tabelle 3-1) sind in der ISO-Norm 7816-2 spezifiziert.

Kontakt	Signal	Bedeutung
C 1	VCC	Versorgungsspannung
C 2	RST	Reset
C 3	CLK	Takt
C 4	RFU	Für zukünftige Anwendungen reserviert
C 5	GND	Masse
C 6	VPP	Programmierspannung
C 7	I/O	Dateneingang / -ausgang
C 8	RFU	Für zukünftige Anwendungen reserviert

Tabelle 3-1: Kontaktbelegung nach ISO-Norm 7816-2

Quelle: [Rankl, Effing 1996], S. 60

Die minimale Größe der Kontakte beträgt 1,7mm in der Höhe und 2mm in der Breite. Die maximale Größe ist nicht festgelegt, ergibt sich jedoch aus dem Zwang, die einzelnen Kontakte voneinander zu isolieren. Die Kontaktform selbst ist nicht durch Norm festgelegt, sondern ist abhängig vom jeweiligen Chiphersteller.

3.3.4 Sicherheitsmerkmale

Chipkarten werden im wesentlichen als Berechtigungen für bestimmte Dinge oder zur Identifizierung des Inhabers genutzt ([Rankl, Effing 1996], S. 52ff). Bisher wurden verschiedene visuelle Sicherheitsmerkmale entwickelt, die nicht durch eine Maschine, sondern durch einen Menschen auf ihre Echtheit hin überprüft werden konnten. Früher war das die einzige Möglichkeit, Karten auf Echtheit zu prüfen.

Merkmale	**Name und Beschreibung**
Unterschriftsstreifen	Einfaches Verfahren zur Identifizierung. Ein mit dem Kartenkörper fest verbundener Unterschriftsstreifen. Einmal beschrieben kann er nicht mehr verändert werden, ohne ihn zu zerstören (radierfest).
Guillochen	Meist runde oder ovale geschlossene und miteinander verwobene Linienfelder, die farbig auf Folien aufgedruckt unter der obersten transparenten Schicht der Karte eingefügt werden.
Mikroschrift	Mikroschrift-Linien basieren auf der Sicherheit von fein gedruckten Strukturen. Sie sind für das Auge nur als Linie wahrnehmbar und können nur mit der Lupe erkannt werden.
Hologramm	Hologramme werden in einem Heißprägeverfahren auf die Karte aufgebracht und können von dieser nicht mehr gelöst werden, ohne sie zu zerstören.
Multiple Laser Image	Multiple Laser Image (MLI) sind eine Art Kippbild, das einfachen Hologrammen sehr ähnlich ist. Es können karten-individuelle Informationen, beispielweise der Name des Kartenbesitzers als Kippbild aufgebracht werden.
Lasergravur	Lasergravur nennt man die Schwärzung von speziellen Kunststoffschichten durch Verbrennen mit einem Laser. Im Gegensatz zum Hochprägen ist sie eine besonders sichere Art, individuelle Daten auf die Karte zu schreiben.
Hochprägung	Hochprägung ist eine weitere Variante des Aufbringens von Benutzerdaten auf den Kartenkörper. Da die Hochprägung sehr leicht manipulierbar ist, werden einige Zeichen überlappend in das Hologramm gelegt.

Tabelle 3-2: Übersicht über gängige Sicherheitsmerkmale von Chipkarten

Durch heutige integrierte Microcontroller und damit verbundene kryptographische Verfahren sind diese Echtheitsmerkmale etwas in den Hintergrund getreten. Sie haben jedoch noch solange eine Bedeutung, solange Karten durch Menschen und nicht durch Maschinen auf Echtheit geprüft werden sollen. Tabelle 3-2 bietet einen Überblick über visuelle Sicherheitsmerkmale.

3.3.5 Spannungsversorgung

Die Versorgungsspannung einer Chipkarte beträgt 5V mit einer Toleranz von ±10%. Diese Spannung ist für alle auf dem Markt und in der Anwendung befindlichen Chipkarten der Standardwert. Die Entwicklung im Mobilfunkbereich führt jedoch bei Chipkarten zu einem Spannungsbereich von 3 bis 5V mit einer Toleranz von ±10%, da alle Bauteile eines Mobilfunktelefons in 3V Technik verfügbar sind. Daraus ergibt sich letztendlich ein effektiver Spannungsbereich von 2,7 bis 5,5V.

3.4 Lebenszyklus einer Chipkarte

Eine Chipkarte setzt sich aus unterschiedlichen Komponenten zusammen, dem Kartenkörper und dem Chipmodul. Der Lebenszyklus einer Chipkarte wird durch die unterschiedlichen Phasen der Herstellung, der Benutzung und des Recyclings der Chipkarte bestimmt. Dabei kann man fünf Phasen unterscheiden [Weikmann 1997]:

- Phase 1: Chip- und Kartenherstellung
- Phase 2: Kartenkomplettierung
- Phase 3: Anwendungsvorbereitung
- Phase 4: Anwendungsnutzung
- Phase 5: Recycling

Am Anfang der Chipkartenherstellung kann zwischen den beiden parallelen Schritten der Chipherstellung und der Herstellung des Kartenkörpers unterteilt werden. Zu einem späteren Zeitpunkt wird der Chip dann in den Kartenkörper integriert und ergibt dadurch die Chipkarte. Dann erfolgt ein sequentieller Prozeß der Kartenkomplettierung und der Vorbereitung der Anwendungsnutzung. Das Ende einer Chipkarte liegt in ihrem Recycling. Der Lebenszyklus einer Chipkarte wird in Abbildung 3-10 dargestellt.

Phase 1 beginnt mit der Software-Erstellung des Chipkarten-Betriebssystems und der darauf aufbauenden Anwendung. Der Programmcode ist dabei an den jeweiligen Chiptyp angepaßt und ist nur mit zusätzlichem Aufwand auf andere Chiptypen portierbar. Das Betriebssystem wird als ROM-Maske entwickelt, das sich im späteren ROM des Microcontrollers befindet. Der Halbleiterhersteller benutzt die erhaltene ROM-Maske, um daraus das ROM zu erstellen. Die Produktion des Halbleiters erfolgt mit den dafür üblichen Verfahren von zum Beispiel Dotierung, Belichten, Ätzen. Nach der Fertigung werden die Microcontroller getestet. Anschließend werden die Chipmodule vervollständigt und erneut getestet. Parallel dazu wird der Kartenkörper aus mehreren Einzellagen verschiedener Kunststoffolien hergestellt. Die Heißverklebung der einzelnen Folien wird Laminierung genannt. Während dieses Produktionsschrittes können Unterschrift-, Magnetstreifen und Sicherheitsmerkmale eingebracht werden. Der Kartenkörper wird in dem gewünschten Format ausgestanzt. Dann erfolgt das Fräsen der Modulaussparung, damit die Module implantiert werden können. In diesem Produktionsschritt kommen Chipmodul und Kartenkörper zusammen.

In **Phase 2** des Lebenszyklus der Chipkarte, werden nach einem elektrischen Test der Chipkarte die notwendigen Daten der Anwendungsanbieter beziehungsweise der Kartenherausgeber auf die Chipkarte gebracht. Das Betriebssystem befindet sich bereits zu großen Teilen im ROM, nun müssen geheime Schlüssel in den geschützten Bereich des EEPROM geladen werden, damit später die Programmteile der Anwendung nach erfolgreicher Authentisierung mit diesem Schlüssel geladen werden können. Das Laden dieser EEPROM-Teile wird als Komplettieren des Betriebssystems bezeichnet.

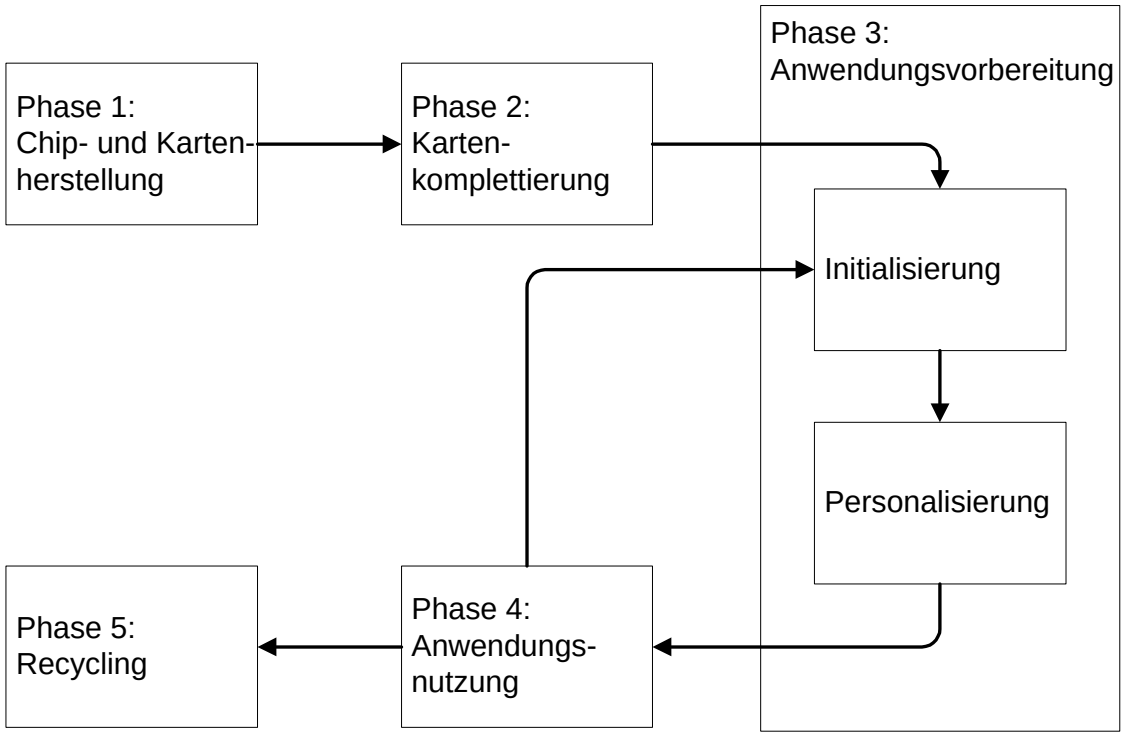


Abbildung 3-10: Lebenszyklus einer Chipkarte

Die Vorbereitung der Anwendung erfolgt in **Phase 3**. Nach der Komplettierung werden zunächst alle globalen Daten geladen. Dies sind alle Daten der Anwendung und alle personenunabhängigen Daten, die sich von Karte zu Karte nicht ändern. Dies wird als Initialisierung bezeichnet. Es werden alle notwendigen Dateien (MF, DFs und EFs) erzeugt und soweit wie möglich mit Anwendungsdaten gefüllt. Um den Produktionsprozeß so effizient wie möglich zu gestalten, wird zwischen globalen, allgemeinen Daten und individuellen, personenbezogenen Daten unterschieden.

Die personenbezogenen Daten werden in der sogenannten Personalisierung auf die Chipkarte gebracht. Dies sind beispielsweise Name, Anschrift oder auch kartenbezogene Schlüssel. Je nach Sensitivität der individuellen Daten, werden kryptographische Verfahren zur Datenübertragung in die Chipkarte verwendet. Der Personalisierer soll keine Kenntnis über die von ihm eingebrachten Daten haben. Um das zu ermöglichen, werden die Personalisierungsdaten vom Kartenherausgeber verschlüsselt. In einem speziellen Sicherheitsmodul an der Personalisierungsmaschine werden die verschlüsselten Personalisierungsdaten „umgeschlüsselt“ und in die Chipkarte eingegeben.

In **Phase 4** des Lebenszyklus der Chipkarte, der eigentlichen Benutzung einer Anwendung, können weitere Anwendungen geladen werden, die bereits bei der Herstellung eingeplant wurden und deren geheime Authentisierungsschlüssel bereits gespeichert sind. Andere neue Anwendungen können nicht geladen werden, ohne daß auf eine Authentisierung verzichtet wird.

Phase 5 stellt bisher den am wenigsten erforschten Teil des Lebenszyklus dar. Ein Großteil nicht mehr benutzter Chipkarten wird nicht einer Entsorgung oder dem Recycling zugeführt, sondern befindet sich im Besitz privater Sammler. Das Recycling der aus teilweise verschiedenen Kunststoffen laminierten Chipkarten ist heutzutage schwer möglich. Die heterogenen Materialien lassen sich nur schwer in homogene Materialien trennen. So bleibt nur die Verbrennung bei hoher Temperatur, was auf Dauer keine umweltfreundliche Lösung darstellt. Es ist notwendig, über eine sinnvolle Möglichkeit des Recyclings nachzudenken, da abgelaufene und nicht mehr benötigte Chipkarten immer noch gültige Schlüssel enthalten können. Dies stellt ein erhebliches Sicherheitsrisiko dar, wenn geheime Schlüssel und aktuelle Anwendungen auf nicht mehr benötigten Chipkarten nicht gelöscht werden.

Die meisten Chipkarten werden heutzutage für eine einzige Anwendung hergestellt. Ist diese Anwendung nicht mehr gewünscht oder soll eine andere Anwendung realisiert werden, wird die Chipkarte meistens durch eine neue Chipkarte ersetzt. Einige Chipkarten erlauben es, im nachhinein weitere Anwendungen auf die Chipkarte zu integrieren. Dafür werden Speicherbereiche von Anfang an für weitere Anwendungen reserviert. Wie bereits in Phase 3 beschrieben, muß zu Beginn der Chipkartenherstellung bereits bekannt sein, wieviele Anwendungen mit welchem Speicherbedarf zusätzlich nachgeladen werden sollen. Dafür müssen die entsprechenden Schlüssel bereitgehalten werden, die ein authentisches Nachladen ermöglichen.

Das authentische Nachladen von Anwendungen stellt jedoch ein großes Problem dar. Gerade bei multifunktionalen Chipkarten ist es sinnvoll, einige Anwendungen erst später nachzuladen und andere Anwendungen nur für einen bestimmten Zeitraum zu nutzen, ohne daß von vornherein klar ist, um welche Anwendungen es sich handelt. Bei multifunktionalen Chipkarten stellt sich die Anforderung, einige Anwendungen nur für begrenzte Zeit zu nutzen, dann neue Anwendungen für einen anderen Zeitraum zu nutzen und eventuell nach einiger Zeit die ersten Anwendungen wieder auf die Chipkarte zu laden.

In dieser Situation muß sichergestellt sein, daß tatsächlich nur authentische Anwendungen geladen werden können. Es muß feststellbar sein, ob die Anwendung von ihrem angegebenen Anbieter stammt, ob sie die angegebene Funktionalität erfüllt und keine maliziöse Nebenwirkung hat. Ein Schritt zur Lösung dieser Probleme sind zertifizierte Anwendungen, die mit einer digitalen Signatur von einer zertifizierten Stelle ausgestattet sind (siehe Kapitel 4.3). Diese Signatur kann jedoch nur die Authentizität einer Anwendung nachweisen, nicht jedoch ihre funktionale Korrektheit. Dieses Problems muß man sich bewußt sein, wenn man die Thematik von multifunktionalen Chipkarten diskutiert. In diesem Bereich besteht ein noch großer Handlungsbedarf (siehe Kapitel 8).

3.5 Chipkarten-Betriebssysteme

Dieses Unterkapitel beschreibt die wesentlichen Aufgaben von Betriebssystemen im allgemeinen und von Chipkarten-Betriebssystemen im speziellen am Beispiel von STARCOS [Kock 1997/1999] und am Beispiel des Betriebssystems der Java-Card.

Ein Betriebssystem wird wie folgt definiert:

> *„Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen Betriebsarten des digitalen Rechensystems bilden und insbesondere die Abwicklung von Programmen steuern und überwachen.“ [DIN 44300]*

Hierbei wird deutlich, daß keine Mindestgröße für Betriebssysteme gefordert wird. Lediglich die Funktionalität ist entscheidend. Ein Betriebssystem verwaltet alle Ressourcen eines Computersystems [Tanenbaum 1995]. Es hat die Aufgabe, eine geordnete und kontrollierte Zuteilung der Prozessoren, Speicher und I/O-Geräte für die konkurrierenden Programme durchzusetzen. Ein Betriebssystem erfüllt folgende Hauptaufgaben:

- Speicherverwaltung:
  Verwaltung der freien und belegten Speicherbereiche, Zuteilung von Speicherbereichen an Prozesse, wenn sie diesen benötigen, Freigabe, wenn sie abgeschlossen sind. Durchführung von Auslagerung zwischen dem Arbeitsspeicher und einem Erweiterungsspeicher, falls der Arbeitsspeicher nicht groß genug ist, um alle Programme auf einmal zu halten.
- Prozeßverwaltung:
  Kreierung, Betreuung, Abschluß und Entfernung eines Prozesses im System.
- Geräteverwaltung:
  Überwachung und Steuerung aller I/O-Geräte eines Computers, Weiterleitung von Befehlen an die I/O-Geräte, Abfangen von Unterbrechungen und Fehlerbehandlung. Bereitstellung einer einfachen und leicht zu benutzenden Schnittstelle zwischen den I/O-Geräten und dem Rest des Systems.

Das Betriebssystem stellt somit dem Anwendungssystem definierte Funktionalitäten zur Verfügung. Ein Chipkarten-Betriebssystem und seine wesentlichen Aufgaben lassen sich differenziert von den oben angegebenen Aufgaben wie folgt charakterisieren [Weikmann 1997]:

Allgemeine derzeitige Aufgaben von Chipkarten-Betriebssystemen:

- Miniatur-Betriebssystem mit wenigen kByte Speichergröße
- Single Processing System (derzeit noch)
- Maschinenschnittstelle in Form eines seriellen Interfaces beziehungsweise Modulationsinterface bei kontaktlosen Chipkarten

Spezielle Sicherheitsaufgaben:

- Verwaltung einer Sicherheitshardware auf Basis von Single-Chip-Microcontrollern
- Sichere Speicherung von Daten
- Kryptographische Verfahren zur Authentifizierung von Systemkomponenten und Ent- und Verschlüsselung von vertraulichen und geheimen Daten
- Gewährleistung der informationstechnischen Sicherheit von Anwendungen durch eine spezielle Struktur des Datei-Systems

Die historische Entwicklung der Chipkarten-Betriebssysteme von 1980 bis heute läßt sich anhand der Chipkarten für die deutschen Mobiltelefonnetze aufzeigen. Seit 1987 wird im C-Netz eine Chipkarte eingesetzt, für deren Anwendung ein spezielles Betriebssystem mit eigenem Übertragungsprotokoll, Spezialbefehlen und einem eigenen Dateiaufbau konzipiert wurde.

Die Weiterentwicklung dieser Speziallösung zeigt sich an den ersten GSM-Karten, die wesentlich offener und multifunktionaler aufgebaut sind. Wichtig für die Kompatibilität zwischen unterschiedlichen Betriebssystemen sind internationale Standards. Bei der Konzipierung der GSM-Karten gab es schon die ersten Normenentwürfe für den Befehlssatz und die Datenstrukturen von Chipkarten.

Die gegenwärtigen Chipkarten-Betriebssysteme für GSM besitzen Funktionen wie Speicherverwaltung, mehrere Dateistrukturen und einen großen Befehlssatz. Einige weisen mehrere Übertragungsprotokolle und meist sehr aufwendige Zustandsautomaten auf. Mit diesen modernen Betriebssystemen ist es möglich, mehrere Anwendungen auf einer Chipkarte zu integrieren, ohne daß diese sich gegenseitig beeinflussen [Rankl, Effing 1996].

Aus der Übersicht über gängige Chipkarten-Normen, die sich im Anhang Normen befindet, werden an dieser Stelle die grundlegenden Normen für Chipkarten-Betriebssysteme aufgelistet (siehe Tabelle 3-2).

Im folgenden werden die Grundlagen eines Chipkarten-Betriebssystems sowie dessen Speicherorganisation und Datenstrukturen aufgezeigt. Anschließend wird die Übertragung der Daten zwischen Terminal und Chipkarte skizziert. Dazu wird ausschließlich die ISO-Norm 7816-4 verwendet.

Norm	Name und Beschreibung
prETS 300 608 (1995) und GSM 11.11 (1995)	(Phase 2) - Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.11) Ist identisch mit: GSM 11.11 Version 4.16.0 European digital cellular telecommunications systems Spezifizierte Schnittstelle zwischen der Chipkarte dem Mobiltelefon für GSM
EN 726-3 (1994)	Identification card systems - Telecommunications integrated circuit(s) card and terminals - Part 3: Application independent card requirements Definition von Dateistrukturen, Befehle, Returncodes, Dateien für Chipkarten im Bereich der Telekommunikation.
ISO-Norm 7816-4 (1995)	Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 4: Interindustry commands for interchange Definition der Dateiorganisation, Dateistrukturen, Sicherheitsarchitekturen, Beschreibung der Befehle für Chipkarten
prEN 1546-3 (1995)	Identification card systems - Intersector electronic purse Part 3: Data Elements and interchanges Beschreibung der Datenelemente, Dateien, Befehle der branchenübergreifenden elektronischen Geldbörse
EMV-2 (1995)	Integrated Circuit Card Specifications for Payment Systems - Part 2: Data Elements and Commands Spezifikation der APDUs, logischen Kanäle, Befehle für Kreditkarten mit Chip

Tabelle 3-3: Normen für Chipkarten-Betriebssyssteme

3.5.1 Sicherheitsanforderungen

Im Gegensatz zu den allgemein bekannten Betriebssystemen weisen Chipkarten-Betriebssysteme keine Benutzungsoberfläche oder Zugriffsmöglichkeiten auf externe Speichermedien auf, denn für sie gelten ganz andere Anforderungen als für die gängigen Betriebssysteme. Im Vordergrund stehen hierbei die Sicherheit bei der Programmausführung und der geschützte Zugriff auf Daten [Rankl, Effing 1996].

Das Betriebssystem und die Hardware einer Chipkarte haben die Aufgabe, die folgenden drei Grundbedrohungen abzuwehren [Weikmann 1997]:

- Verlust der Vertraulichkeit durch Ausspähen beziehungsweise unbefugte Preisgabe von Programmen und Daten, wie beispielsweise Schlüssel, PIN und Anwenderdaten
- Verlust der Integrität durch Manipulation beziehungsweise die nicht autorisierte Veränderung von Informationen wie Identifikationsnummer und Zählerstand von elektronischen Geldbörsen
- Verlust der Verfügbarkeit durch unberechtigte Vorenthaltung von Informationen beziehungsweise Beeinträchtigung der Funktionalität eines Systems

Hieraus wird deutlich, daß Chipkarten-Betriebssysteme sehr zuverlässig und robust sein müssen. Von außen kommende Befehle dürfen keinen unerwünschten Einfluß auf Funktionalität und Sicherheit des Systems ausüben. Systemzusammenbrüche oder unkontrollierte Reaktionen auf fehlerhafte Befehle müssen ausgeschlossen sein.

Ferner darf es weder Falltüren noch andere Hintereingänge für Systemprogrammierer geben, so daß es gänzlich unmöglich ist, am Betriebssystem vorbei, Daten unautorisiert auszulesen oder zu verändern.

Eine weitere Anforderung ist die enorme Leistungsfähigkeit. Die kryptographischen Funktionen müssen in sehr kurzer Zeit ablaufen. Es ist daher notwendig, den Programmcode dahingehend zu optimieren, daß eine schnellstmögliche Abarbeitung der Befehle erfolgen kann.

Die Benutzung von kryptographischen Modulen stellt besondere Anforderungen an Chipkarten-Betriebssysteme im speziellen und an Betriebssysteme im allgemeinen. Diese Anforderungen sind im Federal Information Processing Standard (FIPS) des Department of Commerce der USA zusammengefaßt [FIPS 140-1].

Diese Norm spezifiziert Anforderungen an kryptographisch-basierte Sicherheitssysteme, die zum Schutz von sensitiven Daten genutzt werden. Der Schutz eines kryptographischen Moduls innerhalb eines Sicherheitssystems ist notwendig, um die Vertrauenswürdigkeit und Integrität der Daten zu gewährleisten, die von eben diesem kryptographischen Modul geschützt werden sollen. Die Norm formuliert die Sicherheitsanforderungen, die von kryptographischen Modulen erfüllt werden müssen und unterscheidet dabei zwischen vier verschiedenen Sicherheitsniveaus. Folgende Sicherheitsniveaus werden in dieser Norm definiert.

- **Sicherheitsniveau 1**: Sicherheitsniveau 1 stellt das niedrigste Sicherheitsniveau zur Verfügung. Hier werden Basisanforderungen an kryptographische Module definiert, zum Beispiel dürfen nur solche kryptographische Algorithmen verwendet werden, die FIPS-bewährte Algorithmen sind. Sicherheitsniveau 1 bezieht sich ausschließlich auf Einzelbenutzer- und Einzelprozeßsysteme und es erlaubt, softwarebasierte kryptographische Funktionen auf einem allgemeinen PC (general purpose PC) auszuführen.

- **Sicherheitsniveau 2**: Zusätzlich zu Sicherheitsniveau 1 werden Anforderungen an die physikalische Sicherheit gestellt. Es werden Anforderungen an äußere Schichten und Siegel gestellt, die Angriffe nachweisen können (Tamper Evident Coatings and Seals) und an diebstahlsichere Schlösser (Pick Resistant Locks). Level 2 unterstützt rollenbasierte Authentisierung und erlaubt softwarebasierte Kryptographie auf einem C2-zertifizierten oder vergleichbarem Betriebssystem (siehe Kapitel 2.2).
- **Sicherheitsniveau 3**: Sicherheitsniveau 3 fordert eine erweiterte physikalische Sicherheit. In diesem Level muß ein Angreifer daran gehindert werden, Zugriff auf sicherheitskritische Daten in dem kryptographischen Modul zu bekommen. Sobald die Oberfläche des Moduls beschädigt wird oder sonstige physikalische Schäden an der äußeren Ummantelung des Moduls entstehen, werden sicherheitskritische Parameter gelöscht (zeroised). Level 3 unterstützt identitätsbasierte Authentisierung, die strenger als die rollenbasierte Authentisierung aus Level 2 ist. Datenein- und Ausgänge für sicherheitskritische Parameter müssen physikalisch von den restlichen Datenein- und Ausgängen getrennt sein. Level 3 fordert ein B1-zertifiziertes oder vergleichbares Betriebssystem (siehe Kapitel 2.2).
- **Sicherheitsniveau 4**: Dies ist das höchste Sicherheitsniveau. Es fordert einen vollständigen physikalischen Schutz rund um das kryptographische Modul. Zusätzlich zu Sicherheitsniveau 3 schützt es vor ungewollten Spannungs- und Temperaturschwankungen. Es setzt ein B2-zertifiziertes oder vergleichbares Betriebssystem voraus (siehe Kapitel 2.2).

Die Anforderungen betreffen verschiedene Bereiche, die in engem Bezug zum sicheren Design und zur Implentierung von kryptographischen Modulen stehen. Diese Bereiche beinhalten Basisdesign und -dokumentation, Modulschnittstellen, autorisierte Rollen und Dienste, physikalische Sicherheit, Zustandsautomaten, Softwaresicherheit, Betriebssystemsicherheit, Schlüsselmanagement, kryptographische Algorithmen, elektromagnetische Verträglichkeit (EMV), und Selbsttests. Eine Übersicht über die einzelnen Anforderungen für die verschiedenen Bereiche in den einzelnen Sicherheitsniveaus findet sich in Tabelle 3-4.

Wird ein kryptographisches Modul eingesetzt, sollte es an allen, in diesem Standard aufgelisteten Anforderungen getestet werden. Je nachdem auf welchem Sicherheitsniveau es eingesetzt werden soll, muß es diese erfüllen. In Bereichen, die keine Unterscheidung zwischen den unterschiedlichen Sicherheitsniveaus haben, müssen die genannten Anforderungen für alle Sicherheitsniveaus erfüllt werden. Viele der Anforderungen sind spezielle Anforderungen an die Dokumentation. Eine ausführliche Beschreibung dieser Anforderungen findet sich im Anhang A der Norm FIPS Pub 140-1.

Aus dieser Norm sind die Bereiche für Zustandsautomaten und Betriebssysteme für diese Arbeit relevant und wurden aus diesem Grund hier erwähnt. Die vorliegende Arbeit hat zum Ziel, ein Sicherheitsmodell (siehe Kapitel 6) für multifunktionale

Chipkarten im elektronischen Zahlungsverkehr zu entwickeln. Das Sicherheitsmodell (R&A-Modell) wird in Form eines Zustandsautomatens definiert (siehe Kapitel 6.4) und auf den elektronischen Zahlungsverkehr angewendet (siehe Kapitel 7). Für die Anwendung werden kryptographische Module benötigt. Wird das R&A-Modell mit Chipkarten realisiert, muß es in das Chipkarten-Betriebssystem integriert werden (siehe Kapitel 8.2).

	Sicherheits-niveau 1	**Sicherheits-niveau 2**	**Sicherheits-niveau 3**	**Sicherheits-niveau 4**
Krypto Module	Spezifikation der kryptographischen Module. Beschreibung der Module inklusive Hardware, Software, Firmware und der Sicherheitspolitik des Moduls			
Modul-Schnittstellen	Spezifikation aller erforderlichen und optionalen Schnittstellen und aller interner Datenpfade		Sicherheitskritische Datenein- und Ausgänge getrennt von den übrigen Datenein-/Ausgängen	
Rollen und Dienste	Logische Trennung von erforderlichen und option. Rollen und Diensten	Rollenbasierte Authentisierung	Identitätsbasierte Authentisierung	
Zustands-automat	Spezifikation des Zustandsautomaten. Erforderliche und optionale Zustände. Zustandsdiagramm und Spezifikation der Überführungsfunktionen			
Physische Sicherheit	Qualitäts-Produktions-ausstattung	Schlösser und Nachweisbarkeit eines Angriffs	Angriffserkennung und -vermeidung für Türen und Oberflächen	Angriffserkennung und -vermeidung für die gesamte Umgebung. Temperatur- und Spannungsschwankungsschutz
Software-sicherheit	Spezifikation des Software-Designs. Software muß sich auf Zustandsautomaten beruhen		Hochsprachen-Implementierung	Formales Modell. Vor- und Nachbedingungen

Fortsetzung ...

Betriebs-system Sicherheit	Ausführbarer Code. Authentisiert, Single User, Single Process	Kontrollierter Zugriffsschutz (C2 oder vergleichbar)	Gekennzeich-neter Schutz (B1 oder vergleichbar). Vertrauenswür-dige Kommuni-kationspfade	Strukturierter Schutz (B2 oder vergleichbar)
Schlüssel-management	FIPS-bewährte Erzeugungs- und Verteilungsmechanismen		Ein- und Ausgang von Schlüsseln in verschlüsselter Form	
Krypto Algorithmen	FIPS-bewährte kryptographische Algorithmen zum Schutz von Informationen			
EMV	FCC Teil 15, Subteil J, Klasse A		FCC Teil 15, Subteil J, Klasse B	
Selbsttests	Power-up Tests und Bedingungstests			

Tabelle 3-4: Sicherheitsanforderungen an kryptobasierte Systeme
Quelle: [FIPS 140-1], S. 20

Um ein hohes Sicherheitsniveau zu erreichen (Level 4), muß der Zustandsautomat mit allen seinen Zuständen vollständig spezifiziert sein. Ein detailliertes Zustandsdiagramm und die Spezifikation der Überführungsfunktionen muß vorliegen. Das Betriebssystem muß nach den Sicherheitskriterien der Stufe B2 der Trusted Computer Secuity Evaluation Criteria (TCSEC) evaluiert sein (siehe Kapitel 2.2.1). Niedrigere Sicherheitsniveaus reduzieren nicht die formale Spezifikation des Zustandsautomatens, sondern stellen geringere Anforderungen an die Evaluationsstufe des Betriebssystems. Für Sicherheitsniveau 3 ist die Evaluation nach B1 erforderlich und für Sicherheitsniveau 2 nach C2. Für Sicherheitsniveau 1 ist keine Evaluation des Betriebssystems erforderlich.

3.5.2 Ablaufsteuerung

Bevor ein Befehl, der vom Terminal an die Chipkarte gesendet wurde, ausgeführt wird, durchläuft er eine Reihe von Sicherheitsüberprüfungen. Die typische Befehlsabarbeitung innerhalb eines Chipkarten-Betriebssystems veranschaulicht Abbildung 3-11.

Erhält die Chipkarte über die *I/O-Schnittstelle* einen Befehl, so sichert der *I/O-Manager* als erstes die fehlerfreie Übertragung. Sind die Daten verschlüsselt, entschlüsselt der *Secure Messaging Manager* die mit dem Befehl übermittelten Daten und leitet den Befehl mit den zugehörigen Daten an den *Befehlsinterpreter* weiter.

Dieser decodiert den Befehl und übergibt ihn an den *Logical Channel Manager*, der den ausgewählten Kanal ermittelt, die aktuellen Zustände ermittelt und den entsprechenden *Zustandsautomaten* aufruft. Dieser prüft, ob der Befehl mit den gesetzten Parametern im aktuellen Zustand überhaupt erlaubt ist. Falls dies zutrifft, wird der Befehl ausgeführt, das heißt mit der *Dateiverwaltung* und der *Speicherverwaltung* wird auf die entsprechende Datei im *EEPROM* zugegriffen.

Wurde der Befehl erfolgreich ausgeführt, wird über den *Returncode Manager* der entsprechende Returncode über den *Secure Messaging Manager* und den *I/O-Manager* an die *I/O-Schnittstelle* übergeben.

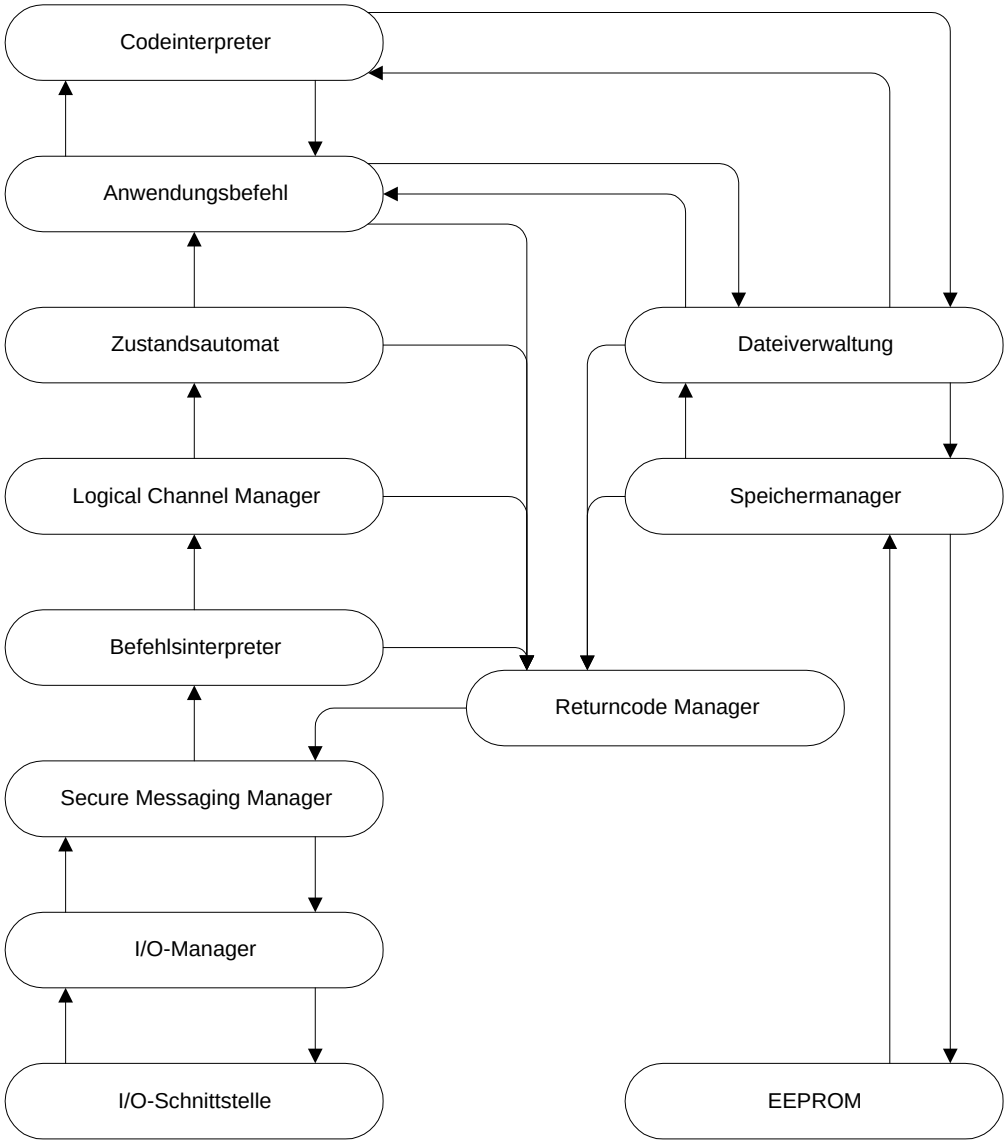


Abbildung 3-11: Befehlsabarbeitung innerhalb eines Chipkarten-Betriebssystems
Quelle: [Rankl, Effing 1996], S. 124

3.5.3 Speicherorganisation

Die Speicherorganisation gestaltet sich wie im folgenden beschrieben [Rankl, Effing 1996]. Ein Chipkarten-Microcontroller besitzt in der Regel drei unterschiedliche Speicherarten: ROM, RAM und EEPROM. Das ROM kann nur in seiner Gesamtheit während der Herstellung des Microcontrollers programmiert werden. Es kann während der ganzen Lebensdauer der Chipkarte nicht mehr verändert werden. Im ROM werden die Basismodule des Betriebssystems gespeichert.

Das RAM kann beliebig oft mit sehr hoher Geschwindigkeit beschrieben und auch wieder gelöscht werden. Jedoch bleibt der Inhalt im RAM nur solange erhalten, wie Spannung an der Chipkarte liegt. Durch einen Spannungsausfall werden alle Daten im RAM gelöscht.

Das RAM gliedert sich in die Bereiche Register, Stack, allgemeine Variablen, Arbeitsbereiche für kryptografische Algorithmen und dem I/O-Puffer. Genügt der Speicherplatz der einzelnen Bereiche im RAM nicht aus, wenn beispielsweise ein größerer I/O-Puffer gefordert wird oder zusätzliche Variablen im RAM gespeichert werden müssen, so werden die betroffenen RAM-Inhalte ins EEPROM ausgelagert. Dies ist trotz der begrenzten Lebensdauer des EEPROMs und der erhöhten Schreibzeiten oft die einzige Möglichkeit, dieses Speicherplatzproblem zu lösen.

Der Programmcode im ROM bedarf bis auf die Interruptvektoren keiner höheren Struktur. Die einzelnen Programmteile können in beliebiger Reihenfolge gespeichert werden, wobei jedoch die Entfernungen der Sprungadressen minimiert werden sollten.

Das EEPROM ist in Seiten aufgeteilt. Im EEPROM gespeicherte Daten bleiben auch ohne Versorgungsspannung erhalten. Dabei treten jedoch folgende Nachteile auf. Das EEPROM hat eine begrenzte Lebensdauer. Es kann nicht unendlich oft beschrieben und gelöscht werden. Hinzu kommt, daß Schreib- und Löschzugriffe sehr lange dauern (ca. 1ms pro Byte).

Das EEPROM ist gegenüber den anderen beiden Speicherarten wesentlich komplizierter strukturiert. Die Fertigungsdaten stehen in einem besonders geschützten Bereich am Anfang des EEPROM und beinhalten beispielsweise eine Identifikationsnummer. Dieser Bereich kann nur einmal beschrieben und anschließend nur noch gelesen werden. Die Größe dieses Bereichs beträgt typischerweise 32 Byte.

Danach folgen die Tabellen und Zeiger für das Betriebssystem. Dieser Bereich ergibt mit den Basisprogrammodulen im ROM erst das eigentliche Betriebssystem. Er wird durch eine Prüfsumme (EDC), die vor jedem Zugriff nachgerechnet wird, abgesichert. Der anschließende Bereich beinhaltet zusätzlichen Programmcode für die Anwendungen. Es können anwendungsspezifische Befehle oder Algorithmen geladen werden. Auch hier ist es möglich, die Daten mit einer Prüfsumme zu schützen. Der Dateibereich umfaßt alle Dateistrukturen, also den gesamten nach außen sichtbaren Dateibaum. Am Ende des EEPROM existiert ein Freispeicher, dessen Verwaltung ein eigener Manager durchführt. Oft ist dieser Freispeicher einzelnen Anwendungen im Dateibereich zugeordnet.

Das Betriebssystem prüft zur Laufzeit die Einhaltung der Speichergrenzen einer Anwendung. Deshalb müssen alle Speicherbereiche der betroffenen Anwendung in einem physikalisch zusammenhängenden Speicherbereich abgelegt sein. Wenn eine Anwendung Dateien anlegen und löschen möchte, muß sie einen eigenen Freispeicherbereich besitzen, der bei der Neuanlage von Dateien verwendet wird. Nach Löschung einer Datei fällt der frei werdende Platz wieder dem Freispeicher zu. Bei dieser Methode kann nur der Speicherplatz der jeweils letzten Datei im Dateibereich beim Löschen dem

Freispeicher zugeordnet werden. Dies stellt eine starke Einschränkung für Anwendungen dar.

3.5.4 Dateistrukturen

Die neueren Betriebssysteme für Chipkarten verfügen über ein vollständiges und hierarchisch organisiertes Dateiverwaltungssystem mit symbolischer und hardwareunabhängiger Adressierung. Die Dateien sind objektorientiert aufgebaut, so daß alle Informationen über eine Datei in dieser Datei selbst abgespeichert sind. Eine Datei besteht immer aus einem Kopf, in dem Informationen über den Namen, den Typ, die Struktur, die Größe, Zugriffsbedingungen und Attribute der Datei sowie eine Referenz zum Dateibaum enthalten sind, und einem Körper, auf den vom Kopf verwiesen wird. Dieser speichert die eigentlichen Daten. Die Aufteilung in Kopf und Körper erhöht die physikalische Sicherheit der Datenbestände. Da das EEPROM in Seiten aufgeteilt ist, befinden sich Kopf und Körper stets auf getrennten Speicherseiten.

Der Aufbau von Chipkarten Dateisystemen ist in der ISO-Norm 7816-4 festgelegt. Es existieren zwei unterschiedliche Dateikategorien; Dedicated Files (DF) und Elementary Files (EF). *Dedicated Files* und *Elementary Files* sind hierarchisch strukturiert. Vergleicht man das Chipkarten Dateisystem mit dem von DOS oder Unix, so entsprechen *Dedicated Files* den Verzeichnissen und die *Elementary Files* den Dateien.

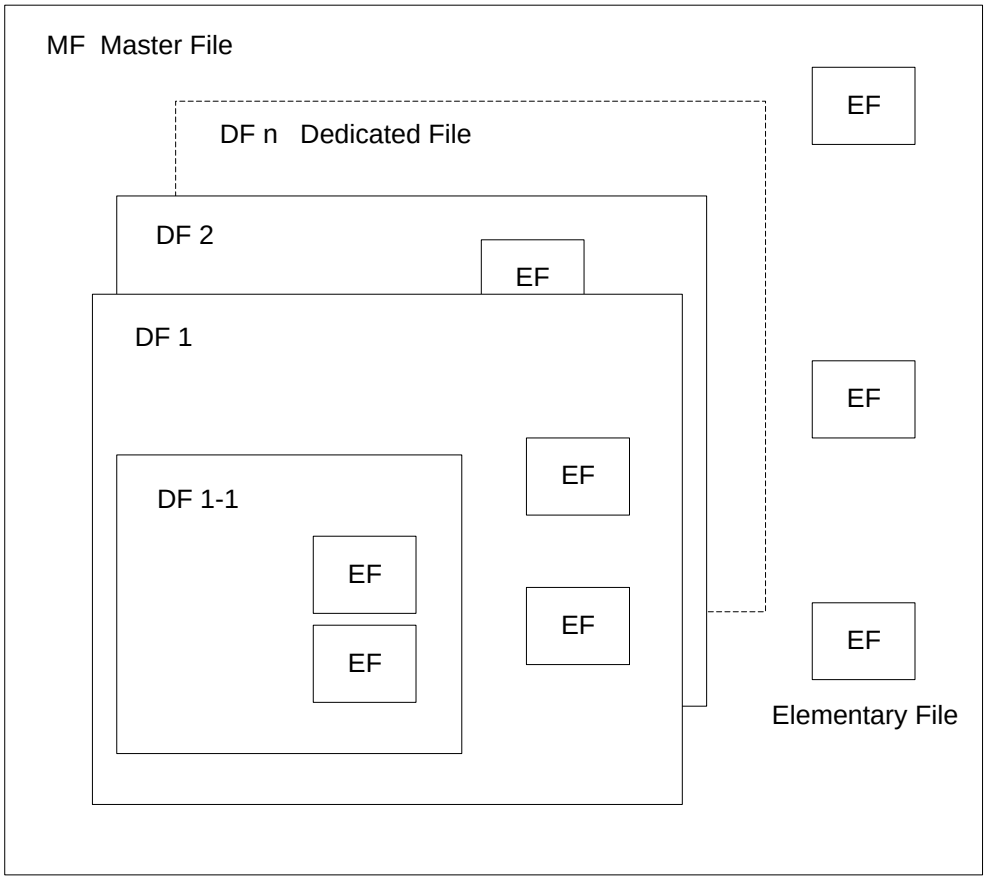


Abbildung 3-12: Dateiorganisation

Das DF, das dem Wurzelverzeichnis entspricht, wird *Master File* (MF) genannt. Innerhalb jedes DFs können sich weitere DFs befinden. Während das MF immer vorhanden ist, ist die Existenz weiterer DFs optional. Die Schachtelungstiefe der DFs wird lediglich durch den Speicherplatz begrenzt. In der Praxis werden die Daten einer Anwendung in einem separaten DF abgespeichert. EFs können direkt unter dem MF oder innerhalb eines DFs angeordnet sein.

Die EFs unterteilen sich in zwei Typen. In den *internen Systemdateien* (*Internal Secret File* - ISF) werden Daten für das Betriebssystem, für den Ablauf einer Anwendung oder geheime Schlüssel aufbewahrt. Der Zugriff auf diese Dateien ist vom Betriebssystem besonders geschützt. Sie sind im jeweiligen Anwendungs-DF versteckt angeordnet und können nicht selektiert werden. Die Verwaltung übernimmt völlig transparent das Betriebssystem. Die *Arbeitsdateien* (*Working Elementary Files* - WEF) beschreiben den zweiten Typ der EFs. In ihnen werden die eigentlichen Nutzdaten einer Anwendung gespeichert. Die Abbildung 3-12 gibt einen Überblick über die Anordnung von DFs und EFs.

Die Dateien in Chipkarten-Betriebssystemen werden ausschließlich logisch adressiert. Alle Dateien, einschließlich der Verzeichnisse, besitzen einen zwei Byte langen *File Identifier* (FID), unter dessen Verwendung sie ausgewählt werden. Die niederwertigen fünf Bit des FID werden auch noch als Short-FID bezeichnet. Dieser wird zur impliziten Selektion von Dateien verwendet.

Das MF hat den FID „3F00h“. Der logische Dateiname „FFFFh“ ist für zukünftige Anwendungen reserviert und darf nicht verwendet werden. Ansonsten kann der FID grundsätzlich frei gewählt werden. Dabei muß sichergestellt sein, daß die Dateien eindeutig selektiert werden können. Dabei gelten folgende Regeln:

- EFs innerhalb eines DFs dürfen nicht den gleichen FID haben
- Verschachtelte DFs dürfen nicht den gleichen FID haben
- EFs innerhalb eines DFs dürfen nicht den gleichen FID wie das über- oder untergeordnete DF haben

Die DFs stellen die Zusammenfassung der EFs für die einzelnen Anwendungen dar. Bei einer Vielzahl von verschachtelten DFs kann der zur Verfügung stehende Adressraum mit den zwei Byte langen FID zu klein werden. Deshalb wird für die DF noch ein *Application Identifier* (AID) vergeben. Dieser hat eine Länge von fünf bis 16 Byte und setzt sich aus folgenden Datenelementen zusammen:

- *Registered Identifier* (RID): Dieser hat eine Länge von fünf Byte und wird von einer nationalen oder internationalen Registrierungsstelle vergeben. Er beinhaltet einen Ländercode, eine Anwendungskategorie und eine Nummer für den Anwendungsanbieter. Dieser Zahlencode ist weltweit eindeutig und kann zur Identifizierung einer Anwendung verwendet werden.

- *Proprietary Application Identifier Extension* (PIX): Der Anwendungsanbieter hat die Möglichkeit, diese der RID nachzustellen. Sie ist bis zu elf Byte lang und kann beispielsweise eine Serien- oder Versionsnummer enthalten.

Bevor auf ein EF zugegriffen werden kann, muß es selektiert werden. Es kann immer nur ein EF gleichzeitig selektiert werden. Mit einer erfolgreichen Selektierung eines neuen EF wird die bisherige Selektion ungültig. Es gibt unterschiedliche Möglichkeiten der Selektion:

- **Selektion von MF und DFs:** Das MF kann von jedem Ort im Dateibaum selektiert werden, da es einen eindeutigen Namen besitzt (‘3F00’). Die DF können mit ihrem FID oder dem registrierten und somit einzigartigen AID ausgewählt werden.
- **Explizite Selektion von EFs:** Hierbei wird vor dem eigentlichen Zugriff ein eigener Befehl mit dem FID als Parameter zur Auswahl der gewünschten Datei zur Chipkarte gesendet. Anschließend kann auf die Datei zugegriffen werden.
- **Implizite Selektion von EFs:** Hierbei erfolgt der eigentliche Zugriff ohne vorherige explizite Selektion der entsprechenden Datei. Der Short-FID wird als Parameter des entsprechenden Befehls an die Chipkarte übergeben. Die implizite Selektion funktioniert nur bei EFs innerhalb des aktuellen DFs oder MF. Über Verzeichnisgrenzen hinweg ist keine Auswahl der Dateien möglich. Weiterhin kann sie nur mit Befehlen verwendet werden, bei denen als Parameter der Short-FID übergeben werden kann.

In Abhängigkeit vom Verwendungszweck können bei EFs unterschiedliche Datenstrukturen verwendet werden (Abbildung 3-13).

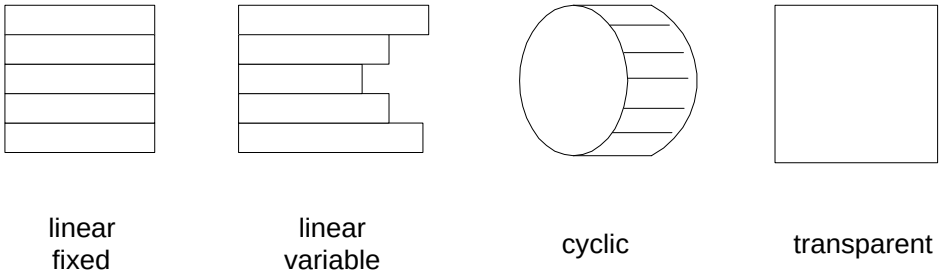


Abbildung 3-13: Datenstrukturen von Elementary Files
Quelle: [Weikmann 1997]

3.5.5 Datenübertragung zur Chipkarte

Der Informationsaustausch zwischen einem Endgerät (Interface Device - IFD) und einer Chipkarte (Integrated Circuit Card - ICC) erfolgt mit Hilfe von Transportprotokollen in Befehl-Antwort-Paaren. Dabei unterscheidet man zwischen synchroner Datenübertragung bei Speicherkarten und asynchroner Datenübertragung bei Mikroprozessorkarten. Grundsätzlich wird die Kommunikation mit der Chipkarte vom Terminal

angestoßen, die Chipkarte reagiert damit nur auf die Befehle des Terminals. Daraus ergibt sich ein Master-Slave-Verhalten, mit dem Terminal als Master und der Chipkarte als Slave.

Nachdem die Chipkarte in Kontakt mit dem Terminal steht, führt sie automatisch einen Power-On-Reset aus und sendet ein Answer to Reset (ATR) zum Terminal. Dieses wertet den ATR aus, der verschiedene Kartenparameter anzeigt, und schickt anschliessend den ersten Befehl. Die Chipkarte bearbeitet diesen, erzeugt eine Antwort und schickt diese anschließend zurück zum Terminal. Diese Folge von Befehl empfangen und Antwort zurücksenden setzt sich bis zum Deaktivieren (Entfernen aus dem Terminal) der Chipkarte fort.

Das Terminal hat die Möglichkeit, nach dem ATR an die Chipkarte noch ein Protocol Type Select (PTS) zu senden. Dieser Befehl erlaubt dem Terminal verschiedene Übertragungsparameter des Protokolls der Chipkarte einzustellen [Rankl, Effing 1996].

Analog zum Schichtenmodell der OSI läßt sich die gesamte Datenübertragung von und zur Chipkarte in drei Protokollschichten einteilen [Rankl, Effing 1996]:

- Physikalische Schicht (Schicht 1)
- Leitungsschicht (Schicht 2)
- Anwendungsschicht (Schicht 7)

3.5.5.1 Physikalische Schicht

Die physikalische Übertragungsschicht ist in der internationalen Norm für Chipkarten, der ISO-Norm 7816-3, festgelegt. Der gesamte Datenaustausch zwischen Terminal und Chipkarte findet auf digitalem Wege in serieller Form statt. Weiterhin läuft sie asynchron ab, was bedeutet, daß jedes übertragene Byte mit zusätzlichen Synchronisationsbits ausgestattet werden muß. Vor einem übertragenden Byte wird ein Startbit gesendet. Am Ende jeden Bytes werden sowohl ein Paritätsbit als auch zwei Stopbits hinzugefügt.

Chipkarten-Microcontroller verfügen nur über taktabhängige Timer. Deshalb wird die Dauer eines Datenbits nicht in absoluten Zeitwerten angegeben, sondern in Abhängigkeit vom Takt festgelegt. Dazu ist ein Teiler definiert, der die Anzahl der Takte pro Bit angibt. Die Zeitdauer für ein Bit wird als eine ETU (elementary time unit) bezeichnet [Rankl, Effing 1996].

3.5.5.2 Leitungsschicht

Für die Kommunikation zwischen Chipkarte und Terminal sind insgesamt 15 Übertragungsprotokolle vorgesehen und in ihrer Grundfunktionalität definiert. Diese gehören im OSI-Modell in die Schicht 2 (Leitungsschicht). International haben sich zwei Übertragungsprotokolle durchgesetzt. Zum einen ist dies das T=0 Protokoll, das 1989 in der internationalen ISO-Norm 7816-3 genormt wurde, und zum anderen das T=1 Protokoll, das 1992 im Anhang zu dieser Norm definiert wurde. Die durch die Übertra-

gungsprotokolle transportierten Dateneinheiten bezeichnet man als TPDU (Transmission Protocol Data Unit). Sie unterscheiden sich je nach Protokoll und nehmen die Aufgabe wahr, Daten von und zur Chipkarte zu transportieren [Rankl, Effing 1996].

Übertragungsprotokoll T=0

Das T=0 Protokoll arbeitet byteorientiert. Die kleinste Einheit, die übertragen werden kann, ist ein einzelnes Byte. Die TPDU beim T=0 Protokoll setzt sich aus einem Datenkopf mit Class-, Instruction- und drei Parameterbytes sowie einem Datenkörper zusammen (Abbildung 3-14). Der dritte Parameter (P3) im Kopf gibt entweder die Länge der Daten des Befehls oder die Länge der Antwort an.

Datenkopf					Datenkörper
CLA	INS	P1	P2	P3	Daten-Feld

Abbildung 3-14: Aufbau eines Befehls bei T=0

Quelle: [Rankl, Effing 1996], S. 169

Durch die Byteorientierung muß direkt nach einem erkannten Übertragungsfehler das nicht korrekt empfangene Byte nochmals angefordert werden. Jedem Byte folgt ein Paritätsbit, an dem zum Beispiel ein Übertragungsfehler erkannt wird. Nachdem der Empfänger einen Übertragungsfehler erkannt hat, setzt er die I/O-Leitung für die Dauer einer ETU auf den Pegel „low“. Damit signalisiert er dem Kommunikationspartner, daß dieser das zuletzt gesendete Byte wiederholen muß.

Bei der Kommunikation mit der Chipkarte sendet das Terminal zuerst den Header der TPDU. Erst wenn die Chipkarte mit einem Acknowledge (ACK) dem Terminal signalisiert, daß der Header korrekt empfangen wurde, sendet das Terminal den Datenteil. Nach Ausführung des Befehls überträgt die Chipkarte die Antwort mit dem entsprechenden Returncode. Falls in der Antwort auch Daten enthalten sind, wird dies dem Terminal durch den Returncode mitgeteilt. Dieses fordert daraufhin mit einem Befehl die Daten von der Chipkarte an.

Übertragungsprotokoll T=1

Das Übertragungsprotokoll T=1 ist ein asynchrones Halbduplexprotokoll für Chipkarten. Es gehört zur Klasse der Blockprotokolle, das heißt, daß ein Block die kleinste Dateneinheit ist, die zwischen Chipkarte und Terminal übertragen werden kann.

Prologfeld			Informationsfeld	Epilogfeld
Knotenadresse NAD	Protokollkontrollbyte PCB	Länge LEN	Dateneinheit APDU	Fehler-erkennung EDC
1 Byte	1 Byte	1 Byte	0-254 Bytes	1-2 Byte

Abbildung 3-15: Aufbau eines Übertragungsblocks bei T=1

Quelle: [Rankl, Effing 1996], S. 175

Die zu übertragenden Blöcke werden einerseits zur transparenten Übertragung von anwendungsspezifischen Daten und andererseits für die Steuerdaten des Übertragungsprotokolls beziehungsweise für die Behandlung von Übertragungsfehlern verwendet. Die Übertragungsblöcke setzen sich aus einem Prologfeld, dem Informationsfeld und einem Epilogfeld zusammen. Während Prolog- und Epilogfeld vorhanden sein muß, ist das Informationsfeld optional.

Im T=1 Protokoll unterscheidet man drei verschiedene Blocktypen:

- Die Informationsblöcke (I-Blöcke) werden zum transparenten Austausch von Daten der Anwendungsschicht genutzt.
- Der Empfangsbestätigungsblock (R-Block), der kein Informationsfeld besitzt, dient für positive oder negative Empfangsbestätigungen.
- Systemblöcke (S-Block) nutzt man für Steuerinformationen, die das Protokoll selbst betreffen. In Abhängigkeit von der Steuerinformation können sie ein Informationsfeld besitzen.

Das Prologfeld besteht aus folgenden Unterfeldern:

- **Knotenadresse (NAD):** Dieses Feld enthält die Ziel- und Quellknotenadresse des Blocks.
- **PCB-Feld:** Dies ist das Protokollkontrollbyte. Es dient zur Kontrolle und Steuerung des Übertragungsprotokolls. Es enthält primär den Blocktyp sowie dazu notwendige Zusatzinformationen.
- **LEN-Feld:** Dieses Feld gibt die Länge des Informationsfeldes an.

Das Informationsfeld enthält im Falle eines I-Blocks die Daten für die Anwendungsschicht, also entweder Befehls- oder Antwortdaten. Im Falle eines S-Blockes werden in diesem Feld die Daten für das Übertragungsprotokoll übertragen. Das Epilogfeld enthält einen Fehlererkennungscode über alle vorangegangenen Bytes des Blocks.

Beim T=1 Protokoll sind Kommunikationsablauf und Fehlerbehandlung erheblich komplexer als beim T=0 Protokoll, so daß von deren Erläuterung hier abgesehen wird.

3.5.6 Das Chipkarten-Betriebssystem STARCOS

Giesecke & Devrient begann 1990 mit der Entwicklung von STARCOS (Smart Card Chip Operating System), einem vollständigen Betriebssystem für Chipkarten [STARCOS 1997]. Die Entwicklung orientierte sich an der ISO-Norm 7816.

3.5.6.1 Datenstrukturen

STARCOS unterstützt, wie in ISO-Norm 7816-4 definiert, ein hierarchisches Datei-System bestehend aus *Master File* (MF), *Dedicated Files* (DF) und *Elementary Files*. Letztere untergliedern sich in die eigentlichen *Elementary Files* (EF) und die *Internal Secret Files* (ISF) (siehe Kapitel 3.5.4). Im Unterschied zur Norm existiert unter der Ursprungsebene (MF) nur eine Unterebene (DF).

STARCOS bietet neben den in der ISO-Norm 7816-4 beschriebenen Datenstrukturen *linear fixed*, *cyclic* und *transparent* folgende weitere Strukturen an:

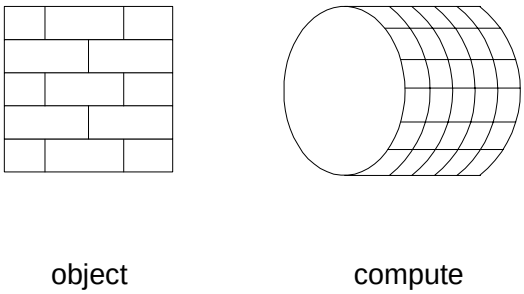


Abbildung 3-16: Datenstrukturen object und compute
Quelle: [STARCOS 1997], S. 11

Die Datenstruktur *object* basiert auf der Datenstruktur *transparent*. Die Daten werden als TLV-Objekte im ASN.1-Format [ISO 8825] gespeichert und werden vom Betriebssystem interpretiert. Die Datenstruktur *compute* beruht auf der Datenstruktur *cyclic*.

3.5.6.2 Attribute

Für jede Datei werden beim Anlegen verschiedene Attribute definiert. STARCOS unterstützt folgende Attribute:

Application Identifier / File Identifier: Jede Datei ist über den *File Identifier* (FID) adressierbar. Jedes DF hat zusätzlich einen *Application Identifier* (AID) (siehe Kapitel 3.5.4). STARCOS verlangt, daß die DF-FID ungleich den EF-FIDs im MF sind. Ferner müssen alle DFs unterschiedliche AIDs und FIDs haben.

Short Identifier: Der *Short Identifier* kann zur direkten Adressierung eines EFs verwendet werden. Er hat einen Wertebereich zwischen 0 und 31 und kann unabhängig vom FID gewählt werden.

Access Conditions: Die *Access Conditions* (AC) legen fest, in welchem Zustand auf eine Datei mit einem bestimmten Befehl zugegriffen werden kann. ACs sind feste Bestandteile des EF-Header, der Preheader von EF, ISF und DF sowie von Key Records.

Operation Mode: Der *Operation Mode* (OM) einer Datei legt Zugriffsattribute fest, die höhere Priorität haben als die ACs. Es gibt folgende Kategorien, die durch Bits oder Bitkombinationen den Zugriff bestimmen:

Attribut für EFs:

- EF locked
  Dieses Bit bestimmt, ob der Zugriff auf die Datei erlaubt ist.

Attribute für ISFs:

- Write (WR)
  Dieses Bit legt fest, ob ein existierender Schlüssel mit dem Befehl WRITE KEY überschrieben werden darf.

- Write Once (WO)
  Dieses Bit legt fest, daß der Key Record (Datensatz im ISF) nur einmal mit WRITE KEY überschrieben werden darf.

- Virgin
  Dieses Bit zeigt an, ob der Key Record schon einmal beschrieben wurde.

3.5.6.3 Befehlsablaufschema

Das Kommunikationsverhalten zwischen Terminal und Chipkarte ist ein reines Master-Slave-Verhalten, mit dem Terminal als Master und der Chipkarte als Slave (siehe Kapitel 3.5.5). Die Chipkarte reagiert nur auf Befehle des Terminals. Das grundsätzliche Ablaufschema vom Empfangen eines Befehls bis zum Senden der Antwort wird in Abbildung 3-17 skizziert.

Empfängt die Chipkarte einen Befehl vom Terminal, so überwacht als erstes der *Transmission Manager* die fehlerfreie Übertragung. Soll die Übertragung verschlüsselt ablaufen, sichert die *Secure-Messaging*-Ebene anschließend die Datenübertragung durch kryptographische Algorithmen.

Beispielsweise entschlüsselt *Secure Messaging* die mit dem Befehl übermittelten und nach dem DES verschlüsselten Daten und leitet den Befehl mit den dazugehörigen Daten an den *Befehlsinterpreter* weiter. Dieser erkennt den Befehl, überprüft die mitgesandten Parameter und führt den Befehl aus. Erfolgt hierbei ein Dateizugriff, beispielsweise das Schreiben in ein EF, kontrolliert der *File Manager* die Einhaltung der erforderlichen Berechtigungen. Dabei steuert die objektorientierte Sicherheitsstruktur durch Definition von Ausgangs- und Folgezuständen den Ablauf einer Anwendung.

Als nächstes sind zusätzlich dazu die Zugriffsbedingungen für die einzelnen Dateien in ihrem Header hinterlegt. Nach Abarbeitung des Befehls wird eine Antwort generiert, deren Inhalt noch verschlüsselt werden kann. Zuletzt überwacht der *Transmission Manager* die fehlerfreie Datenübertragung zum Terminal. Die Verarbeitung eines Befehls muß vollständig und fehlerfrei ausgeführt werden, bei auftretenden Fehlern erfolgt eine entsprechende Meldung.

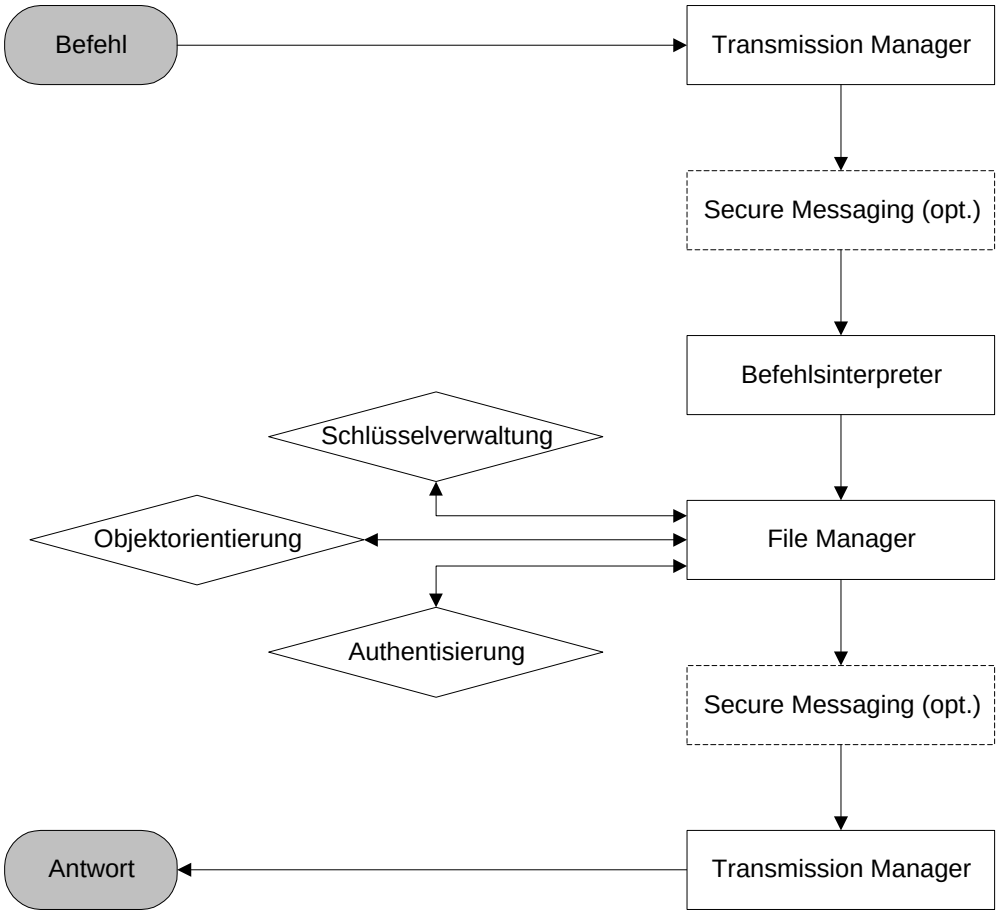


Abbildung 3-17: Befehlsablaufschema
Quelle: [STRACOS 1997], S. 21

3.5.7 Das Chipkarten-Betriebssystem der Java-Card

Eine Java-Card ist eine Chipkarte, die Programme in der Sprache Java ablaufen lassen kann. Java-Programme werden Applets genannt, die plattformunabhängig interpretiert werden. Java ist eine objektorientierte Programmiersprache. Daten werden in Objekten abgelegt und zu Klassen zusammengefaßt. Java unterstützt das Vererbungsprinzip und alle wesentlichen Eigenschaften der objektorientierten Programmierung.

Die Programmiersprache Java gibt es in reduziertem Sprachumfang auch für Chipkarten (Java-Cards). Diese Sprachvariante wurde im Frühjahr 1997 von der Firma Sun und nahezu allen großen Chipkartenherstellern entwickelt. Dabei entstand das international tätige Standardisierungsgremium Java-Card Forum (JCF). Dieses Gremium hat die Aufgabe, die Sprachspezifikation für die Java-Card zu definieren, den Rahmen für den

Java-Card Interpreter (JCVM) und allgemeine und anwendungsspezifische APIs als Schnittstelle zwischen dem Chipkarten-Betriebssystem und der Java-Umgebung auf der Java-Card festzulegen.

Die zur Zeit aktuellen Spezifikationen haben die Bezeichnung Java-Card Version 2.1, Application Programming Interface, Language Subset and Virtual Machine Specification and Programming Concepts [Sun 1999].

3.5.7.1 Systemarchitektur

Eine Java-Card besitzt nach der Java-Card Spezifikation eine Java-Card Virtual Machine, die in der Kartenfertigung aktiviert und am Ende des Kartenlebenszyklus deaktiviert wird [Rankl, Effing 1999]. Die Systemarchitektur ist in Abbildung 3-18 dargestellt.

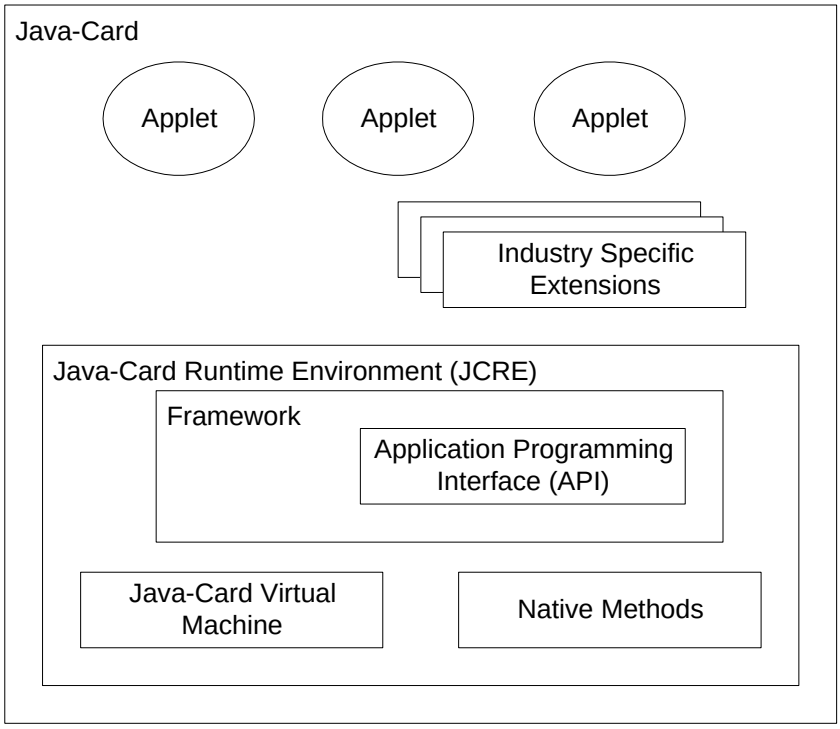


Abbildung 3-18: Systemarchitektur der Java-Card

Quelle: [Sun 1998b], S. 1-4

Die *Java-Card Virtual Machine* (JCVM) hat zwei grundsätzliche Aufgaben. Zum einen definiert sie den abstrakten Rechner, der Java-Bytecode ausführt, zum anderen unterstützt sie die Applet-Erstellung. *Native Methods* sind Funktionen, die die Abwicklung der I/O-Befehle, die Ausführung kryptographischer Algorithmen und die Speicherallozierung unterstützen. Das Java-Card *Framework* dient dazu, Chipkarten-Applikationen auf einfache Weise zu erstellen, wobei keine Kenntnisse über Systemdetails der Chipkarten notwendig sind. Es versorgt den Anwendungsentwickler mit einer relativ einfachen Programmierschnittstelle. Dieser kann anhand der vorgegebenen *Application Programming Interface* (API) seine eigenen Applets erstellen, diese sind die eigentlichen Applikationen der Chipkarte. Industriespezifische Erweiterungen

(*Industry Specific Extensions*) enthalten zusätzliche Klassen für die Erweiterung der auf der Chipkarte installierten Applets. Das *Java-Card Runtime Environment* (JCRE) beinhaltet die *Java-Card Virtual Machine*, das *Framework*, die *Native Methods* und die *Application Programming Interfaces*.

Ein Dateisystem nach ISO-Norm 7816-4 ist nicht vorgesehen, da sich dieses auch mit Objekten innerhalb eines Java Applets aufbauen läßt [Rankl, Effing 1999]. Dazu existieren einige Klassen, die den Aufbau eines ISO-Norm 7816-4 konformen Dateisystems relativ einfach ermöglichen. Der Programmcode sowie der dazugehörige Dateibaum sind Teil eines Applets, das in die Chipkarte geladen wird. Das Applet kann dort mit einer eindeutigen AID und dem Select-Befehl ausgewählt werden. Nach der Selektion des Applets erhält es automatisch alle Befehle zur Abarbeitung. Der Programmcode des Applets kann dann die Befehle und deren Daten auswerten und bearbeiten sowie die entsprechenden Zugriffe auf das Dateisystem durchführen.

Dieses Prinzip schafft große Flexibilität und Kompatibilität, da sich Anwendungen inklusive Dateibaum innerhalb eines Applets befinden. Die eigentlichen Kartenbefehle, wie Read Binary oder Mutual Authenticate, sind als Programmcode innerhalb der Applets enthalten. So ist es dadurch beispielsweise möglich, gleiche Befehle mit unterschiedlicher Codierung und differierendem Programmablauf, innerhalb einer Chipkarte in zwei getrennten Applets unabhängig voneinander zu unterstützen.

Dieser Vorteil kostet jedoch erheblichen Speicherplatz, da die einzelnen Applets zwangsläufig redundante Daten und Programmcode enthalten. Aus diesem Grund ist es möglich, einige Datenobjekte eines Applets mit anderen Applets zu teilen (siehe Kapitel 3.5.7.3). Aus Sicherheitsgründen läßt sich dies nur von dem Applet aus durchführen, das das Objekt auch erzeugt hat. Der Vorgang selber kann nicht mehr rückgängig gemacht werden. Das heißt, ein Objekt, auf das der Zugriff freigegeben wurde, bleibt bis zum Ende des Kartenlebenszyklus freigegeben.

Die einzigen appletunabhängigen Befehle dienen zum Laden von Applets in die Chipkarte. Diese werden dann im EEPROM abgelegt und von der Java-Card Virtual Machine (JCVM) ausgeführt.

3.5.7.2 Java-Card Klassen

Die Kernklassen der Java-Card sind wesentlich kompakter als die der Java Plattform. Kernklassen (packages) stellen Klassen zur Verfügung, die in dem objektorientierten Konzept von Java Anwendung finden [Sun 1997]. Das Java-Card Framework beinhaltet vier Kernklassen (Tabelle 3-5).

In der Kernklasse javacardx.framework gibt es die Klasse FileSystem, mit der auf objektorientierter Basis ein Dateisystem erzeugt werden kann, das kompatibel zu der ISO-Norm 7816-4 ist. Java-Card benutzt Objekte, um Daten zu repräsentieren, zu speichern oder zu verändern. In dem von der Klasse FileSystem zur Verfügung gestellten Dateisystem sind Elementary Files (EF) und Dedicated Files (DF) vorgesehen.

Folgende Operationen können mit den Dateien vorgenommen werden [Sun 1998b]:

- Erzeugen einer neuen Datei
- Hinzufügen einer neuen Datei zu einem Dedicated File (DF)
- Setzen von Sicherheitsattributen für eine Datei
- Selektion einer Datei

Name	Beschreibung
javacard.framework	Dies ist die Basis-Kernklasse der Chipkarte. Sie definiert Klassen zur elementaren Konstruktion von Java-Card Programmen und stellt Systemroutinen zur Verfügung.
javacardx.framework	Dieses Package stellt ein objektorientiertes Design für ein ISO-Norm 7816-4 kompatibles Dateisystem zur Verfügung. Es unterstützt Elementary Files (EF), Dedicated Files (DF) und dateiorientierte APDU.
javacardx.crypto und javacardx.cryptoEnc	Diese beiden Packages unterstützen kryptographische Funktionalitäten, die in der Chipkarte notwendig sind.

Tabelle 3-5: Kernklassen der Java-Card
Quelle:[Sun 1997]

Die Attribute einer Datei ermöglichen deren Schutz vor unbefugtem Lese- oder Schreibzugriff. Jede Datei besitzt zwei Attribute, eines für Lesezugriff und eines für Schreibzugriff. Beide können unabhängig voneinander gesetzt werden. Für jedes Attribut können folgende vier Werte gesetzt werden:

- ALLOW_ANY (Lesen / Schreiben ist erlaubt)
- ALLOW_NONE (Lesen / Schreiben ist untersagt)
- ALLOW_AUTH1 (Lesen / Schreiben ist nur erlaubt, wenn AUTH1 wahr ist)
- ALLOW_AUTH2 (Lesen / Schreiben ist nur erlaubt, wenn AUTH2 wahr ist)

AUTH1 und AUTH2 werden nur nach erfolgreicher Authentisierung wahr.

3.5.7.3 Sicherheitskonzept

Im Vergleich zu Standard Java unterstützt die Sprachvariante für die Java-Card, neben den wesentlichen Sicherheitseigenschaften, wie der Byte Code Überprüfung (Byte Code Verifier) und dem Sandbox-Sicherheitsmodell, auch die gemeinsame Benutzung ein und desselben Objektes von mehreren unterschiedlichen Applets (Object Sharing).

Das Sandbox-Sicherheitsmodell stammt aus der Spezifikation von Standard Java und definiert die prinzipielle Beziehung zwischen einem Applet und den Systemdiensten. Im

Sandbox-Modell wird jedoch keine Aussage über Beziehungen zwischen Applets getroffen. Dies kann jedoch für multifunktionale Chipkarten von großer Bedeutung sein. Aus diesem Grund wurde eine Ergänzung des Sandbox-Sicherheitsmodell definiert, das sogenannte Gateway-Modell. Das Gateway-Modell basiert auf einer Definition des Capability-Modells, das im Rahmen des Java Electronic Commerce Framework (JECF) für Standard Java definiert wurde [Sun 1998a].

Das Gateway-Modell benutzt die Sicherheitsmechanismen von Java und die Infrastruktur des Sandbox-Sicherheitsmodells. Es stellt für Entwickler die Möglichkeit zur Verfügung, Vertrauensbeziehungen zwischen den einzelnen Applets herzustellen, um gemeinsame Objekte zu benutzen.

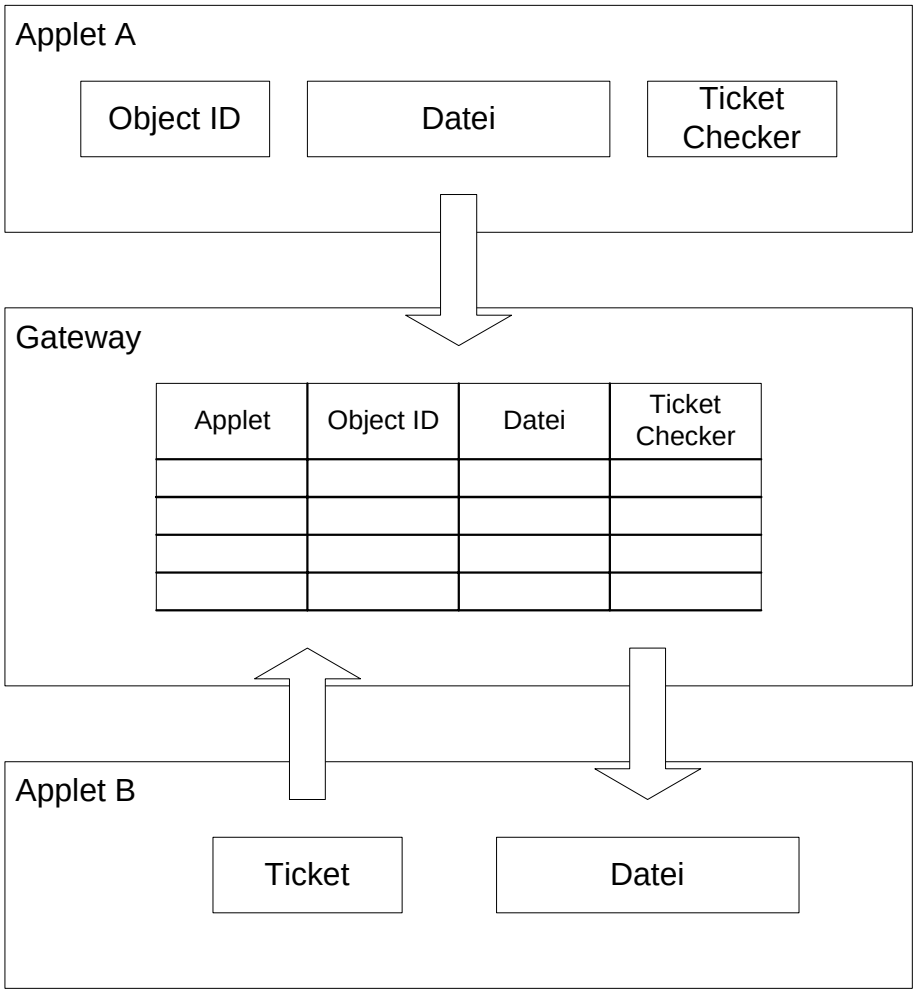


Abbildung 3-19: Gateway-Modell
Quelle: [Java Card Forum 1997], S. 14

Die einzelnen Applets sind, nach dem Sandbox-Prinzip, voneinander durch eine Art „Firewall“ getrennt. Ein Applet muß während des Installationsprozesses explizit eine Gateway-Funktion aufrufen, um mit anderen Applets Objekte gemeinsam nutzen zu können. Nur während der Installationsphase kann das Objekt für andere Applets zugreifbar gemacht werden. Gemeinsam genutzte Objekte müssen eindeutig über eine Objektidentifikation (Object ID) ansprechbar sein.

Folgende Definitionen werden in der Spezifikation des Gateway-Modells verwendet [Java Card Forum 1997]:

Ticket Checker: Ein Ticket Checker ist ein autorisierendes Objekt, das von dem freigebenden Applet erzeugt wurde, um ein bestimmtes Objekt für andere Applets nutzbar zu machen. Ein Ticket Checker kann ein spezifisches Ticket mittels Paßwortvergleich oder Challenge Response Verfahren überprüfen.

Ticket: Ein Ticket ist eine Einheit, die den Benutzer repräsentiert, der auf ein bestimmtes Objekt zugreifen darf. Tickets werden von Applets erzeugt, die Zugriff auf Objekte anderer Applets haben möchten. Das Ticket-erzeugende Applet muß im voraus wissen, welche Art von Tickets benötigt werden.

Rights: Ein Recht ist ein abstraktes Privileg. Es stellt die Erlaubnis dar, ein Objekt eines anderen Applets nutzen zu können.

Das Gateway-Modell beschreibt ein Konzept, das Applets erlaubt, spezielle Objekte mit anderen Applets zu teilen. Dies wird in Abbildung 3-19 dargestellt. Um ein Objekt gemeinsam nutzen zu können, muß das Applet, welches das Objekt erzeugt und freigeben möchte, ein autorisierendes Objekt erzeugen, den sogenannten *Ticket Checker*. Ein Applet, das das Objekt nutzen möchte, muß ein gültiges *Ticket* vorweisen können. Das Gateway nutzt nun den *Ticket Checker* des freigebenden Applets, um das *Ticket* des anfragenden Applets zu prüfen.

4 Kryptographische Verfahren

Seit den Anfängen der Chipkarten-Entwicklung kommt der Kryptographie eine wesentliche Bedeutung zu. Die praktische Anwendung von kryptographischen Verfahren und Methoden ist die zentrale Aufgabenstellung für Chipkarten. Neben der Funktion als Datenträger dienen Chipkarten als Berechtigungsträger und Verschlüsselungsmodule. Die Verfahren der Kryptographie gehören mittlerweile fest zur Chipkartentechnik. Kryptographie wird als die Wissenschaft von Methoden der Ver- und Entschlüsselung bezeichnet ([Rankl, Effing 1996], S. 86).

Ziel der Kryptographie ist zum einen die Geheimhaltung von Daten und zum anderen die Sicherstellung der Authentizität von Nachrichten. Beide Ziele sind unabhängig voneinander und stellen unterschiedliche Anforderungen an das jeweilige System.

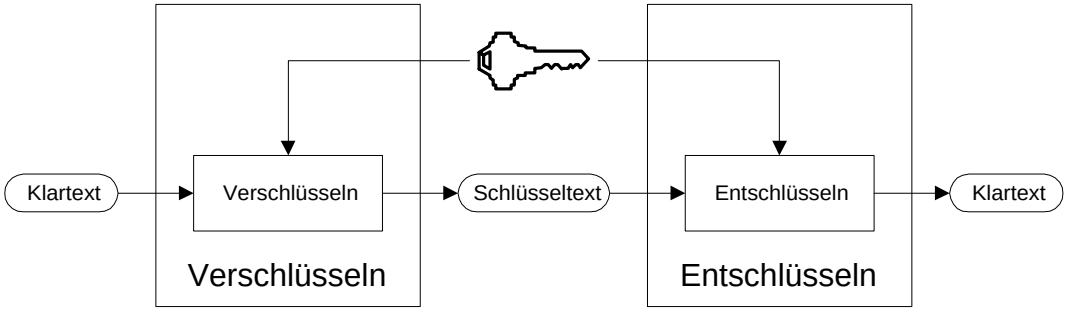


Abbildung 4-1: Ver- und Entschlüsselung bei symmetrischen Algorithmen

Kryptoalgorithmen teilt man in symmetrische und asymmetrische Algorithmen ein. Diese Einteilung bezieht sich auf die verwendeten Schlüssel. Bei symmetrischen Verfahren wird derselbe geheime Schlüssel sowohl zum Verschlüsseln als auch zum Entschlüsseln verwendet. Bei asymmetrischen Verfahren verwendet man ein Schlüsselpaar. Dieses Schlüsselpaar besteht aus einem öffentlichen Schlüssel (Public Key) und einem geheimen Schlüssel (Private Key). Zum Verschlüsseln verwendet man den öffentlichen Schlüssel und zum Entschlüsseln den geheimen Schlüssel. Asymmetrische Kryptoalgorithmen wurden 1976 von Whitfield Diffie und Martin E. Hellman postuliert [Diffie, Hellman 1976].

Bei der Kryptographie unterscheidet man zwischen theoretischer und praktischer Sicherheit eines Algorithmus oder Systems. Ein System ist theoretisch sicher, wenn einem Angreifer unbegrenzte Zeit und Hilfsmittel zur Verfügung stehen und es für ihn selbst dann unmöglich ist, das System zu brechen. Stehen einem Angreifer nur begrenzte Zeit und Hilfsmittel zur Vefügung und kann er damit das System nicht brechen, bezeichnet man das System als praktisch sicher. Die Stärke eines Systems hängt vom Schlüssel ab und nicht von der Geheimhaltung des Verfahrens. Um ein System zu brechen, muß somit der Schlüssel gebrochen werden. Dafür gibt es verschiedene Angriffsmöglichkeiten, wie zum Beispiel „ciphertext only attack“ (Angreifer kennt nur den Schlüsseltext), „known plaintext attack“ (Angreifer kennt Klartext-Schlüsseltext-Paare), „chosen plaintext attack“ oder „chosen ciphertext attack“

(Angreifer kann eigene Klar- beziehungsweise Schlüsseltexte erzeugen) und letztendlich „brute force attack“ (Angreifer probiert auf Grundlage eines bekannten Klartext-Schlüsseltext-Paares alle möglichen Schlüssel aus). Durch die zunehmende Vernetzung von Computern sind die Angriffsmöglichkeiten in letzter Zeit rapide gewachsen. Man betrachte DES-Challenge-I [DES-Challenge 1997] und DES-Challenge-II [DES-Challenge 1998], wobei in kürzester Zeit durch Zusammenschalten von Rechnerkapazitäten im Internet die Schlüssel herausgefunden wurden. Die neueste Entwicklung zeigt sich in der DES-Challenge-III, wobei in 22 Stunden der 56 Bit lange Schlüssel durch Zusammenschalten vieler Rechner im Internet herausgefunden wurde [DES-Challenge 1999].

4.1 Symmetrische Algorithmen

Als symmetrische Algorithmen bezeichnet man Verfahren, die zur Ver- und Entschlüsselung den gleichen Schlüssel verwenden. Der bekannteste symmetrische Algorithmus ist der Data Encryption Algorithm (DEA), der 1977 als US-Norm publiziert wurde. Die Norm, die den DEA beschreibt, wird als DES (Data Encryption Standard) bezeichnet [FIPS 46]. Der DEA ist ein symmetrischer Blockverschlüsselungsalgorithmus, wobei Klartext- und Schlüsseltextblöcke gleich lang sind. Die Blocklänge beträgt ebenso wie die Schlüssellänge 64 Bit (56 Bit Schlüssel und acht Bit Parität).

Im Januar 1996 wurde von führenden Kryptologen nachgewiesen, daß die Schlüssellänge von 64 Bit nicht ausreichend sicher ist [Blaze, Diffie, Rivest, Schneier, Shimomura, Thomson, Wiener 1996]. Der DEA kann wie jeder Blockverschlüsselungsalgorithmus in vier verschiedenen Betriebsarten betrieben werden, die in der ISO-Norm 8372 genormt sind. Zwei sind speziell für sequentielle Texte ohne Blockstruktur geeignet, zwei für Texte mit Blockstruktur von acht Byte Größe.

ECB Electronic Code Book: Im ECB wird der Klartext in acht Byte lange Blöcke aufgeteilt, die dann mit demselben Schlüssel unabhängig voneinander verschlüsselt werden. Die Entschlüsselung läuft analog ab.

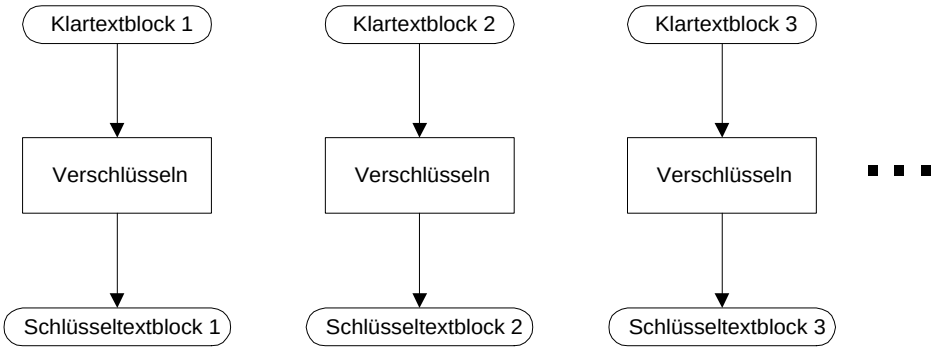


Abbildung 4-2: DES im ECB-Modus

CBC Cipher Block Chaining: In diesem Modus wird der Klartext mit einem acht Byte langen Initialisierungsvektor mit XOR verknüpft, bevor er mit DES verschlüsselt wird. Das Ergebnis ist der Schlüsseltext, der wiederum mit dem folgenden acht Byte langen Klartextblock XOR verknüpft wird. Analog wird mit allen folgenden Blöcken verfahren. Durch die XOR-Verkettung wird erreicht, daß die nachfolgenden Blöcke von den vorhergehenden abhängig werden. Damit kann das Vertauschen, Einfügen oder Löschen von verschlüsselten Blöcken zuverlässig erkannt werden.

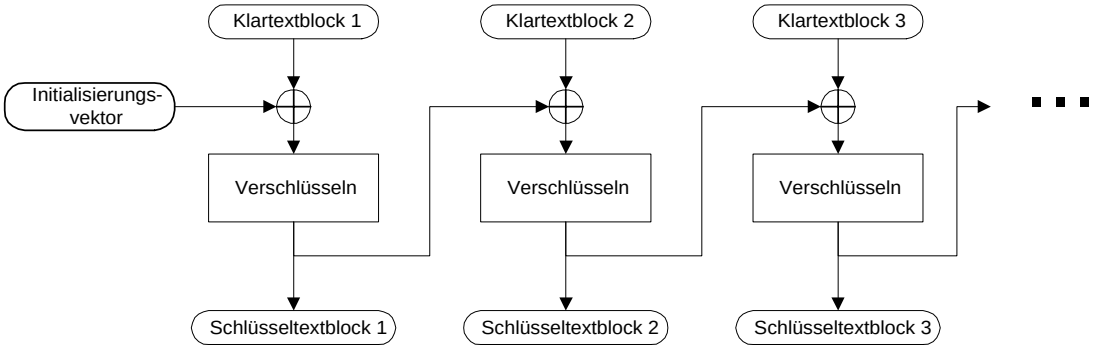


Abbildung 4-3: DES im CBC-Modus

OFB und CFB Output FeedBack, Cipher FeedBack: Diese beiden Modi sind zeichenorientiert. Üblicherweise besteht ein Zeichen aus ein Bit beziehungsweise acht Bit. Dabei wird in jedem Schritt ein Klartextzeichen mit einem Schlüssel mit XOR verknüpft. Der benötigte Schlüsselstrom wird mit Hilfe des DES erzeugt. Dieser verschlüsselt den Inhalt eines 64 Bit Schieberegisters. Von der Ausgabe werden die linken (obersten) Bits als Schlüssel für das XOR genommen. Vor der nächsten Runde wird ein Zeichen von rechts in das Schieberegister eingelesen und so ein neuer Schlüssel erzeugt. Beim OFB ist dieser Wert das Ergebnis des DES, beim CFB das Ergebnis des XOR.

Zusätzlich zu den vier Betriebsmodi eines Blockverschlüsselungsalgorithmus existiert eine weitere Variante, Triple-DES, die zur Erhöhung der Sicherheit beiträgt. Dazu schaltet man drei DEA-Operationen mit abwechselnder Ver- und Entschlüsselung hintereinander. Es besteht die Möglichkeit, für jeden Vorgang einen eigenen Schlüssel zu verwenden. Deshalb kann sich die Anzahl der Schlüssel auf drei erhöhen. In der Regel werden jedoch nur zwei Schlüssel benutzt (Abbildung 4-4).

Die Entschlüsselung des so dreifach verschlüsselten Blocks erfolgt durch Umdrehen der Reihenfolge der Operationen, also Entschlüsselung - Verschlüsselung - Entschlüsselung mit dem jeweils richtigen Schlüssel. Sind alle drei Schlüssel identisch, so erhält man durch die abwechselnde Ver- und Entschlüsselung das gleiche Ergebnis wie bei einer einfachen Verschlüsselung. Es werden statt einem 64 Bit Schlüssel also drei benötigt, von denen aber der erste und der dritte gleich sind. Dies führt zu einer Schlüssellänge von 128 Bit (112 Bit Schlüssel + 16 Bit Parität). Damit kann das Verfahren mit normalen DEA realisiert werden und erfordert außer einem zweiten Schlüssel und der dreifachen Verschlüsselungszeit keinerlei Mehraufwand.

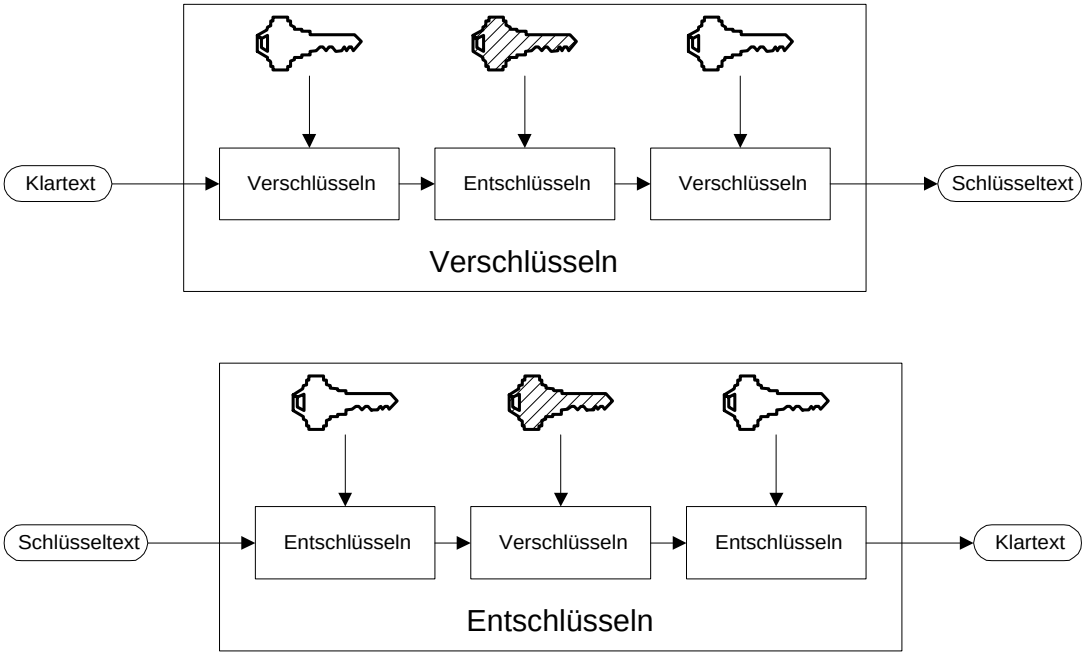


Abbildung 4-4: Prinzipieller Ablauf bei Triple-DES

Neben dem DEA existieren noch viele weitere symmetrische Kryptoalgorithmen. Stellvertretend dafür sei an dieser Stelle der IDEA (International Data Encryption Algorithm) genannt. Er wurde im Jahre 1990 veröffentlicht und im Jahr 1992 als PES (Proposed Encryption Standard) verbessert. Er ist jedoch heute als IDEA bekannt. Der IDEA ist wie der DEA ein blockorientierter Kryptoalgorithmus und benutzt ebenfalls 8 Byte lange Klar- und Schlüsseltextblöcke. Die Schlüssellänge beträgt im Gegensatz zum DEA jedoch 128 Bit, so daß sich ein wesentlich größerer Schlüsselraum mit 2^{128} verschiedenen Schlüsseln ergibt. Durch seinen Aufbau ist er bis auf die vergrößerte Schlüssellänge zum DEA kompatibel. Zu Triple-DES Systemen, die zwei acht Byte lange Schlüssel verwenden, besteht ebenfalls eine vollständige Kompatibilität. Kompatibilität bedeutet in diesem Zusammenhang jedoch nicht, daß mit DEA verschlüsselte Daten mit dem IDEA entschlüsselt werden können.

Bei der Entwicklung von IDEA ist man davon ausgegangen, daß die Berechnungen durch einen 16 Bit Prozessor durchgeführt werden. Da Chipkarten zur Zeit hauptsächlich 8 Bit Prozessoren im Einsatz haben, wird die Verwendung von IDEA erst dann von Interesse sein, wenn in Chipkarten durchgängig 16 Bit Prozessoren eingesetzt werden. Dies wird nach heutigem Entwicklungsstand jedoch nicht mehr lange dauern.

Im Januar 1997 wurde vom National Institute of Standard and Technology (NIST) eine Ausschreibung mit dem Ziel gestartet, den über 20 Jahre alten Data Encryption Standard (DES) durch einen neuen Advanced Encryption Standard (AES) abzulösen. Die US-Behörde rief in ihrer Ausschreibung öffentlich auf, bis zum Juni 1998 einen Vorschlag für den neuen AES einzureichen [NIST 1997]. Dabei wurden strenge Anforderungen an den zu entwickelnden Algorithmus gestellt. Im August 1998 fand die erste AES Konferenz statt, auf der 15 Kandidaten für den neuen Algorithmus offiziell nominiert wurden (Tabelle 4-1).

Nach dieser Konferenz startete eine technische Überprüfungsphase (Round One), in der die einzelnen Algorithmen getestet werden sollten. Die Algorithmen wurden für ausführliche Tests der Öffentlichkeit zur Verfügung gestellt. Die Diskussion über die Algorithmen erfolgt ebenfalls öffentlich, so daß jede Kapazität auf dem Gebiet der Kryptologie mit einbezogen werden kann.

Algorithmus	**Entwicklers**
CAST-256	Entrust Technologies
CRYPTON	Future Systems
DEAL	Richard Outerbridge, Lars Knudson
DFC	Centre National pour la Recherche Scientifique - Ecole Normale Superieure (CNRS)
E2	Nippon Telepgraph and Telephone Coorporation (NTT)
FROG	TecApro
HPC	Rich Schroeppel
LOKI97	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
MAGENTA	Deutsche Telekom
MARS	IBM
RC6	RSA Laboratories
RIJNDAEL	Joan Daemen, Vincent Rijmen
SAFER+	Cylink Coproration
SERPENT	Ross Anderson, Eli Biham, Lars Knudson
TWOFISH	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

Tabelle 4-1: Vorgeschlagene Algorithmen für den Advanced Encryption Standard (AES)
Quelle: [NIST 1999], S. 2

Im März 1999 endet die erste Überprüfungsphase (Round One) mit der zweiten AES Konferenz in Rom, auf der die besten fünf Algorithmen ermittelt werden. Anschließend geht es in eine zweite Überprüfungsphase (Round Two), für die minimale Verbesserungen der ausgewählten Kandidaten durchgeführt werden dürfen. Diese Runde endet mit der dritten AES Konferenz, auf der dann der endgültige Algorithmus für den Advanced Encryption Standard (AES) ermittelt wird, dies wird voraussichtlich Ende 1999 sein.

4.2 Asymmetrische Algorithmen

Asymmetrische Verschlüsselungsalgorithmen basieren auf zwei unterschiedlichen Schlüsseln, von denen der eine öffentlich und der andere geheim ist. Das Prinzip des öffentlichen und geheimen Schlüssels ermöglicht es, daß jemand mit dem öffentlichen Schlüssel eine Nachricht verschlüsselt, die nur derjenige entschlüsseln kann, der im Besitz des geheimen Schlüssels ist. 1978 stellten Ronald L. Rivest, Adi Shamir und Leonard Adleman einen Algorithmus (RSA) vor, der die obigen Voraussetzungen erfüllte [Rivest, Shamir, Adleman 1978]. Es handelt sich um einen asymmetrischen Algorithmus, bei dem es egal ist, in welcher Reihenfolge die Schlüssel verwendet werden. Daher ist es möglich, dasselbe Schlüsselpaar sowohl zum Ver-/Entschlüsseln als auch zum Signieren von Nachrichten zu verwenden. Das Verfahren gilt bei ausreichender Schlüssellänge (größer 1024 Bit) als recht sicher, allerdings sind die notwendigen Berechnungen sehr zeitaufwendig. Daher verwendet man RSA in der Regel nur, um einen Schlüssel für ein symmetrisches Verfahren zu verschlüsseln.

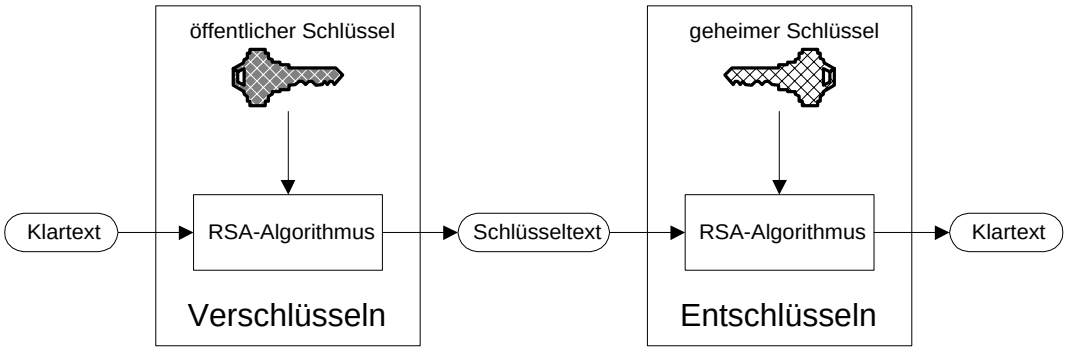


Abbildung 4-5: Ver- und Entschlüsselung bei asymmetrischen Algorithmen

Der nach seinen drei Erfindern benannte RSA-Algorithmus ist der bekannteste und der am vielseitigsten einsetzbare asymmetrische Kryptoalgorithmus, der zur Zeit verwendet wird. Das Funktionsprinzip basiert auf der Arithmetik großer Integerzahlen. Die beiden Schlüssel werden auf der Grundlage von zwei großen Primzahlen erzeugt. Die Ver- und Entschlüsselung läßt sich mathematisch wie folgt ausdrücken:

Verschlüsseln: $y = x^e \bmod n$

Entschlüsseln: $x = y^d \bmod n$

mit $n = p * q$

wobei
- x: Klartext
- y: Schlüsseltext
- e: öffentlicher Schlüssel
- d: geheimer Schlüssel
- n: öffentlicher Modulus
- p, q: geheime Primzahlen

Die Sicherheit des Algorithmus basiert auf dem Faktorisierungsproblem großer Zahlen. Es ist sehr einfach, den öffentlichen Modulus aus den beiden Primzahlen durch

Multiplikation zu berechnen, aber sehr schwierig, den Modulus wieder in seine beiden Primfaktoren zu zerlegen, da keine effektiven Algorithmen dafür bekannt sind.

Die Erzeugung von Schlüsseln für den RSA-Algorithmus erfolgt nach einem Schema, anhand dessen ein kleines Beispiel durchgerechnet werden soll ([Rankl, Effing 1996], S. 96):

1) Zuerst werden die zwei Primzahlen p und q gesucht. $p = 3, q = 11$
2) Nun berechnet man den öffentlichen Modulus. $n = p * q = 33$
3) Berechnung einer Hilfsvariablen z für die Schlüsselerzeugung. $z = (p - 1) * (q - 1) = 20$
4) Berechnung des öffentlichen Schlüssels e mit den folgenden Eigenschaften: $(e < z)$ und $(ggt(z,e) = 1)$. Das heißt, der größte gemeinsame Teiler der Zahlen z und e ist eins. Da es mehrere Zahlen mit dieser Eigenschaft gibt, wähle man eine aus. $e = 7$
5) Berechnung des geheimen Schlüssels d mit der Eigenschaft: $(d * e) \bmod z = 1$. $d = 3$

Nun ist die Schlüsselberechnung abgeschlossen. Der öffentliche und der geheime Schlüssel können nun an einem weiteren Zahlenspiel zur Ver- und Entschlüsselung mit dem RSA-Algorithmus getestet werden.

1) Als Klartext x $(x < n)$ verwendet man die Zahl 4. $x = 4$
2) Verschlüsselung. $y = 4^7 \bmod 33 = 16$
3) Es ergibt sich aus der Berechnung ein Schlüsseltext y mit dem Wert 16 $y = 16$
4) Entschlüsselung. $x = 16^3 \bmod 33 = 4$

Das Ergebnis der Entschlüsselung des Schlüsseltextes ist wiederum der Klartext.

Mit steigender Schlüssellänge steigt auch der Rechenaufwand zur Ver- und Entschlüsselung. Diese Steigerung ist jedoch nicht linear, sondern exponentiell. Es ist jedoch eine der Stärken des RSA, daß er nicht auf eine bestimmte Schlüssellänge fixiert ist. Die Verwendung des RSA zur Verschlüsselung von Daten wird wegen der langen Rechenzeiten selten genutzt, obwohl sie sehr sicher ist. Das Haupteinsatzgebiet ist im Bereich der digitalen Signatur, da hier die Vorteile dieses asymmetrischen Verfahrens voll zur Geltung kommen. Die Verbreitung von RSA hemmen jedoch die in einigen Ländern gemachten Patentansprüche zum Algorithmus, sowie die großen Restriktionen der USA bei der Ausfuhr von RSA benutzenden Geräten. Chipkarten mit einem Coprozessor für den RSA-Algorithmus fallen unter diese Bestimmungen. Dieselben Beschränkungen gelten ebenso für DES.

Mitte 1991 veröffentlichte das NIST (US National Institute of Standards and Technology) einen Entwurf für einen kryptographischen Algorithmus zum Signieren von Nachrichten. Der mittlerweile in einer US-Norm [FIPS 186] genormte Algorithmus hat die Bezeichnung DSA (Digital Signature Algorithm) und die ihn beschreibende Norm den Namen DSS (Digital Signature Standard). Mit dieser Norm sollte ein Verfahren verbreitet werden, mit dem nur Signaturen erstellt werden können, jedoch keine Verschlüsselung von Daten möglich ist. Mittlerweile ist eine Möglichkeit gefunden worden, mit diesem Algorithmus Daten zu verschlüsseln. Welcher der beiden Algorithmen RSA oder DSA zur Erstellung von digitalen Signaturen sich langfristig durchsetzt oder sicherer ist, läßt sich zum heutigen Zeitpunkt nicht sagen.

Eine weitere Entwicklung im Bereich digitaler Signaturen stellen Systeme auf der Basis elliptischer Kurven dar [Koblitz 1987]. Die Sicherheit der meisten implementierten Signatursysteme beruht auf der Komplexität des Faktorisierungsproblems oder der des Diskreten Logarithmusproblems ([Fox, Röhm 1996] und [Hühnlein 1996]). Für beide zahlentheoretischen Probleme existieren Lösungsalgorithmen mit subexponentiellem Aufwand. Für elliptische Kurven läßt sich ein Diskretes Logarithmusproblem definieren, für das bis heute kein allgemein anwendbarer Lösungsalgorithmus mit subexponentiellem Aufwand bekannt ist, sondern nur Algorithmen mit höherem (exponentiellem) Aufwand [Hühnlein 1998].

Aus diesem Grund kann man digitale Signatursysteme auf der Basis geeigneter elliptischer Kurven definieren, die bei vergleichbarem Sicherheitsniveau mit erheblich kürzeren Schlüsseln arbeiten als beispielsweise RSA oder DSA. Geringere Schlüssellängen führen zu kürzeren Signaturen und Zertifikaten. Sie reduzieren so den Speicherbedarf, ermöglichen schnellere arithmetische Operationen und erhöhen damit die Geschwindigkeit der Prüf- und Signaturalgorithmen. Dies ist für die Implementierung auf Chipkarten von großer Bedeutung.

4.3 Hash-Algorithmen

Da Nachrichten im Regelfall einige tausend Byte lang sind, wird, um die Rechenzeit zur Bildung der kryptographischen Prüfsumme in akzeptablen Grenzen zu halten, über den gesamten Datenstring zuerst ein Hash-Wert gebildet.

Hash-Funktionen sind, vereinfacht ausgedrückt, Einwegfunktionen zur Komprimierung von Daten. Aus einem Dokument mit variabler Länge wird ein Wert mit fester Länge berechnet. Diese Komprimierung ist nicht umkehrbar, man kann also aus den komprimierten Daten nicht wieder das Original herstellen.

Bekannt sind der Message Digest 4 (MD4) und seine Weiterentwicklung MD5 von R. L. Rivest aus den Jahren 1990 und 1991. Sie basieren auf einem eigenständigen Algorithmus und produzieren einen 128 Bit langen Hashwert. 1992 wurde vom NIST der Secure Hash Algorithmus (SHA) veröffentlicht, als eine Hash-Funktion für den DSS. Eine Überarbeitung seiner Schwächen wurde 1995 als SHA-1 veröffentlicht.

Da die Berechnung eines Hash-Wertes sehr schnell ist, werden Hashfunktionen bei der Erzeugung und Prüfung von digitalen Signaturen verwendet.

4.4 Digitale Signaturen

Digitale Signaturen, oft auch elektronische Unterschriften genannt, werden zur Feststellung der Authentizität von elektronisch übermittelten Nachrichten oder elektronischen Dokumenten verwendet ([Rankl, Effing 1996], S. 277). Durch Überprüfung der digitalen Signatur läßt sich weiterhin feststellen, ob Nachrichten oder Dokumente verändert wurden.

Eine handschriftliche Unterschrift hat die Eigenschaft, daß sie nur von einem einzigen Menschen korrekt erzeugt, aber von allen Empfängern der Nachricht überprüft werden kann, zumindest von denen, die eine echte Unterschrift schon einmal gesehen haben, oder denen sie zum Vergleich vorliegt. Dies muß auch eine wesentliche Eigenschaft einer digitalen Signatur sein. Nur eine einzige Person soll ein Dokument digital unterschreiben können, aber jedermann kann überprüfen, ob die digitale Signatur echt ist.

Aufgrund dieser geforderten Eigenschaft stellen asymmetrische kryptographische Verfahren die ideale Ausgangsbasis dar. Technisch gesehen ist eine digitale Signatur eng verwandt mit kryptographischen Prüfsummen. In den folgenden Abbildungen wird exemplarisch jeweils der RSA-Algorithmus für das Erzeugen und das Prüfen der digitalen Signatur verwendet.

Abbildung 4-6 verdeutlicht den prinzipiellen Ablauf des Erzeugens und des Prüfens einer Signatur ohne Hash-Funktionen. Bei der Erzeugung von Signaturen wird prinzipiell zwischen dem Erzeugen ohne Hash-Funktionen und mit Hash-Funktionen unterschieden.

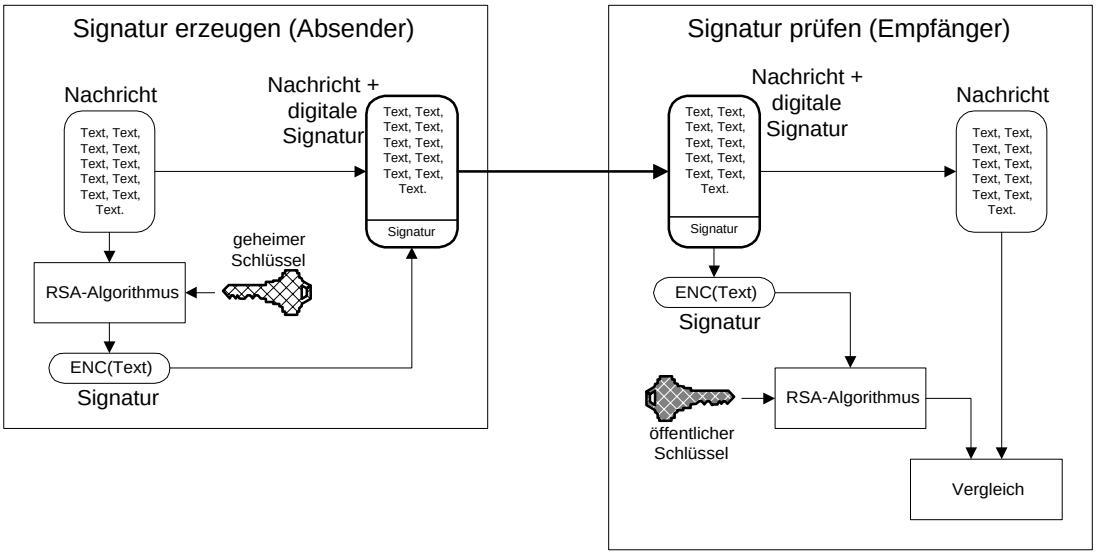


Abbildung 4-6: Erzeugen und Prüfen einer digitalen Signatur

Häufig werden jedoch Hash-Funktionen zum Erzeugen von digitalen Signaturen verwendet, wenn es sich um umfangreiche Texte handelt, die signiert werden sollen. Hash-Funktionen erzeugen aus einem Text variabler Länge einen Text konstanter Länge, den sogenannten Hash-Wert. Abbildung 4-7 beschreibt den Ablauf des Erzeugens und des Prüfens einer digitalen Signatur unter Verwendung von Hash-Funktionen.

Im folgenden wird das Erzeugen und das Prüfen im Zusammenhang mit Hash-Funktionen erläutert, da diese Art der Signaturerzeugung eine breitere Verwendung als ohne Hash-Funktionen findet.

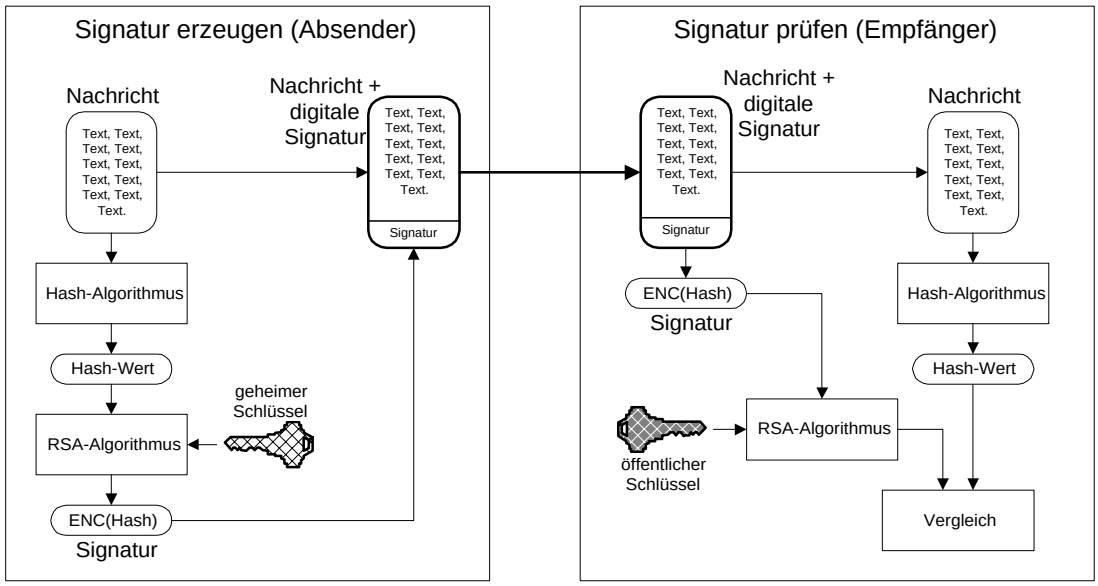


Abbildung 4-7: Erzeugen und Prüfen einer digitalen Signatur unter Verwendung einer Hash-Funktion

4.4.1 Erzeugen einer Signatur

Zur Erzeugung einer digitalen Signatur wird ein Hash-Algorithmus (siehe Kapitel 4.3) verwendet, der die Nachricht in eine konstante Länge verarbeitet. Dieser Hashwert der Nachricht wird dann signiert.

Jeder Benutzer hat ein eigenes Schlüsselpaar aus öffentlichem und geheimem Schlüssel. Der geheime Schlüssel kann beispielsweise auf einer Chipkarte unauslesbar gespeichert werden, während der öffentliche Schlüssel für alle Kommunikationspartner verfügbar ist. Die Erzeugung einer Signatur läuft nun wie folgt ab: Aus einer Nachricht, zum Beispiel einer mit einer beliebigen Textverarbeitung erstellten Datei, wird mit einem Hash-Algorithmus ein Hash-Wert gebildet. Dieser Hash-Wert wird mit einem asymmetrischen Kryptoalgorithmus entschlüsselt. Man beachte, daß gemäß der Konvention von asymmetrischen Kryptoalgorithmen immer der geheime Schlüssel zum Entschlüsseln und der öffentliche Schlüssel zum Verschlüsseln verwendet wird.

In Abbildung 4-6 und Abbildung 4-7 wird dazu der RSA-Algorithmus einmal mit und ohne Hash-Funktion verwendet. Zum Entschlüsseln wird der geheime Schlüssel des Nachrichtenerzeugers verwendet. Das Ergebnis der Berechnung ist die eigentliche digitale Signatur, die an die Nachricht angehängt wird.

4.4.2 Prüfen einer Signatur

Die Nachricht kann nun über einen unsicheren Weg zum Empfänger geschickt werden. Dieser trennt Nachricht und Unterschrift wieder voneinander. Die Nachricht wird mit dem gleichen Hash-Algorithmus wie beim Sender komprimiert. Die digitale Signatur

wird mit dem öffentlichen Schlüssel des Nachrichtenerzeugers verschlüsselt und mit dem Ergebnis der Hash-Berechnung verglichen (Abbildung 4-7). Sind beide Hash-Werte gleich, so wurde die Nachricht auf ihrem Übertragungsweg nicht verändert und der Absender der Nachricht ist tatsächlich der Nachrichtenerzeuger. Im anderen Fall wurde entweder die Nachricht oder digitale Signatur während der Übertragung verändert oder ein falscher Schlüssel verwendet.

Zur Erstellung von digitalen Signaturen läßt sich nicht nur der RSA-Algorithmus verwenden, sondern ebenso der bereits erwähnte DSA (Digital Signature Algorithm). Dieser hat gegenüber RSA, wegen dessen Exportbeschränkungen, große Vorteile für den internationalen Einsatz.

4.5 Zertifikate und Infrastrukturen

Ein Zertifikat ist eine Sammlung von Informationen, die von einer Institution signiert wurde. Diese Institution wird von den Nutzern der Zertifikate als vertrauenswürdig anerkannt ([Ford, Baum 1997], S. 193ff). Zertifikate können in verschiedenen Arten unterschiedliche Zwecke erfüllen. Eine der wichtigsten Zertifikatsarten sind Zertifikate über die Echtheit öffentlicher Schlüssel. Diese Zertifikate bestätigen die Zuordnung eines öffentlichen Schlüssels zu einer Person oder Institution, dem Subjekt des Zertifikates. Im wesentlichen handelt es sich um digitale Zertifikate, die von einer Zertifizierungsinstanz mit einer digitalen Signatur versehen werden. Die Zertifizierungsinstanz bezeugt damit die Identität oder andere Attribute des Schlüsselbesitzers.

Öffentliche Schlüssel werden im wesentlichen für zwei Zwecke gebraucht. Zum einen wird der öffentliche Schlüssel des Empfängers bei asymmetrischen Verfahren zum Verschlüsseln von Nachrichten benötigt (siehe Kapitel 4.2). Zum anderen wird der öffentliche Schlüssel eines Absenders einer signierten Nachricht benötigt, um dessen Signatur zu prüfen. Mit dem öffentlichen Schlüssel des Signaturerzeugers kann die Echtheit der Signatur geprüft werden (siehe Kapitel 4.4.2).

Die Echtheit dieser öffentlichen Schlüssel ist notwendig, damit keine gefälschten Schlüssel versehentlich benutzt werden. Ein System von Zertifikaten über die Echtheit öffentlicher Schlüssel arbeitet mit Zertifizierungsinstanzen. Diese sind Institutionen, die von den Nutzern der Zertifikate anerkannt sind. Sie stellen Zertifikate für die Besitzer von öffentlichen-geheimen Schlüsselpaaren aus.

Diese Zertifikate (Abbildung 4-8) enthalten den öffentlichen Schlüssel und Identifikationsinformationen des Subjekts, also einer Person oder einer Institution, die den korrespondierenden geheimen Schlüssel besitzt. Dieses Zertifikat wird von der Zertifizierungsinstanz mit deren geheimen Schlüssel digital signiert.

Schlüsselpaare haben in der Regel eine begrenzte Gültigkeitsdauer. Deshalb sind auch die Zertifikate mit einem bestimmten Gütligkeitszeitraum versehen. Nach dem Ablaufdatum eines Zertifikats ist die Zuordnung eines öffentlichen Schlüssels zu dem Subjekt des Zertifikats nicht mehr gewährleistet und das Zertifikat ist nicht mehr vertrauenswürdig.

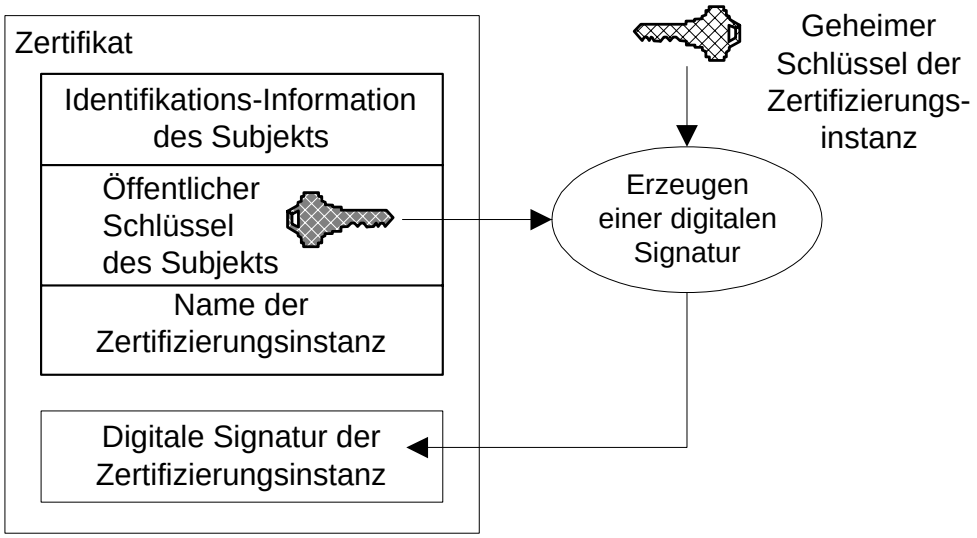


Abbildung 4-8: Generelles Zertifikat
Quelle:[Ford, Baum 1997], S. 195

Zertifikate können auch vor Ablauf ihrer Gültigkeit unbenutzbar werden, wenn der Besitzer eines Zertifikats zum Beispiel nicht mehr für eine bestimmte Organisation arbeitet, obwohl sein Schlüsselpaar noch Gültigkeit besitzt. Die Zertifizierungsinstanz widerruft in einem solchen Fall das Zertifikat (Revocation) und beendet damit dessen Gültigkeitszeitraum. Dieser Widerruf kann den anderen Zertifikatsnutzern in einer Widerrufsliste (Certificate Revocation List, CRL) bekannt gemacht werden.

Zertifikate können über unsichere Kanäle verteilt werden, ohne mit den typischen Sicherheitsmaßnahmen gegen Verlust von Vertraulichkeit oder Integrität geschützt werden zu müssen, da die digitale Signatur im Zertifikat selbst die Vertraulichkeit und Integrität unterstützt. Auf diese Weise kann ein Nutzer die öffentlichen Schlüssel unzähliger Parteien erhalten, ohne diese alle kennen zu müssen. Es reicht die Kenntnis eines einzigen öffentlichen Schlüssels, nämlich der Zertifizierungsinstanz, aus, mit dessen Hilfe man über einen Zertifizierungspfad zu dem gewünschten Zertifikat gelangt. Dies setzt voraus, daß alle Beteiligten dem öffentlichen Schlüssel der Wurzel-Zertifizierungsinstanz vertrauen, die alle weiteren Zertifikate digital signiert.

Es gibt prinzipiell verschiedene Wege, einen Zertifizierungspfad von einem gewünschten Zertifikat zu der Wurzel-Zertifizierungsinstanz aufzubauen. Die benötigte Infrastruktur, die dem zugrundeliegt, kann zum Beispiel hierarchisch oder vernetzt aufgebaut sein.

Eine generelle hierarchische Struktur ist eine Baumstruktur von Zertifizierungsinstanzen, deren Knoten sowohl die darunter als auch die darüber liegenden Knoten zertifiziert. Zusätzliche Verbindungen können zwischen den einzelnen Knoten eingefügt werden.

Im Unterschied zu einer generellen Zertifizierungshierarchie zertifiziert eine Top-Down-Zertifizierungshierarchie (Abbildung 4-9) ausgehend von einer Wurzelinstanz jeweils nur die darunterliegenden Instanzen.

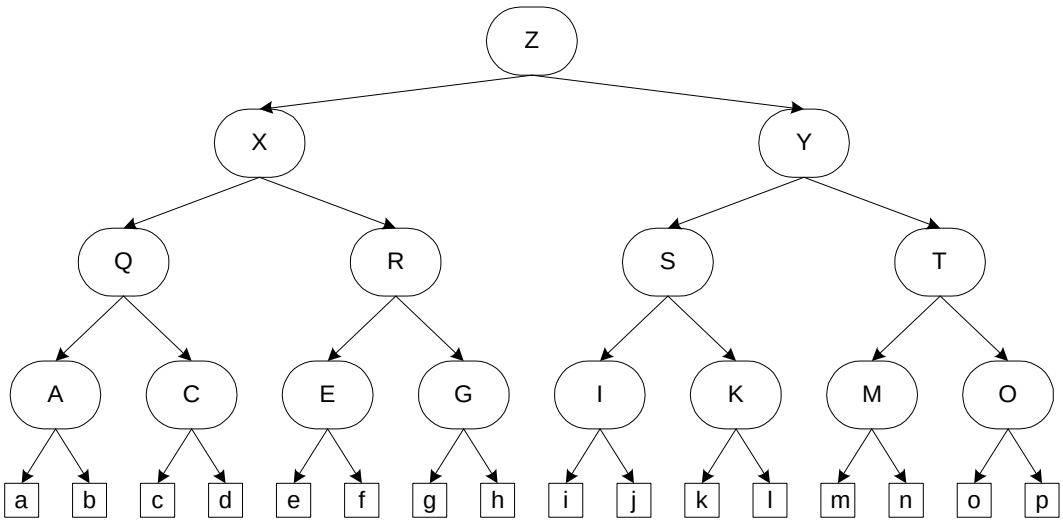


Abbildung 4-9: Top-Down-Zertifizierungshierarchie

Quelle: [Ford, Baum 1997], S. 270

Bei einer Top-Down-Hierarchie beruhen alle Zertifikate auf dem Zertifikat der obersten Zertifizierungsinstanz. Jede Zertifizierungsinstanz signiert jeweils die Zertifikate der darunterliegenden Instanz, bis hin zu den Zertifikaten der einzelnen Nutzer. Die Tatsache, daß alles auf dem Vertrauen in die oberste Instanz beruht, kann Vorteil und Nachteil zugleich sein. Vorteil ist die einfache Struktur und die zentrale Instanz zur Vertrauensschaffung, während der Nachteil darin liegt, daß ein Angreifer nur eben diese zentrale Stelle manipulieren muß.

Neben Hierarchien, die auch Querverbindungen zulassen und Verknüpfungen von Hierarchien, ist die netzbasierte Infrastruktur erwähnenswert, wie sie zum Beispiel bei Pretty Good Privacy verwendet wird ([Zimmermann 1995] und [Garfinkel 1995]). Pretty Good Privacy (PGP) arbeitet nicht mit Zertifizierungsinstanzen, sondern jeder Benutzer kann öffentliche Schlüssel von anderen Benutzern signieren. So entsteht ein Vertrauensnetz, das sich quer durch alle PGP-Nutzer zieht. In einer netzbasierten Infrastruktur ist es schwieriger, einen Zertifizierungspfad von einem gewünschten Zertifikat zu einer vertrauenswürdigen Stelle zu erhalten, da es keine Wurzel-Zertifizierungsinstanz gibt.

Ein erweitertes Vertrauensnetz ist im X.509-Standard beschrieben [ISO 9594-8], in dem jede Zertifizierungsinstanz jeder anderen Instanz ein Zertifikat ausstellen darf, das erweiterte Attribute, wie zum Beispiel Angaben über die Zertifizierungspolitik, enthält. Eine Zertifizierungspolitik beschreibt, auf welche Weise Zertfikate vergeben werden, und macht Aussagen über die Anwendbarkeit von Zertifikaten.

Zertifikate stellen nicht nur hohe technische Anforderungen an die Infrastruktur, in die sie eingebettet sind, sondern auch organisatorische und rechtliche Anforderungen. Sie sind ein wesentlicher Bestandteil für die Wahrung von Vertraulichkeit und Integrität. Neben Geheimhaltung und Authentisierung von Nachrichten ist auch die Identifizierung und Authentisierung von Personen eine wichtige Aufgabe für sichere elektronische Transaktionen.

4.6 Identifizierung und Authentisierung

Die Identifizierung von Personen und deren Authentisierung ist seit langem ein Thema, mit dem sich Menschen beschäftigen ([Rankl, Effing 1996], S. 258ff). Die einfachste Form der Identifizierung und Authentisierung ist die Benutzung eines Ausweises mit Photo oder die Leistung einer Unterschrift in Gegenwart eines Prüfers. Das Ergebnis des Vergleichs ist eine Aussage über die Echtheit der Person, die sogenannte Authentisierung. Im informationstechnischen Bereich ist dieser Vergleich nicht immer so einfach, da ein Computer den Vergleich vornehmen muß. Deshalb hat sich die Eingabe von Paßworten durchgesetzt, die der Computer mit gespeicherten Referenzwerten vergleicht. Grundsätzlich gibt es jedoch drei verschiedene Arten der Authentisierung eines Benutzers:

- Wissen, zum Beispiel Eingabe einer Geheimzahl
- Besitz, zum Beispiel Schlüssel für eine Tür
- Biometrische Merkmale, zum Beispiel Fingerabdruck

Die ersten zwei Möglichkeiten haben den Nachteil, daß die zu identifizierende Person sich etwas merken oder eine Sache mitführen muß. Für die dritte Variante der Identifizierung wird dies nicht benötigt, doch ist die auszuführende Messung in den meisten Fällen technisch noch sehr aufwendig.

Der Zweck der Authentisierung ist die Überprüfung der Identität und Authentizität eines Kommunikationspartners ([Rankl, Effing 1996], S. 268). Übertragen auf die Chipkartenwelt heißt dies, daß die Chipkarte oder das Terminal feststellt, ob der Kommunikationspartner ein echtes Terminal, eine echte Chipkarte beziehungsweise ein echter Benutzer ist. Die beiden Kommunikationsteilnehmer müssen dazu ein gemeinsames Geheimnis (symmetrisches Verfahren) wissen, das mit Hilfe eines Authentisierungsverfahrens überprüft wird. Eine Gefahr dieses Verfahrens besteht darin, daß ein Geheimnis, falls es im Klartext zur Chipkarte gesendet wird, durch einen Angreifer durch Abhören sehr einfach ermittelt werden kann. Prinzipiell kann Authentisierung auch mit asymmetrischen Verfahren erfolgen.

Bei Authentisierungsverfahren unterscheidet man zwischen statischer und dynamischer Authentisierung. Bei den statischen Verfahren werden immer die gleichen, statischen Daten zur Authentisierung benutzt. Die dynamischen Verfahren hingegen sind so aufgebaut, daß sie für jede Authentisierung eine unterschiedliche Datenmenge verwenden. Im folgenden werden nun die beiden Varianten *Identifizierung durch Wissen* und *Biometrische Verfahren* näher erläutert. Da die Variante *Identifizierung durch Besitz* im wesentlichen die Tatsache beschreibt, daß der Benutzer im Besitz einer bestimmten Sache ist, wird dies hier nicht weiter vertieft. Prinzipiell sind auch Kombinationen der drei angesprochenen Varianten denkbar und sinnvoll.

4.6.1 Identifizierung durch Wissen

Die häufigste Benutzeridentifizierung ist die Eingabe einer Geheimzahl, welche allgemein als PIN bezeichnet wird. Diese wird mit einem Referenzwert verglichen und dann das Ergebnis des Vergleichs übergeordneten Systemen mitgeteilt. Üblicherweise hat die PIN vier Zeichen, die alle Dezimalzahlen zwischen „0“ und „9“ sein können. Nach einer Empfehlung der ISO-Norm 9564-1 sollte die PIN aber aus vier bis zwölf alphanumerischen Zeichen bestehen, damit die Wahrscheinlichkeit, durch bloßes Probieren die richtige PIN herauszufinden, gering ist. Die Eingabe von nicht-numerischen Zeichen ist aber an vielen Stellen aus technischen Gründen überhaupt nicht möglich, da nur eine numerische Tastatur vorhanden ist.

In vielen Fällen ist die Eingabe und Überprüfung der PIN nicht bloß eine Identifikation des Benutzers, sondern stellt gleichzeitig auch eine Willenserklärung dar. Der Benutzer erklärt sich mit Eingabe seiner PIN mit einem bestimmten Vorgang einverstanden. Als Beispiel kann man hier die Eingabe der PIN an einem Geldausgabeautomaten anführen. Zum einen authentisiert sich der Benutzer durch die Kenntnis der geheimen PIN. Zum anderen ist sie aber auch die Willenserklärung des Benutzers, daß er damit einverstanden ist, einen bestimmten Betrag ausbezahlt zu bekommen.

4.6.2 Biometrische Verfahren

Der immer häufigere Einsatz von Paßworten und PINs führt bei den Anwendern zu einem Problem. Nur sehr wenige Mensche können sich viele verschiedene vierstellige Zahlen merken. Dies ist sicherlich einer der Gründe, warum man in vielen Bereichen biometrische Verfahren favorisiert. Diese Verfahren sind nicht unbedingt schneller oder sicherer als eine PIN-Eingabe, doch für den Benutzer stellen sie eine Vereinfachung dar. Biometrische Merkmale sind nicht wie eine PIN auf andere Menschen übertragbar. Man authentisiert also die Person selber und nicht ein Geheimnis (die PIN), dessen Kenntnis eine Person mit einem Systembetreiber teilt.

Die Benutzerakzeptanz ist gerade im Zusammenhang mit der Prüfung von biometrischen Merkmalen ein wesentlicher Aspekt. Die Benutzer werden ein Verfahren umso mehr akzeptieren und bereit sein, es einzusetzen, je ähnlicher es bereits vorhandenen und bekannten Verfahren gleicht. Ein typisches Beispiel ist die eigenhändige Unterschrift, die seit Generationen zur Identifizierung und Willenserklärung in beinahe allen Kulturkreisen benutzt wird. Auch soziale Aspekte spielen eine große Rolle. Fingerabdruckverfahren werden in vielen Ländern vor allem im Bereich von Polizei und Sicherheitsbehörden eingesetzt. Dieser Hintergrund wird oft auch bei darauf beruhenden biometrischen Verfahren gesehen, worunter dann die Akzeptanz leiden kann.

Ein weiteres Akzeptanzproblem stellen medizinische und hygienische Bedenken der Benutzer dar. Das können Ängste vor einer Übertragung von Krankheiten beim optischen Abtasten des Augenhintergrundes sein oder die Befürchtung, Schaden am Auge durch den verwendeten Laser zu erleiden. Diese Ängste können das Verhalten und die Akzeptanz der Benutzer erheblich beeinflussen.

Zur Authentisierung einer Person eignen sich nicht alle biologischen Merkmale. Die folgenden Punkte müssen zumindest erfüllt sein, damit sich ein Merkmal auch sinnvoll nutzen läßt:

- Das Merkmal muß eindeutig einer bestimmten Person zuzuordnen sein, es darf nicht übertragbar sein
- Eine Veränderung des Merkmals in betrügerischer Absicht darf nicht möglich sein
- Die natürlichen Veränderungen des Merkmals im Laufe der Zeit dürfen nicht zu groß sein
- Das Merkmal muß technisch gut meßbar sein (Meßmethode, Meßdauer, Meßkosten)
- Die erzeugten Referenzdaten müssen klein sein (maximal wenige hundert Bytes)
- Die Meßmethode und das Merkmal müssen von den zukünftigen Benutzern akzeptiert werden

Im Zusammenhang mit Willenserklärungen eignen sich nicht alle biometrischen Merkmale, da es einige gibt, die auch ohne explizite Zustimmung der Person geprüft werden können.

Ein biometrisches Authentisierungsverfahren ist ein Verfahren, das auf der Grundlage von einzigartigen, individuellen und biologischen Merkmalen die Identität einer Person eindeutig verifizieren kann. Dabei unterscheidet man zwischen physiologischen und verhaltensbasierten Merkmalen. Sind die durch das biometrische Authentisierungsverfahren geprüften Merkmale direkt mit dem Körper einer Person verbunden und haben keine Abhängigkeit von bewußten Verhaltensmustern, so spricht man von physiologischen biometrischen Merkmalen. Das Gegenteil, die verhaltensbasierenden biometrischen Verfahren, benutzen als Grundlage bestimmte Merkmale, die innerhalb gewisser Grenzen bewußt veränderbar, aber doch typisch für eine Person sind.

Die folgende Tabelle beschreibt die wesentlichsten und am häufigsten behandelten physiologischen Merkmale:

Merkmal	Kurzbeschreibung
Gesicht	Das Gesicht wird mit einer Kamera im Normalbereich abgetastet. Die Daten werden entsprechend aufbereitet. Dafür braucht man leistungsfähige Computer, Fuzzy-Logik und neuronale Netze.
Netzhaut	Die Netzhaut des menschlichen Auges ist aufgrund ihrer Knoten und Verzweigungen der Blutbahnen bei jeder Person unterschiedlich. Abgetastet wird mit einem die Pupille durchdringenden Lichtstrahl im infraroten Wellenlängenbereich. Das von der Netzhaut reflektierte Licht wird von einer CCD-Kamera aufgenommen, die dann die Bilddaten an einen Analysecomputer weitergibt.
Iris	Bei diesem Verfahren wird die an der Oberfläche des Auges befindliche Iris im normaloptischen Bereich durch eine CCD-Kamera aufgenommen. Die Auswertung erfolgt etwa analog der Auswertung der Netzhaut. Bei der Abtastung kann man mehr Abstand zum messenden Gerät halten, als bei der Prüfung der Netzhaut.
Geometrie der Hand	Als Ausgangspunkt der Messungen können beispielsweise Fingerlänge, Fingerdurchmesser und Radius der Fingerkuppen verwendet werden. Die Abtastung kann technisch sehr einfach durch Infrarot-LEDs und Photodioden geschehen. Die Grundlage ist dabei die vollständige oder teilweise Abdeckung der Infrarotstrahlengänge durch die Geometrie der Hand.
Fingerabdruck	Der Daumen oder eine andere Fingerkuppe wird auf eine durchsichtige Platte gelegt und eine darunter angebrachte Kamera tastet die Hautoberfläche berührungslos ab.

Tabelle 4-2: Physiologische Merkmale

Die nicht bewußt änderbaren physiologischen Merkmale unterliegen zum Teil nur sehr geringen Veränderungen über die Zeit. So ändern sich die charakteristischen Muster von Fingerabdrücken im gesamten Leben nie, genausowenig wie die Blutgefäße der Netzhaut. Eine Ausnahme stellt dabei sicherlich das Gesicht dar, das sich natürlich prinzipiell nicht verändert, aber durch Frisur, Bart und ähnliches sehr stark verwandelt werden kann. Grundsätzlich ist zu sagen, daß bei biometrischen Merkmalen, die auf der Physiologie von erwachsenen Menschen beruhen, keine laufenden Anpassungen der Referenzmuster notwendig sind.

Die verbreitetsten verhaltensbasierenden Merkmale werden in Tabelle 4-3 beschrieben:

Merkmal	Kurzbeschreibung
Schreibrythmus	Die Eingabe von Zeichen mittels Tastatur erfolgt bei einzelnen Personen sehr unterschiedlich. Der zu identifizierende Benutzer tippt eine vorgegebene Zeichenkette auf der Tastatur ein und der daran angeschlossene Computer wertet den Schreibrythmus aus und vergleicht die Werte mit gespeicherten Referenzwerten.
Stimme	Die zu identifizierende Person spricht einen oder mehrere Sätze in ein Mikrophon. Mit den Wellenformen der gesprochenen Sätze wird eine Fourier-Analyse durchgeführt, um das für jede Person charakteristische Wellenspektrum herauszufinden. Diese vergleicht man anschließend mit einem Referenzwert.
Dynamische Unterschrift	Die bei diesem Verfahren gemessenen Parameter können beispielsweise Gestalt der Unterschrift, Beschleunigung, Geschwindigkeit, Anpreßkräfte des Stifts auf den Schreibuntergrund und die Zeitdauer sein. Zur Messung können spezielle Stifte verwendet werden oder eine für die zu messenden Parameter sensitive Schreibunterlage.

Tabelle 4-3: Verhaltensbasierende Merkmale

Verhaltensbasierende Merkmale bleiben über die Zeit hinweg bei vielen Personen nicht stabil. Beispielsweise ist die Unterschrift im Laufe des Lebens starken Veränderungen unterworfen. Diese Veränderungen treten aber in den seltensten Fällen schlagartig auf, sondern ganz allmählich und langsam. Deshalb benutzen viele Systeme adaptive Verfahren, die festgestellte Veränderungen an dem biometrischen Merkmal nach korrekter Authentisierung in das bisherige Referenzmuster übernehmen.

Die auf einem kryptographischen Algorithmus basierenden Authentisierungsverfahren teilt man noch in symmetrische und asymmetrische Verfahren ein. Im Bereich der Chipkarten werden momentan fast ausschließlich symmetrische Verfahren eingesetzt. Die asymmetrischen, also auf RSA oder ähnliche Algorithmen aufbauenden Verfahren, haben bei Chipkarten aufgrund der langsamen Ausführungsgeschwindigkeit zur Zeit keine praktische Bedeutung. Es ist aber absehbar, daß sich dies in Zukunft ändern wird.

Im folgenden werden symmetrische und asymmetrische Authentisierung am Beispiel der Authentisierung von Chipkarte und Chipkartenterminal vorgestellt. Dabei werden die technischen Randbedingungen, denen Chipkarten unterliegen, mit berücksichtigt.

Das Prinzip der Authentisierung im Chipkartenbereich basiert immer auf dem Frage-Antwort-Verfahren. Dabei stellt der eine Kommunikationspartner dem anderen eine zufällig erzeugte Frage (Challenge), dieser berechnet mit einem Algorithmus eine Antwort und sendet sie an den Fragesteller zurück (Response). Der Algorithmus ist natürlich vorzugsweise eine Verschlüsselung mit einem geheimen Schlüssel, der das gemeinsame Geheimnis der beiden Kommunikationspartner darstellt.

4.6.3 Symmetrische Authentisierung

Bei der Authentisierung unterscheidet man grundsätzlich zwischen einseitigen und gegenseitigen Verfahren. Mit der einseitigen Authentisierung vergewissert man sich der Vertrauenswürdigkeit eines Kommunikationspartners. Dazu ist es notwendig, daß beide ein gemeinsames Geheimnis besitzen, dessen Kenntnis durch Authentisierung überprüft wird. Das Geheimnis ist ein Schlüssel für einen Verschlüsselungsalgorithmus, an dem die gesamte Sicherheit des Verfahrens hängt. Wird dieser Schlüssel bekannt, dann kann sich ein Angreifer genauso authentisieren wie der echte Kommunikationspartner.

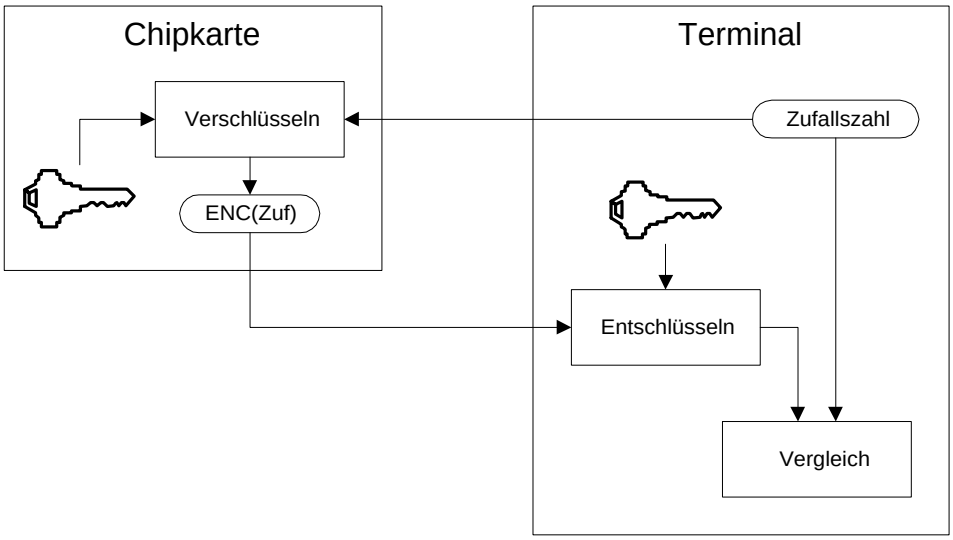


Abbildung 4-10: Einseitige, symmetrische Authentisierung der Chipkarte durch das Terminal

Das Chipkarten-Terminal generiert eine Zufallszahl und sendet diese zur Chipkarte, was man als Anfrage (Challenge) bezeichnet. Nach dem Empfang dieser Zufallszahl wird sie von der Chipkarte verschlüsselt. Der dabei verwendete Schlüssel ist nur dem Terminal und der Chipkarte bekannt. Das Ergebnis der Verschlüsselung sendet die Chipkarte zum Terminal zurück. Dies ist die Antwort (Response) auf die Anfrage (Challenge). Das Terminal führt nun mit der erhaltenen und verschlüsselten Zufallszahl eine Entschlüsselung mit dem geheimen Schlüssel aus. Danach vergleicht es das Ergebnis mit der ursprünglich an die Chipkarte gesendete Zufallszahl. Stimmt beides überein, so weiß das Terminal, daß die Chipkarte den geheimen Schlüssel kennt und schließt daraus, daß die Chipkarte authentisch ist.

Das Prinzip der gegenseitigen Authentisierung beruht auf einer zweifachen einseitigen Authentisierung. Da der Kommunikationsaufwand so gering wie möglich gehalten werden soll, gibt es ein Verfahren, bei dem zwei einseitige Authentisierungen miteinander verflochten sind.

Jede Chipkarte hat eine individuelle Kartennummer. Diese kann dazu verwendet werden, einen kartenindividuellen Authentisierungsschlüssel zu berechnen. Dazu benötigt das Terminal als erstes die Kartennummer (Abbildung 4-11). Nachdem das Terminal die Kartennummer erhalten hat, berechnet es den individuellen Kartenschlüssel für diese Chipkarte. Dann fordert es von der Chipkarte eine Zufallszahl

an und generiert ebenfalls eine Zufallszahl. Nun setzt das Terminal beide Zufallszahlen hintereinander, verschlüsselt sie mit dem geheimen Authentisierungsschlüssel und sendet den erhaltenen Schlüsseltext zur Chipkarte.

Die Chipkarte kann den erhaltenen Block entschlüsseln, da sie ebenfalls über den geheimen Authentisierungsschlüssel verfügt, und prüfen, ob die vorher an das Terminal gesendete Zufallszahl mit der zurückerhaltenen übereinstimmt. Ist dies der Fall, so weiß die Chipkarte, daß das Terminal den geheimen Schlüssel besitzt. Damit ist das Terminal gegenüber der Chipkarte authentisiert.

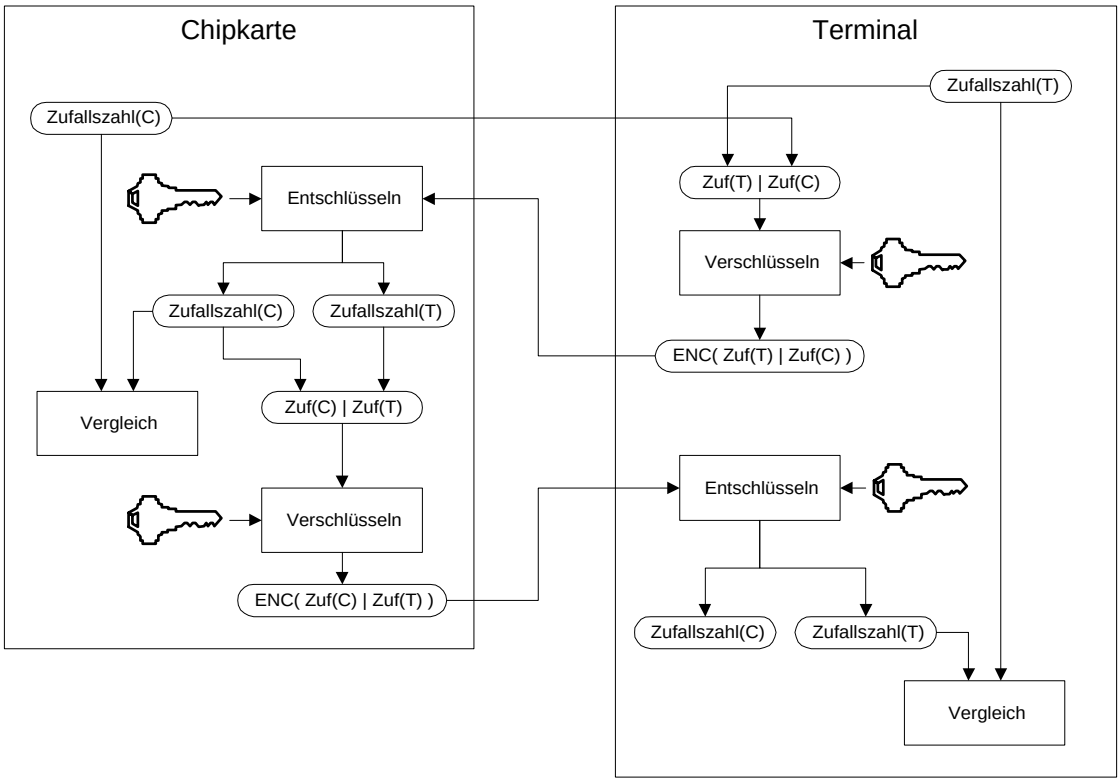


Abbildung 4-11: Gegenseitige symmetrische Authentisierung der Chipkarte durch das Terminal

Daraufhin vertauscht die Chipkarte die beiden Zufallszahlen, verschlüsselt die vertauschten Zufallszahlen mit dem geheimen Schlüssel und schickt das zum Terminal. Das Vertauschen hat den Zweck, Challenge und Response unterschiedlich zu machen. Das Terminal entschlüsselt den erhaltenen Block und vergleicht die zuvor vom Terminal an die Chipkarte gesendete Zufallszahl mit der erhaltenen. Stimmt diese mit der gesendeten überein, so ist die Chipkarte auch gegenüber dem Terminal authentisiert. Damit ist die gegenseitige Authentisierung abgeschlossen und sowohl Chipkarte als auch Terminal wissen, daß der andere jeweils authentisch ist.

4.6.4 Asymmetrische Authentisierung

Nur wenige Chipkarten-Microcontroller besitzen einen Coprozessor, mit dem RSA-Ver-/Entschlüsselung (siehe Kapitel 4.2) durchgeführt werden kann. Das liegt vor allem daran, daß dieser zusätzlichen Platz auf dem Chip benötigt und den Preis erhöht. Wegen

der unterschiedlichen Rechenleistung von Chipkarten unterscheidet man statische und dynamische, asymmetrische Authentisierung.

Bei statischen Verfahren (Abbildung 4-12) wird keine Berechnung auf der Chipkarte ausgeführt, sondern nur im Terminal. Dies ist sicherlich ein Sicherheitsnachteil, aber auch ein Kompromiß zwischen Preis und Performance der Chipkarten. Statische Verfahren arbeiten nach folgendem Prinzip: In jeder Chipkarte wird der öffentliche kartenindividuelle Schlüssel gespeichert und mit einem geheimen Schlüssel unterschrieben. Diese Unterschrift wird auch in der Chipkarte gespeichert.

Das Terminal liest den öffentlichen kartenindividuellen Schlüssel und dessen Unterschrift aus und prüft mit dem öffentlichen Schlüssel die Authentizität des kartenindividuellen Schlüssels.

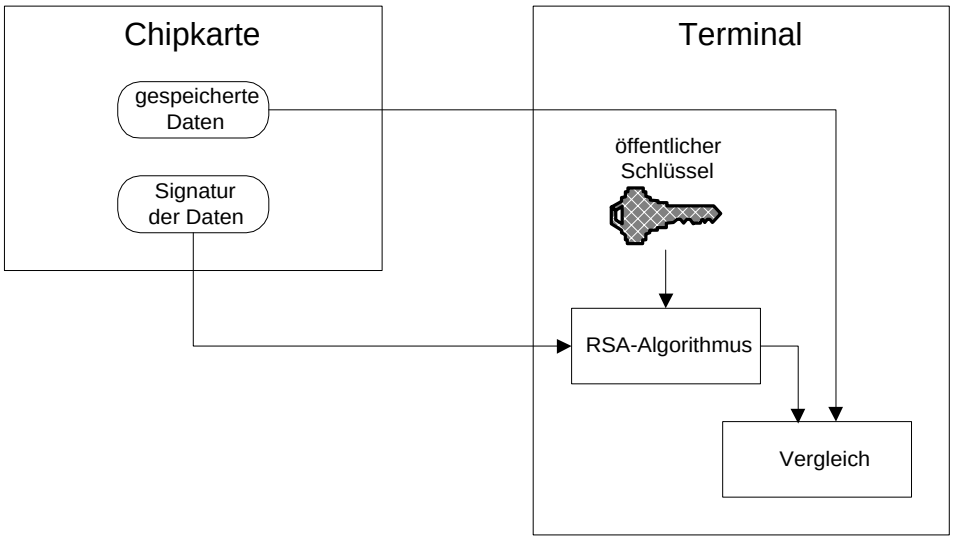


Abbildung 4-12: Einseitige, statische und asymmetrische Authentisierung der Chipkarte durch das Terminal

Ist dieser authentisch, dann liest das Terminal die eigentlichen Daten und deren Unterschrift. Das Terminal prüft die Unterschrift mit dem in der Chipkarte gespeicheten kartenindividuellen öffentlichen Schlüssel. Erst wenn diese Prüfung in Ordnung ist, hat sich die Chipkarte dem Terminal gegenüber authentisiert. Eine gegenseitige asymmetrische Authentisierung läuft äquivalent der symmetrischen gegenseitigen Authentisierung ab. Sobald die Coprozessoren für asymmetrische Kryptoalgorithmen keinen nennenswerten Preisunterschied bei Chipkarten-Microcontrollern mehr verursachen, werden statische asymmetrische Authentisierungsverfahren stark an Bedeutung verlieren und dynamische Verfahren eingesetzt werden. Man beachte, daß gemäß der Konvention von asymmetrischen Kryptoalgorithmen immer der geheime Schlüssel zum Entschlüsseln und der öffentliche Schlüssel zum Verschlüsseln verwendet wird.

Bei dynamischen asymmetrischen Authentisierungsverfahren (Abbildung 4-13) benötigt man einen Coprozessor auf dem Chip, der asymmetrische Kryptoalgorithmen ausführen kann. Analog zur symmetrischen Authentisierung generiert das Terminal eine Zufallszahl und sendet diese zur Chipkarte. Diese entschlüsselt die Zufallszahl mit dem geheimen Schlüssel und sendet das Ergebnis anschließend wieder zum Terminal. Dort

befindet sich der korrespondierende öffentliche Schlüssel, mit dem die empfangene Zufallszahl verschlüsselt wird. Ist das Ergebnis dieser Rechenoperation identisch mit der vorher zur Chipkarte gesendeten Zufallszahl, so ist die Chipkarte gegenüber dem Terminal authentisiert.

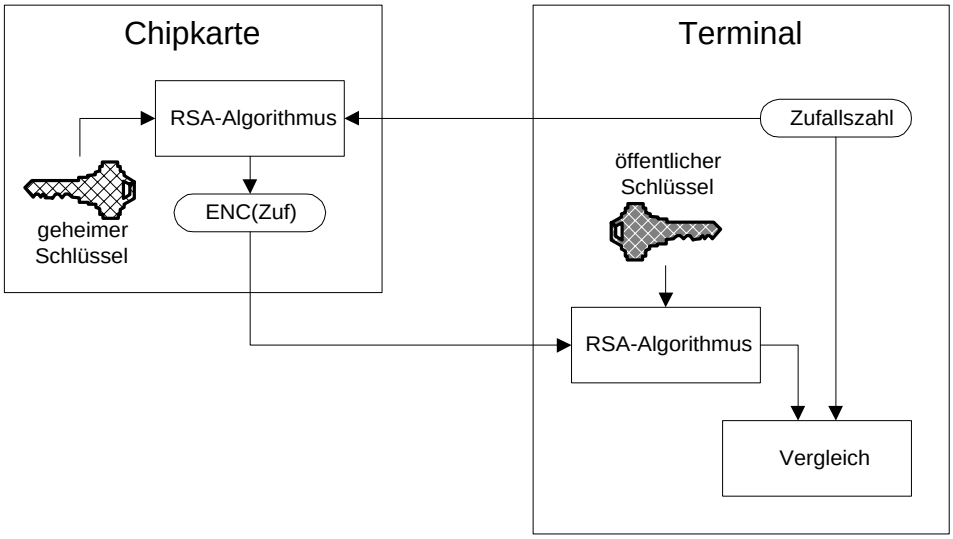


Abbildung 4-13: Einseitige, dynamische und asymmetrische Authentisierung der Chipkarte durch das Terminal

Eine gegenseitige Authentisierung von Chipkarte und Terminal wird grundsätzlich zu der oben beschriebenen einseitigen Autentisierung aufgebaut. Die zweiseitige Autentisierung erfordert aber aufgrund des höheren Datenübertragungsbedarfs und des aufwendigeren asymmetrischen Verschlüsselungsverfahrens relativ viel Zeit, so daß sie momentan sehr selten benutzt wird.

5 Allgemeine Sicherheitsmodelle

In Kapitel 2 wurde auf die Bedeutung der Sicherheit von IT-Systemen im allgemeinen und beim elektronischen Zahlungsverkehr im speziellen eingegangen. In diesem Kapitel wird die Bedeutung von Sicherheitsmodellen für die Entwicklung von IT-Systemen herausgearbeitet (siehe Kapitel 5.1) und die bekanntesten Sicherheitsmodelle werden vorgestellt und anhand der grundlegenden Sicherheitsanforderungen (siehe Kapitel 2.2) bewertet. Aus der Bewertung (siehe Kapitel 5.8) folgt eine Begründung für die Entwicklung eines spezifischen Sicherheitsmodells (siehe Kapitel 5.9).

5.1 Bedeutung von Sicherheitsmodellen

Wie schon erwähnt wurde, ist Sicherheit eine Eigenschaft, die im Gegensatz zu anderen Eigenschaften nicht verifizierbar ist. Will man zum Beispiel nachweisen, daß ein Flugzeug eine bestimmte Fluggeschwindigkeit erreichen kann, testet man das einfach. Will man dagegen nachweisen, daß ein Fluggast im Flugzeug sicher ist, kann man höchstens einige Unfälle simulieren, um zu zeigen, daß der Passagier bei diesen nicht zu Schaden kommt. Man kann jedoch keine Sicherheit nachweisen, da es immer Unfallarten geben wird, gegen die es keinen Schutz gibt. Ähnliches gilt für IT-Systeme. Sicherheitseigenschaften lassen sich nur beispielhaft an simulierten Angriffen demonstrieren. Alle Angriffe gegen ein IT-System kann man jedoch nicht simulieren, so daß die Sicherheit eines IT-Systems nicht nachweisbar ist. Aus diesem Grund möchte man die Möglichkeit haben, Sicherheitseigenschaften soweit wie möglich formal zu beweisen. Dies führt zur begründeten Benutzung von Sicherheitsmodellen als Grundlage für die Entwicklung von komplexen IT-Systemen.

Sicherheit ist eine Eigenschaft, die empirisch nicht verifizierbar ist. IT-Sicherheit (unter dem Aspekt der Vertraulichkeit) wird im allgemeinen in folgender Form gefordert: „Ein Nichtbefugter darf keine Möglichkeit haben, vertrauliche Daten zu erfahren.“ Man kann zwar für eine endliche Zahl von Angriffen demonstrieren, daß das IT-System vor diesen getesteten Angriffen geschützt ist. Jedoch kann es noch weitere Angriffe geben, vor denen das System nicht geschützt ist. Um Sicherheitseigenschaften so weit wie möglich formal beschreiben zu können, nutzt man Sicherheitsmodelle. Mit ihnen können verbale Sicherheitsanforderungen in ein formales Modell abgebildet werden. Dieses kann dann in ein IT-System implementiert werden. Jedoch erweist sich ein vollständiger Nachweis für die Sicherheit eines IT-Systems als ausgesprochen schwierig.

Obwohl allgemein anerkannt ist, daß formal nicht unbedingt gleich sicher bedeutet, liegt der Wert von formalen Sicherheitsmodellen doch gerade in der Tatsache, daß gewisse Eigenschaften zugesichert werden können.

Bei der Entwicklung von IT-Systemen soll eine unternehmensspezifische Sicherheitspolitik realisiert werden. Eine Sicherheitspolitik verkörpert die Ziele eines Unternehmens oder einer Organisation, die sicherheitsrelevant sind. Diese Sicherheitspolitik ist für jedes Unternehmen oder für jede Organisation unterschiedlich. Die Sicherheits-

politik legt fest, welchen Personen, die mit dem IT-System arbeiten, welches Vertrauen entgegen gebracht wird. Sie definiert weiterhin, wer auf welche Daten und Informationen wann, wie und mit welchen Auswirkungen zugreifen darf.

Grundlage für die Umsetzung einer Sicherheitspolitik ist ein Sicherheitsmodell, das formalen oder semiformalen Charakter hat. Ein Sicherheitsmodell kann als eine abstrakte Beschreibung eines IT-Systems angesehen werden, um dessen Sicherheitseigenschaften analysieren zu können.

Bei der allgemeinen Begriffsdefinition „Modell“ muß zwischen zwei wesentlichen Verwendungsarten unterschieden werden. Im Brockhaus findet sich folgende Definition [Brockhaus 1991]:

- Modell als Abbild:

  Ein Abbild der Realität unter Hervorhebung für wesentlich erachteter Eigenschaften und Außerachtlassen als nebensächlich angesehener Aspekte.
  In diesem Sinne ist ein Modell eine Abstraktion.

- Modell als mathematische Logik (Modelltheorie):

  Hier ist ein Modell eine mathematische Struktur, welche die Axiome eines widerspruchsfreien vorgegebenen Axiomensystems erfüllt. Insbesondere folgt aus der Existenz eines Modells, daß das Axiomensystem widerspruchsfrei ist.
  Ein solches Modell ist eine Konkretisierung.

Die Beschreibungssprache von Sicherheitsmodellen erfolgt meist in mathematisch-logischer Sprache. Man spricht dann von einem formalen Sicherheitsmodell (siehe Abbildung 5-1).

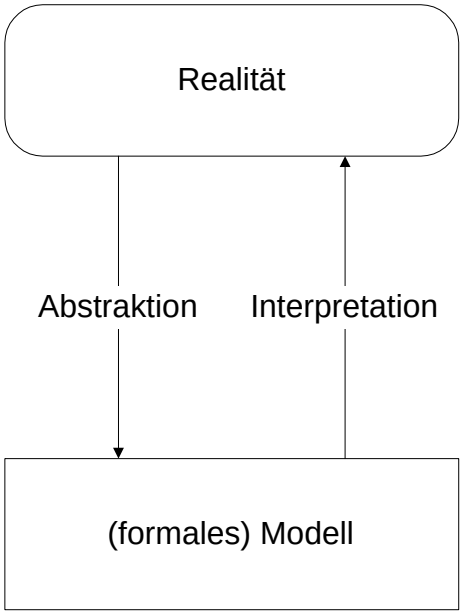


Abbildung 5-1: Realität und Modell
Quelle: [Kessler, Mund 1993], S 9

Dem Modell als Abbild zur Beschreibung der Realität liegen drei Funktionen zugrunde:

- Bildung von Begriffen der Realität
- Erklärung von bereits beobachteten Phänomenen
- Voraussage von zukünftigen Phänomenen

Wird das Modell als Abstraktion genutzt, schafft man Klarheit in komplexen Situationen, durch Hervorhebung relevanter Aspekte und durch Unterdrückung irrelevanter Aspekte. Andererseits besteht die Gefahr, daß durch die Verkürzung eventuell wichtige Aspekte vernachlässigt werden, die fälschlicherweise als irrelevant eingestuft wurden.

In der Regel besteht ein formales Sicherheitsmodell jedoch nicht ausschließlich aus mathematischen Beschreibungen, sondern enthält immer einen informellen Beschreibungsteil. Vorteile eines formalen Sicherheitsmodells liegen in der Präzision und Eindeutigkeit der Sprache, in der Möglichkeit, Beweise zu führen und in der Unterstützung der formalen Spezifikation und Implementierung.

Obwohl bisher viele unterschiedliche Sicherheitsmodelle entwickelt wurden, gibt es keine einheitliche Definition eines Sicherheitsmodells. Ein Versuch einer Definition wurde 1993 im Rahmen einer Studie über Sicherheitsmodelle [Kessler, Mund 1993] begonnen, auf die sich im folgenden bezogen wird.

> *„Ein Sicherheitsmodell ist eine abstrakte Beschreibung der nach der zugrundeliegenden Sicherheitspolitik für wesentlich gehaltene Aspekte der Sicherheit eines IT-Systems, wobei die als nicht sicherheitsrelevant geltenden Aspekte unterdrückt werden.“ ([Kessler, Mund 1993], S. 11)*

Ein Sicherheitsmodell beschreibt also die Sicherheit eines IT-Systems in einer abstrakten Form. Dabei werden nur die für wesentlich erachteten Aspekte berücksichtigt. Die Aspekte, die für die Sicherheit des Systems für irrelevant gehalten werden, werden bei der Beschreibung vernachlässigt.

Für das Modell an sich kann gegebenenfalls ein formaler Sicherheitsbeweis durchgeführt werden. Jedoch ist das Modell nur eine Abstraktion der Wirklichkeit, bei der ausschließlich die für wesentlich erachteten Aspekte berücksichtigt werden. So kann schon bei der Umsetzung der Wirklichkeit in ein Modell kein vollständiger Beweis für die Sicherheit des Systems erbracht werden. Hier kann lediglich argumentiert werden. Ferner ist auch die Implementierung des formalen Modells eine Frage der Softwarekorrektheit und somit eine weitere Lücke in der Beweiskette, da die Korrektheit von Software heutzutage für die wenigsten Programme durchgeführt werden kann.

Aufgrund unterschiedlichster Anwendungssituationen und Sicherheitspolitiken kann es kein universelles Sicherheitsmodell geben, das sämtliche Sicherheitseigenschaften erfüllt, da es sich widersprechende und sich ausschließende Sicherheitseigenschaften gibt. Man benötigt daher viele anwendungsbezogene Sicherheitsmodelle, die entsprechend der Sicherheitsanforderungen individuell entwickelt werden.

Folgende Sicherheitsmodelle und deren Kombinationen wurden bisher entwickelt:

Name	Jahr	Beschreibung
Bell-LaPadula-Modell	1973-76	Vertraulichkeit
Denning-Modell	1976	Vertraulichkeit
Biba-Modell	1977	Integrität
Goguen-Meseguer-Modell	1982	Nichtbeeinflussung
Lipner-Modell	1982	Vertraulichkeit und Integrität
Clark-Wilson-Modell	1987	Integrität durch wohlgeformte Transaktion und Pflichtentrennung
Chinese Wall-Modell	1989	Kombination von wahlfreier und festgelegter Zugriffskontrolle
Terry-Wiseman-Modell	1989	Vertraulichkeit und Integrität
Generalised Framework for Access Control (GFAC)	1990-92	Genereller Ansatz zur Integration verschiedener Sicherheitsmodelle
Rollenbasiertes Zugriffsmodell	1992-95	Rollenbasierter Zugriff
Telekooperationsmodell	1993	Pflichtentrennung, offene Systeme
Formales Datenschutzmodell	1994-97	Datenzugriff nach Zweckbindung und Erforderlichkeit

Tabelle 5-1: Übersicht über Sicherheitsmodelle

Trotz aller genannter Probleme mit Sicherheitsmodellen, sind sie für die Entwicklung sicherer IT-Systeme unersetzlich. Sie erfassen präzise die verbal gegebenen Sicherheitsanforderungen, können diese auf ihre Konsistenz prüfen und die Realisierung vorbereiten. Auch wenn kein vollständiger Beweis für die Sicherheit eines Systems erbracht werden kann, so kann aber das Vertrauen in das System erhöht werden. Ferner werden bei der formalen Analyse die Schwachstellen des Systems herausgearbeitet. So steht genau fest, wo vertrauenswürdige Personen eingesetzt werden müssen.

In den nachfolgenden Unterkapiteln werden neben den klassischen Zugriffskontrollkonzepten die Sicherheitsmodelle von Bell und LaPadula, von Clark und Wilson, das Telekooperationsmodell, das formale Datenschutzmodell und das rollenbasierte Modell erläutert und in einem abschließenden Vergleich bewertet.

Das Vertraulichkeitsmodell von Bell und LaPadula wird näher beschrieben, weil es das erste Sicherheitsmodell darstellt, das den Aspekt der Vertraulichkeit behandelt und somit historische Bedeutung hat. Dies gilt ebenso für das Clark-Wilson-Modell zum Schutz der Integrität. In diesem Modell werden die kommerziellen Sicherheitsanforderungen berücksichtigt, im Gegensatz zu den militärischen Sicherheitsanforderungen aus dem Bell-LaPadula-Modell.

Das Telekooperationsmodell und das rollenbasierte Sicherheitsmodell verwenden Rollen, die für den Anwendungsbereich der multifunktionalen Chipkarten im elektronischen Zahlungsverkehr von großer Bedeutung sind. Die Definition von Rollen ist in beiden Modellen zwar unterschiedlich, jedoch sind beide Modelle wichtig für die Anforderungen, die in Kapitel 2.2 herausgearbeitet wurden.

Von gleichem Interesse ist das formale Datenschutzmodell, das die Datenschutzgrundsätze der Zweckbindung und der Erforderlichkeit auf Aufgaben, die ein Benutzer ausführen darf, anwendet. Aus diesen Gründen werden im folgenden diese Sicherheitsmodelle näher erläutert und im Anschluß bewertet.

5.2 Klassische Zugriffskontrollmodelle

Klassische Zugriffskontrolle, wie sie in [Gasser 1988] definiert wurde, teilt sich in drei Phasen auf:

- Beschreibung, welches Subjekt Zugriff auf welches Objekt hat (Autorisierung)
- Festlegen der Zugriffsrechte als Kombination der einzelnen Zugriffsmodi (lesen, schreiben, ausführen, ...)
- Durchsetzen der Zugriffsrechte

Prinzipiell kann man zwischen verschiedenen Ansätzen der Zugriffskontrolle unterscheiden, wobei Discretionary Access Control (DAC) und Mandatory Access Control (MAC) die bekanntesten Zugriffskontrollarten sind.

Discretionary Access Control (DAC) wird in [TCSEC 1985] definiert und erlaubt Systembenutzern nach eigenem Ermessen, Zugriffrechte an andere Benutzer des Systems weiterzugeben (oder zu entziehen), die sich auf Objekte unter ihrer Kontrolle beziehen. Ein Subjekt mit einer bestimmten Zugriffserlaubnis darf genau diese Erlaubnis an andere Subjekte weitergeben. Die verbreiteste Art der Implementierung ist die Benutzung von Zugriffslisten (Access Control Lists - ACL). In großen Systemen können ACL jedoch sehr umständlich in der Handhabung werden [Ferraiolo, Gilbert, Lynch 1993].

Der System-Administrator muß bei jedem neuen Benutzer oder jeder Änderung der Verantwortlichkeiten die entsprechenden Zugriffsrechte im System festlegen. Verläßt ein Benutzer die Organisation, muß der Benutzer aus jeder ACL herausgelöscht werden. DAC hat einen wesentlichen Nachteil bezüglich trojanischer Pferde. Da der Benutzer nach eigenem Ermessen seine Zugriffsrechte weitergeben darf, kann nicht verhindert werden, daß maliziöse Programme im Namen des Benutzers Zugriffsrechte verändern, weitergeben oder löschen.

Bei Mandatory Access Control (MAC) können die Zugriffsrechte nicht von Benutzern verändert werden. Es gibt fest vorgegebene Sicherheitsattribute, anhand derer das System entscheidet, ob ein Subjekt Zugriff auf ein Objekt haben darf. Die Sicherheitsattribute werden entweder von einer zentralen Person, dem Sicherheits-

administrator oder automatisch vom Betriebssystem durch strikte Regeln zugewiesen. Meist wird MAC zusätzlich mit DAC verwendet, um striktere Zugriffe zu regeln und unterschiedliche Ziele zu erreichen. Soll zum Beispiel verhindert werden, daß Benutzer auf andere Benutzerdateien zugreifen können, genügt Discretionary Access Control. Mit Mandatory Access Control kann verhindert werden, daß ein Programm (in Form eines trojanischen Pferdes) eine Benutzerdatei freigibt, jedoch kann nicht verhindert werden, daß ein Benutzer freiwillig seine Dateien freigibt.

Reine MAC-Verfahren können nicht isoliert, ohne Bezug zu multilevel security, betrachtet werden. Multilevel security (MLS) ist eine mathematische Beschreibung einer militärischen Sicherheitspolitik, die so definiert ist, daß sie implementiert werden kann. Das erste mathematische Modell eines multilevel sicheren Computersystems, bekannt als das Bell-LaPadula-Modell, wird im folgenden Kapitel erläutert.

5.3 Vertraulichkeitsmodell

Das Bell-LaPadula-Modell ist ein formales, hierarchisches Zugriffskontrollmodell zum Schutz der Vertraulichkeit. Es basiert auf Sicherheitsanforderungen aus dem militärischem Bereich und wurde von 1973 bis 1976 von D. Elliott Bell und Leonard J. LaPadula entwickelt [Bell, LaPadula 1973/1976]. In dieser Arbeit soll eine allgemeine Beschreibung des Modells erfolgen [Kessler, Mund 1993], unabhängig von vielen Veränderungen, die im Laufe der Jahre von den Entwicklern selbst vorgenommen wurden.

Die Sicherheitspolitik setzt voraus, daß die Mengen der beteiligten Elemente (Subjekte und Objekte) hierarchisch strukturiert sind, und fordert, daß Informationen von einem Element immer nur zu gleich- oder höherwertig klassifizierten Elementen fließen darf.

5.3.1 Elemente

Im Bell-LaPadula-Modell gibt es zwei Elementarten:

- Die Menge S der (aktiven) Subjekte s_i. Dies sind Benutzer oder Prozesse, die im Auftrag des Benutzers laufen. Innerhalb der Menge der Subjekte gibt es eine Teilmenge TS der vertrauenswürdigen Subjekte (trusted subjects), wobei sich die Vertrauenswürdigkeit nur auf die *-Property bezieht (siehe Kapitel 5.3.4 Axiom 2). Die Komplementärmenge wird mit S’ = S \ TS bezeichnet.
- Die Menge O der (passiven) Objekte o_j. Hierzu zählen alle im System vorhandenen Dateien, die jedoch auch ausführbare Programme sein können.

Die beiden Mengen der Subjekte und Objekte sind nicht disjunkt. Bei einem Zustandsübergang ist ein Element jedoch eindeutig ein Subjekt oder ein Objekt. Subjekte haben kein Gedächtnis. Wenn ein Subjekt Information speichern möchte, muß es diese als Objekt ablegen.

5.3.2 Sicherheitsklassen

Es gibt zwei Attribute für Subjekte und Objekte:

- Klassifikationen: Eine bezüglich „≤“ totalgeordnete Menge C dient dazu, die Elemente einzustufen, zum Beispiel in öffentlich, vertraulich, geheim und streng geheim. Bei einem Subjekt bedeutet diese Klassifikation, daß es eine Zugriffsberechtigung für Objekte bis zu der jeweiligen Stufe besitzt.
- Zuständigkeitsbereiche: Die Objekte werden je nach Inhalt in verschiedene Kategorien eingeteilt. Die Subjekte bekommen Zugang zu Bereichen, die zu ihrem Aufgabengebiet gehören. Die Potenzmenge P(K) der Kategorienmenge K ist durch die Teilmengenbeziehung $\subseteq$ (partiell) geordnet.

Eine Sicherheitsklasse oder ein Sicherheitsniveau ist dann ein Paar (c, k), das aus einer Klassifikation $c \in C$ und einer Teilmenge $k \subseteq K$ besteht. Die Menge $L = C \times P(K)$ ist (partiell) geordnet durch die Dominanzrelation:

$$\forall\, c_i \in C,\ k_i \in P(K):$$

$$(c_1, k_1) \leq (c_2, k_2) \Leftrightarrow c_1 \leq c_2 \wedge k_1 \subseteq k_2$$

Die Menge L besitzt ein kleinstes Element (low) und ein größtes Element (high). Die Mengen S und O werden durch folgende Sicherheitsniveaufunktionen strukturiert:

- Objektlevel f_O: O-> L
- maximales Subjektlevel f_S: S -> L
- aktuelles Subjektlevel f_C: S’ -> L mit $f_C(s_i) \leq f_S(s_i)$, $\forall\, s_i \in S'$

5.3.3 Zugriffsarten

Im Bell-LaPadula-Modell wurden grundsätzlich zwei Elementaroperationen auf Objekten definiert, „beobachten“ und „ändern“. Werden diese beiden miteinander kombiniert, ergeben sich daraus vier Zugriffsarten:

- Lesen (read) - Beobachten eines Objektes, ohne es zu ändern.
- Anhängen (append) - Ändern eines Objektes, ohne es zu beobachten.
- Schreiben (write) - Beobachten und Ändern eines Objektes.
- Ausführen (execute) - Weder Beobachten, noch Ändern eines Objektes.

In diesem Modell werden diese Zugriffsarten (accesses) in einer Menge A = {r, a, w, e} zusammengefaßt und die Zugriffsrechte in einer Matrix M gespeichert, wobei $M(i, j) \in P(A)$ anzeigt, welche Zugriffsrechte das Subjekt s_i auf das Objekt o_j hat.

5.3.4 Systemzustand und Zustandsänderungen

Im Bell-LaPadula-Modell wird ein Systemzustand $v \in V$ durch ein Quadrupel (b, M, f, H) mit folgenden Komponenten beschrieben:

- Die Menge b der aktuellen Zugriffsrechte. Jedes Element dieser Menge besteht aus einem Tripel (s_i, o_j, x) mit Subjekt $s_i \in S$, Objekt $o_j \in O$ und Zugriffsart $x \in A$.
- Die Zugriffskontrollmatrix M.
- Die Levelfunktion $f = (f_S, f_O, f_C) : S \times O \times S' \rightarrow L^3$
- Die Objektwaldstruktur H.

Als Objektwaldstruktur wird eine hierarchische Struktur der Objekte bezeichnet, die im graphentheoretischen Sinn einen Wald ergibt. Formal ist diese Struktur eine Hierarchieabbildung $H : O \rightarrow P(O)$, in der der Graph auf O mit gerichteten Kanten von jedem Objekt zu seinem unmittelbar untergeordneten Objekt zeigt. Dies bedeutet, daß der Lesezugriff eines Subjektes auf ein Objekt nur dann gewährt werden kann, wenn das Subjekt das Leserecht für alle Objekte hat, die auf dem Weg von der Wurzel bis zum gewünschten Objekt liegen. Ebenso kann ein Subjekt die Einträge in M für ein Objekt o_j nur dann ändern, wenn es die Schreiberlaubnis für das darüberliegende Objekt $h(o_j)$ besitzt.

Ein Systemzustand gilt dann als sicher, wenn er die folgenden drei Axiome erfüllt:

Axiom 1: Simple Security (ss-)Property (No read up)

Hat ein Subjekt beobachtenden Zugriff auf ein Objekt, so dominiert der Subjektlevel den Objektlevel:

$$\forall s_i \in S, o_j \in O:$$

$$[(s_i, o_j, r) \in b \vee (s_i, o_j, w) \in b] \Rightarrow f_S(s_i) \geq f_O(o_j)$$

Diese Regel verhindert jedoch nicht einen unerwünschten Informationsfluß. Kann zum Beispiel ein Subjekt ein Objekt mit hohem Sicherheitsniveau beobachten und gleichzeitig ein Objekt mit niedrigem Sicherheitsniveau ändern, so kann es Daten von dem höher klassifizierten zum niedrig klassifizierten Objekt kopieren. Danach können diese Daten auch von Subjekten gelesen werden, die keine hohe Ermächtigung haben. Die Pfeile in Abbildung 5-2 geben die Richtung des Informationsflusses an.

Um diesen Informationsfluß aus Abbildung 5-2 zu verhindern, wurde ein weiteres Axiom eingeführt.

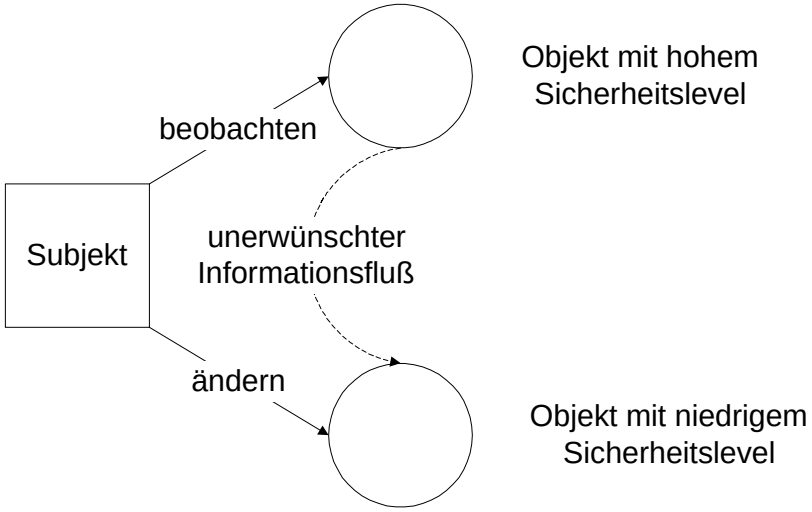


Abbildung 5-2: Unerwünschter indirekter Informationsfluß
Quelle: [Kessler, Mund 1993], S. 26

Axiom 2: Star (*-) Property

Für alle nicht vertrauenswürdigen Subjekte $s_i \in S'$ und alle Objekte $o_j \in O$ gilt:

$$(s_i, o_j, a) \in b \Rightarrow f_C(s_i) \leq f_O(o_j)$$

$$(s_i, o_j, w) \in b \Rightarrow f_C(s_i) = f_O(o_j)$$

$$(s_i, o_j, r) \in b \Rightarrow f_C(s_i) \geq f_O(o_j)$$

Die ersten zwei Bedingungen beschreiben ein sogenanntes „no write down“, während die dritte Bedingung dafür sorgt, daß die ss-Property bezüglich des aktuellen Subjektlevels eingehalten wird.

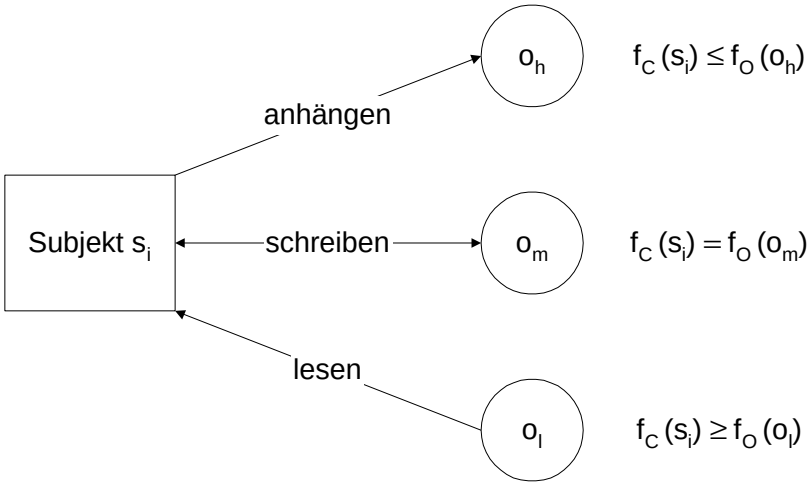


Abbildung 5-3: Zugriffsmöglichkeiten im Bell-LaPadula-Modell
Quelle: [Kessler, Mund 1993], S. 27

Abbildung 5-3 verdeutlicht die Zugriffsmöglichkeiten, die durch die *-Property erlaubt sind.

Das letzte Axiom definiert zusätzlich die DAC Zugriffskontrolle.

Axiom 3: Discretionary Security (ds-)Property

Wenn das Zugriffstripel in der Menge der aktuellen Zugriffsrechte enthalten ist, dann ist die Zugriffsart auch an der zugehörigen Stelle in der Zugriffsmatrix eingetragen.

$$\forall s_i \in S, o_j \in O, x \in A: (s_i, o_j, x) \in b \Rightarrow x \in M(i, j)$$

Ein System ändert seinen Zustand dadurch, daß aufgrund einer Anfrage mittels einer Regel eine Entscheidung getroffen wird und die Zustandsvariablen entsprechend geändert werden. Für eine Zustandsänderung wird Folgendes definiert:

- Die Menge R (Request) der möglichen Anfragen.
- Die Menge D (Decision) der möglichen Entscheidungen.
- Die Menge $W \subset R \times D \times V \times V$ der Zustandsübergänge. Ein Zustandsübergang (r, d, v*, v) liefert ausgehend von einem Zustand $v = (b, M, f, H) \in V$ und einer Anfrage $r \in R$ mittels einer Regelmenge eine Entscheidung $d \in D$ und damit einen neuen Zustand v* = (b*, M*, f*, H*).

5.3.5 „Sicheres“ System

Um ein System zu beschreiben benötigt man:

- eine abzählbare Menge T von durchnummerierten Zeitpunkten, zu denen das System einen Zustand hat, mit dem ausgezeichneten Zeitbeginn $0 \in T$ (üblicherweise T = {0, 1, 2, ..., t, ...})
- die Menge $X = R^T$ der Anfragesequenzen $x = (x_t)_{t \in T}$
- die Menge $Y = D^T$ der Entscheidungssequenzen $y = (y_t)_{t \in T}$
- die Menge $Z = V^T$ der Zustandssequenzen $z = (z_t)_{t \in T}$
- einen Anfangszustand z_0 (üblicherweise $z_0 = (0, M, f, H)$)

Ein System $\Sigma (R, D, W, z_0) \subset X \times Y \times Z$ heißt sicher, wenn jede Ausprägung des Systems sicher ist. Eine Ausprägung (x, y, z) des Systems heißt sicher, wenn die Zustandsfolge z sicher ist. Eine Zustandsfolge $z = (z_t)_{t \in T}$ heißt sicher, wenn jeder enthaltene Zustand z_t sicher ist.

Die Bell-LaPadula-Sicherheitseigenschaften (BLP) eines Systems lassen sich durch folgendes Theorem beschrieben:

Basic Security Theorem:

Ein System Σ (R, D, W, z_0) ist BLP-sicher, wenn z_0 ein BLP-sicherer Anfangszustand ist und jeder mögliche Übergang (r, d, (b*, M*, f*, H*), (b, M, f, H)) $\in$ W den folgenden Bedingungen genügt:

1) Verträglichkeit mit der ss-Property:

   (a) Jedes aktuelle Zugriffsrecht $(s_i, o_j, x) \in b^* \setminus b$ genügt der ss-Property bezüglich f*, d.h.

   $$x \in \{r, w\} \Rightarrow f^*_S(s_i) \geq f^*_O(o_j)$$

   (b) Jedes aktuelle Zugriffsrecht $(s_i, o_j, x) \in b$, welches nicht der ss-Property bezüglich f* genügt, ist nicht in b*

2) Verträglichkeit mit der *-Property:

   (a) Jedes aktuelle Zugriffsrecht $(s_i, o_j, x) \in b^* \setminus b$ mit $s_i \in S'$ genügt der *-Property bezüglich f*, d.h.

   $$x = a \Rightarrow f^*_S(s_i) \leq f^*_O(o_j)$$
   $$x = w \Rightarrow f^*_S(s_i) = f^*_O(o_j)$$
   $$x = r \Rightarrow f^*_S(s_i) \geq f^*_O(o_j)$$

   (b) Jedes aktuelle Zugriffsrecht $(s_i, o_j, x) \in b^* \setminus b$ mit $s_i \in S'$, welches nicht der *-Property bezüglich f* genügt, ist nicht in b*

3) Verträglichkeit mit der ds-Property:

   (a) $(s_i, o_j, x) \in b^* \setminus b \Rightarrow x \in M^*(i, j)$

   (b) $(s_i, o_j, x) \in b \wedge x \notin M^*(i, j) \Rightarrow (s_i, o_j, x) \notin b^*$

Für die Verifikation des Bell-LaPadula-Modells wird auf die Originalliteratur verwiesen [Bell, LaPadula 1973/1976]. Das Modell von Bell und LaPadula hat großen internationalen Einfluß wegen seiner frühen Veröffentlichung und spiegelt sich auch in den Sicherheitskonzepten des amerikanischen „Orange Book“ [DoD 1985] wider. Die Autoren sehen ihr Modell als ein Gerüst an, in dem weitere anwendungsbezogene Sicherheitsanforderungen integriert werden können.

5.4 Integritätsmodell

Im Jahre 1987 wurde von David D. Clark und David R. Wilson ein Integritätsmodell vorgestellt [Clark, Wilson 1987], das im Gegensatz zum Bell-LaPadula-Modell die kommerziellen Sicherheitsanforderungen definiert. Es ist ein semiformales, nichthierarchisches Zugriffskontrollmodell zum Schutz der Integrität der Daten für kommerzielle Anwendungen (Clark-Wilson-Modell).

Hauptziel dieses Modells ist der Schutz der Daten vor beabsichtigtem Fälschen und unbeabsichtigten Fehlern. Vertraulichkeit ist dagegen kaum oder gar nicht gefordert. Die Datenintegrität bedeutet in diesem Modell interne und externe Konsistenz der Daten. Interne Konsistenz bedeutet, daß die Daten innerhalb des IT-Systems keine Inkonsistenzen aufweisen und externe Konsistenz beschreibt den Einklang der Daten mit der Außenwelt. Die Mechanismen zum Schutz der Integrität der Daten wurden von Clark und Wilson in zwei Kategorien geteilt:

- **Wohlgeformte Transaktionen (well formed transactions)**
  Ein Benutzer darf die Daten nicht beliebig, sondern nur nach festgelegten Regeln bearbeiten. Jede Datenmenge darf nur durch eine ausgewählte Menge von Programmen bearbeitet werden. Dadurch wird die interne Konsistenz der Daten gesichert. Es wird jedoch keine Aussage darüber gemacht, ob die Daten auch konsistent mit der Wirklichkeit sind.

- **Pflichtentrennung (separation of duty)**
  Es darf kein Datum eingetragen werden, das nicht mit der Wirklichkeit übereinstimmt. Deshalb wird die Verantwortung für jedes Datum auf mehrere Personen übertragen. Eine Operation wird in mehrere Teile aufgeteilt, die von verschiedenen Personen ausgeführt werden. Jeder Benutzer darf nur eine bestimmte Menge von Programmen benutzen. Mit diesem Mechanismus wird die externe Konsistenz der Daten gesichert und zusätzlich auch die interne Konsistenz.

  Pflichtentrennung bei wohlgeformten Transaktionen kann in der Form erfolgen, daß bei einer Änderung der Menge der Transaktionen die Autorisierung und Ausführung von zwei verschiedenen Personen durchgeführt werden muß. Wichtig ist dabei, daß die ausgewählten Personen unterschiedliche Motive und Ziele haben, so daß die Pflichtentrennung nicht durch eine Verschwörung unterlaufen werden kann [Clark, Wilson 1988].

Die Durchsetzung der beiden Prinzipien wohlgeformter Transaktionen und Pflichtentrennung erfolgt mandatorisch. Ein Benutzer allein kann also weder die Menge der für eine Datei zulässigen Programme ändern, noch die für ein Programm zugelassenen Benutzer beeinflussen.

Ziel des Modells ist es, eine Übereinstimmung der internen und externen Konsistenz der Daten zu erreichen. Dies erreicht man, indem regelmäßige Prüfungen durchgeführt werden, nachdem das System aus einem integeren Zustand gestartet wurde.

5.4.1 Elemente

Die Elemente des Integritätsmodells von Clark und Wilson sind:

- Benutzer, identifiziert durch seine UserID
- Sicherheitsadministrator

- Kontrollierte Daten (constrained data items, CDI)
- Unkontrollierte Daten (unconstrained data items, UDI)

Für kontrollierte Daten ist die interne und externe Konsistenz gewährleistet und wird vom System entsprechend kontrolliert. Unkontrollierte Daten sind meist neu einzugebende Daten, die bisher nur der Kontrolle des Benutzers unterliegen. Sie können in Kontrollierte Daten überführt werden. Ist das nicht möglich, können sie nicht als neue Informationen in das System aufgenommen werden.

5.4.2 Prozeduren

Es gibt zwei Gruppen von Prozeduren:

- Transformations-Prozeduren (TP):
  Sie entsprechen den wohlgeformten Transaktionen und sind ausschließlich interne Prozeduren. Sie überführen das System von einem gültigen Zustand in einen anderen gültigen Zustand.
- Integritätsverifikations-Prozeduren (IVP):
  Sie kontrollieren die CDI hinsichtlich interner und externer Konsistenz zu einem bestimmten Zeitpunkt.

Wenn alle Kontrollierten Daten den anwendungsspezifischen Integritätsanforderungen entsprechen, befindet sich das System in einem gültigen Zustand. Die Integritätsverifikations-Prozeduren prüfen, ob ein Zustand ein gültiger Zustand ist. Wird in einem gültigen Zustand eine Transformations-Prozedur ausgeführt, befindet sich das System anschließend wieder in einem gültigen Zustand. Dies gilt nicht während der Zeit der Ausführung. Deshalb darf nie mehr als eine Transformations-Prozedur gleichzeitig ausgeführt werden.

5.4.3 Regeln

Die Regeln teilen sich in zwei Gruppen auf, eine für die Zertifizierung und eine für die Durchführung. Die Transformations-Prozeduren können nicht vom System selbst überprüft werden, sondern nur vom Sicherheitsadministrator.

- Zertifizierungsregeln C_i (certification):
  Sie werden vom Sicherheitsadministrator angewendet und realisieren die stark anwendungsabhängigen Integritätsanforderungen.
- Durchführungsregeln E_i (enforcement):
  Sie werden vom IT-System angewendet und entsprechen anwendungsunabhängigen Sicherheitsmaßnahmen.

Von Clark und Wilson werden insgesamt neun Regeln (C_i, E_i) definiert:

- Kontrolle der internen und externen Konsistenz der Kontrollierten Daten:

  C_1: Qualität der Integritätsverifikations-Prozeduren
  Wenn eine Integritätsverifikations-Prozedur zu dem Ergebnis kommt, daß sich alle Kontrollierten Daten in einem gültigen Zustand befinden, muß dies auch stimmen.

- Sicherung der internen Konsistenz der Kontrollierten Daten:

  C_2: Qualität der Transformations-Prozeduren
  Die Transformations-Prozeduren müssen als zuverlässig zertifiziert sein, d.h. sie führen gültige Daten in gültige Daten über. Der Sicherheitsadministrator muß für jede TP_i eine Liste (CDI_a, CDI_b,...) von Kontrollierten Daten angeben, bei denen er nachgewiesen hat, daß die Gültigkeit dieser Kontrollierten Daten durch die Anwendung der Transformations-Prozeduren nicht beeinträchtigt wird.

  E_1: Einhaltung der Listen
  Das System muß überwachen, daß die Kontrollierten Daten nur von solchen Transformations-Prozeduren bearbeitet werden, in deren Liste sie stehen.

- Sicherung der externen Konsistenz:

  C_3: Unterstützung der Pflichtentrennung
  Die Umsetzung der Pflichtentrennung erfordert, daß ein Benutzer nur bestimmte Transformations-Prozeduren ausführen darf und diese auch nicht auf alle Kontrollierten Daten anwenden darf, für die diese Transformations-Prozeduren zertifiziert sind. Es werden also Listen der Form (UserID, TP_i, (CDI_a, CDI_b,...)) zertifiziert, die angeben, welche Kontrollierten Daten ein bestimmter Benutzer mit einem bestimmten Programm bearbeiten darf.

  Diese Listen kann man in einer Matrix zusammenfassen, deren Zeilen aus Benutzern und Spalten aus Transformations-Prozeduren bestehen und deren Einträge Teilmengen der CDI_i aus der Liste des jeweiligen Transformations-Prozeduren sind. Falls der Benutzer diese Transformations-Prozedur nicht benutzen darf, ist das Matrixelement eine leere Menge.

  E_2: Einhaltung der Matrix
  Das System muß überwachen, daß ein Benutzer nur mit für ihn zugelassenen Transformations-Prozeduren auf solche Kontrollierten Daten zugreift, die in der Matrix an dieser Stelle eingetragen sind.

- Benutzerauthentisierung:

  E_3: Authentisierung der Benutzer
  Das System muß jeden Benutzer (anhand von vorher festgelegten Merkmalen) authentisieren, bevor er eine Transformations-Prozedur ausführt.

- Rekonstruierbarkeit der Transaktionen:

  C_4: Logbuch
  Alle Transformations-Prozeduren müssen dahingehend zertifiziert sein, daß sie einem bestimmten Kontrollierten Datum, bei dem Einträge nicht gelöscht werden können, die Informationen mitteilen, die zur Rekonstruktion der Aktivität notwendig sind.

- Einfügen von neuen Informationen:

  C_5: Umwandlung von Unkontrollierten Daten
  Jede Transformations-Prozedur, die ein Unkontrolliertes Datum als Eingabe akzeptiert, gibt entweder nach der Ausführung ein Kontrolliertes Datum aus oder gibt das Unkontrollierte Datum unverändert zurück.

- Festgelegte Kontrolle und Pflichtentrennung:

  E_4: Systemänderungen
  Nur diejenigen Benutzer, die Einheiten zertifizieren dürfen, dürfen die Beziehungen dieser Einheiten zu anderen Einheiten (zum Beispiel die Listen und die Matrix) ändern. Derjenige, der eine Einheit zertifiziert, darf keine Ausführungsrechte bezüglich dieser Einheit haben.

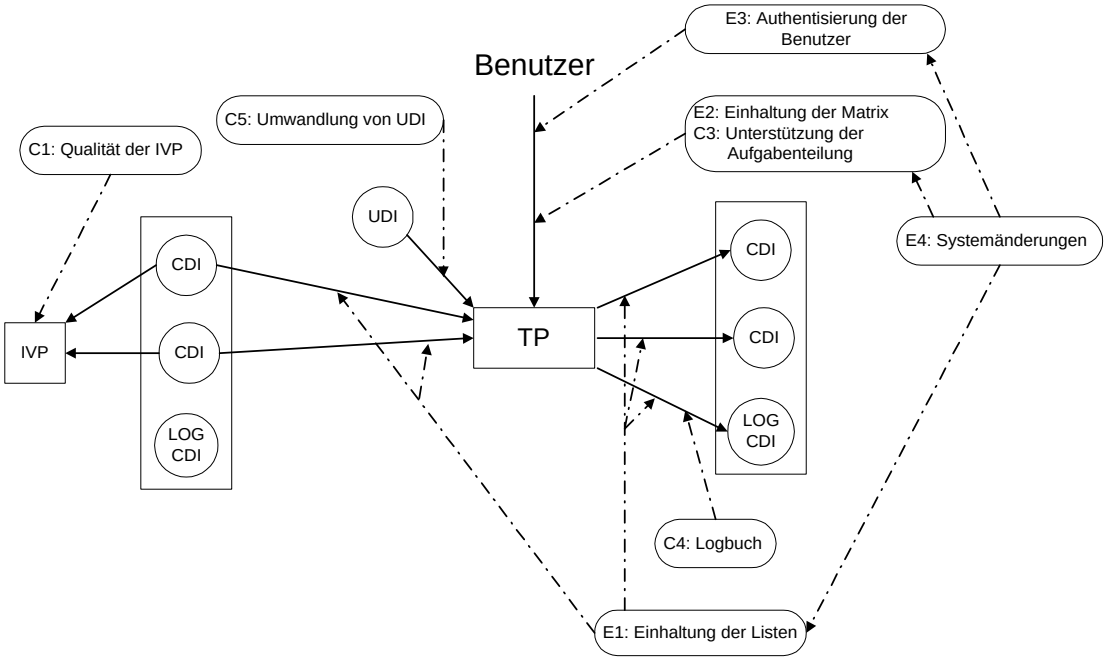


Abbildung 5-4: Grafische Darstellung des Clark-Wilson-Modells
Quelle: [Clark, Wilson 1987], S. 192

Abbildung 5-4 beschreibt, welche Regeln für die Transformations-Prozeduren gelten. Die durchgehenden Pfeile beschreiben den Zugriff des Benutzers auf Kontrollierte Daten (CDI) oder Unkontrollierte Daten (UDI) mit Hilfe von Transformations-

Prozeduren (TP). Die gestrichelten Pfeile beschreiben, bei welchem Zugriff welche Regel (C_i oder E_i) angewendet wird.

Durch das Prinzip der Pflichtentrennung kommt dem Auswerten des Logbuchs (Auditing) eine sicherheitskritische Funktion zu. Wenn ein Prozeß von zwei verschiedenen Personen angestoßen werden muß, braucht man verläßliche Informationen über vergangene Aktionen. Die zweite Person, die ihr Einverständnis für einen Prozeß gibt, muß feststellen können, ob die erste Person vorher ihr Einverständnis gegeben hat. Das Auditing ist also nicht nur zum nachträglichen Auffinden von Sicherheitsverstößen notwendig, sondern dient vor allem der Durchsetzung der Sicherheitsmaßnahmen.

5.5 Telekooperationsmodell

Der Zweck des 1994 vorgestellten Telekooperationsmodell [Grimm 1994] ist die Einbeziehung von Personen und unspezifizierbaren Komponenten in ein Sicherheitsmodell. Das Modell hat als wesentliche Bestandteile Personen, Rollen und Akteure. Es zerlegt dabei den temporären Akteur in zwei verschiedene permanente Beschreibungsbestandteile Person und Rolle. Person und Rolle verschmelzen für den Zeitraum einer Aktivität zu einem Akteur. Rollen beschreiben spezifische Handlungsmuster und Personen werden durch die persönlichen Attribute beschrieben, die für ein Telekooperationssystem relevant sind.

Telekooperation wird als eine besondere Form der Kooperation aufgefaßt, bei der die Kooperationspartner zeitlich oder räumlich getrennt sein können. Kooperation allgemein ist ein auf das Erreichen eines gemeinsamen Zieles hin gerichtetes Zusammenwirken von Personen ([Grimm 1994], S. 76). Der Unterschied zwischen Kooperation und Telekooperation ist nun der Einsatz von Telekommunikationstechnik. In der Telekooperation werden im Gegensatz zur Telekommunikation die handelnden Personen, ihre Ziele und Zwecke ihrer Handlungen mit einbezogen. Durch die Einbeziehung von Personen ins Telekooperationsmodell können spezielle Attribute wie persönliche Verantwortung der Akteure für ihre Handlungen mit aufgenommen werden. Durch Integration von Beweisverfahren kann verläßliche Kommunikationstechnik ein weiterer Baustein der Sicherheitsmaßnahmen darstellen ([Grimm 1994], S. 72).

5.5.1 Begriffsdefinitionen

Das Telekooperationsmodell kennt folgende Begriffe ([Grimm 1994], S. 77ff):

Person, Rolle, Akteur, Aktivität, Ziel, Zweck, Lebenszeit, Kooperationsprinzip, Kooperationsziel, Verträglichkeitsproblem, Koordinationsaufgabe, Konsistenzproblem, Kompetenz, Akteurskompetenz, Autorisierungsproblem, Realisierungsproblem der Akteurskompetenz, Übertragungsproblem von Personenkompetenzen, Nachricht, Handlungsalternative, Nachrichtenaustausch, Aktion, Handlungsschritt, Akteurszustand, Kooperationszustand, Ereignis, Zeit, Interpretationswandel, Kausalität, Basisrolle, Grundoperationen und Standardkooperationen.

Es wird hier auf eine ausführliche Wiedergabe der Begriffsdefinitionen des Telekooperationsmodells an dieser Stelle verzichtet und auf die Originalliteratur verwiesen [Grimm 1994]. Zum Verständnis des Telekooperationsmodells werden einzelne Begriffe erläutert und ihre Bedeutung im Modell dargestellt.

Eine **Person** ist eine natürliche oder juristische Person und existiert „permanent“ und unabhängig von spezifizierten Rollen. Sie verfügt über Kompetenz und handelt zweckorientiert ([Grimm 1994], S. 77).

Eine **Rolle** ist ein Handlungsmuster und existiert permanent und unabhängig davon, ob jemand in ihr aktiv ist. Das Handlungsmuster ist durch die Spezifikation des zielorientierten Handlungsablaufs und der akteursspezifischen Kompetenz definiert, die eine Person braucht, um in der Rolle aktiv sein zu können.

> *„Die spezifizierte Sequenz von Handlungsschritten umfaßt genau einen wohl definierten Eintrittspunkt, eine Folge von Handlungen, die möglicherweise bedingungsgesteuerte, nicht-deterministische Handlungsalternativen enthält, genau zwei Mengen von wohl definierten Austrittspunkten, die eindeutig mit dem Erreichen bzw. Nicht-Erreichen eines Kooperationsziels assoziiert sind.“ ([Grimm 1994], S. 77)*

Ein **Akteur** ist eine Person in einer Rolle. Eine **Aktivität** ist der Ablauf eines Handlungsmusters.

> *„Der Akteur repräsentiert die in der Rolle handelnde Person, die Aktivität bezeichnet den Ablauf der vom Akteur ausgeführten Aktionen. Als Akteur handelt die Person nach den Regeln der Rolle. Ein Akteur repräsentiert genau eine Person. Hingegen kann eine Person gleichzeitig in mehreren Rollen aktiv sein, wodurch ein Verträglichkeits- und Koordinationsproblem auftritt.“ ([Grimm 1994], S. 78)*

> *„Eine Aktion ist ein Handlungsschritt eines Akteurs innerhalb einer Rolle. Sie kostet Zeit und Ressourcen, über die der Akteur verfügt. Innerhalb der Rolle ist eine Aktion eine atomistische Basiseinheit.“ ([Grimm 1994], S. 84)*

Eine Rolle spezifiziert die Menge aller **Handlungsalternativen**, läßt jedoch offen, welche Alternativen von einem Akteur gewählt werden. Der Akteur fällt seine Entscheidungen aufgrund seiner Kompetenz und von externen Einflüssen, die Nachrichten aus der Außenwelt darstellen ([Grimm 1994], S. 84).

Das **Ziel** ist das Erreichen eines Kooperationszieles. Das Erreichen ist eine der beiden Mengen von wohl definierten Austrittspunkten, die andere Menge der Austrittspunkte ist das Nicht-Erreichen eines Kooperationszieles. Im einfachsten Fall hat eine Rolle ein Ziel und einen weiteren Austrittspunkt für das Nicht-Erreichen dieses Ziels ([Grimm 1994], S. 78).

Der **Zweck** einer Handlung ist der Sinn, den eine Person mit ihr und ihrem Ergebnis verbindet. Der Zweck liegt außerhalb der Rollenspezifikation. Es ist Aufgabe der handelnden Person, empfangene Nachrichten zweckorientiert auszuwerten und mit dem Ergebnis ihre Kompetenz zu erweitern ([Grimm 1994], S. 78).

Ein **Verträglichkeitsproblem** besteht, wenn eine Person gleichzeitig mehrere Rollen ausüben möchte, die sich gegenseitig ausschließen. Abhängig von der Anwendungssituation muß geklärt werden, welche Rollen sich gegenseitig ausschließen, d.h. welche Rollenkombinationen können gleichzeitig von einer Person wahrgenommen werden. In diesem Fall besteht die **Koordinationsaufgabe** darin, die Aktivitäten so zu steuern, daß möglichst viele Aktivitäten zum Erfolg führen ([Grimm 1994], S. 79).

Ein **Akteurszustand** ist die Ausprägung einer spezifizierten Eigenschaft zu einem bestimmten Zeitpunkt. Die Korrelation der Akteurszustände ist der **Kooperationszustand**, der den Akteuren zur Steuerung ihrer Aktivitäten dient. Im allgemeinen kennt ein Akteur nicht die tatsächlichen Zustände seiner Kooperationspartner, jedoch die möglichen Zustände ([Grimm 1994], S. 84).

Ein **Ereignis** (Transition) ist der Zustandswechsel eines Akteurs. Es stellt den vollendeten Übergang eines Zustandes in den nächsten dar. Idealerweise wird angenommen, daß ein Ereignis keine Zeit kostet. Der Ressourcenverbrauch eines Ereignisses ist die Ressourcendifferenz der beiden benachbarten Zustände eines Ereignisses ([Grimm 1994], S. 85).

5.5.2 Darstellung des Telekooperationsmodells

Das Telekooperationsmodell besteht aus einem informellen und einem formalen Teil. Der informelle Teil beschreibt Eigenschaften von Personen, die einen gewissen Sinn und Zweck mit bestimmten Handlungszielen verbinden, und repräsentiert damit die reale Welt mit ihren Menschen und Organisationen. Der formale Teil des Telekooperationsmodells beschreibt durch vorgegebene Handlungsmuster, wie Akteure formal definierten Handlungszielen zustreben.

Im täglichen Leben streben Menschen Ziele an, mit denen sie einen bestimmten Sinn und Zweck verbinden. Als Beispiel wird der Abschluß eines Kaufvertrages ([Grimm 1994], S. 87) über ein Auto (Ziel) mit dem Wunsch für den Kunden, einen möglichst günstigen Kaufpreis zu erzielen (Zweck) und dem Wunsch für den Händler, möglichst viel Geld zu verdienen (Zweck).

Wie man in Abbildung 5-5 sieht, ist das Ziel der Kooperationspartner ein gemeinsames mit jedoch unterschiedlichen und sich sogar widersprechenden Zwecken. Um ihre Ziele zu erreichen, folgen Menschen Handlungsmustern, die sie zu ihren Zielen bringen. Das Ziel ist also das Ende eines Handlungsmusters und der Zweck bezieht sich auf den Sinn, den ein Mensch mit dem Erreichen eines Ziels verbindet.

Handlungsmuster sind als Rollen modelliert. Die Regeln einer Rolle sind formal festgelegt, sie umfassen ein Handlungsmuster mit möglichen Handlungsalternativen und eine akteursspezifische Kompetenz, die eine Person besitzen muß, um in dieser Rolle

aktiv werden zu können. In dieser Kompetenz ist verankert, wie sich eine Person in ihrer Rolle zwischen den möglichen Handlungsalternativen entscheiden wird.

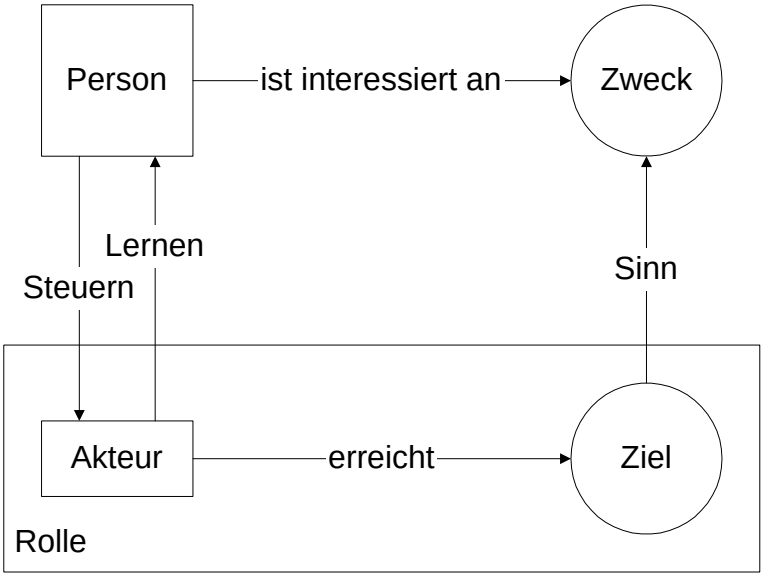


Abbildung 5-5: Beziehung zwischen Rollenziel und Handlungszweck
Quelle: [Grimm 1994], S. 88

Diese Entscheidungen sind nicht formal spezifiziert, sondern abhängig von der jeweiligen akteursspezifischen Kompetenz, die neben einem Rollengedächtnis auch spezifiziertes Rollenwissen umfaßt, sowie die Fähigkeit, vorstrukturierte Nachrichten zu unterscheiden. Eine Person bringt ihre Kompetenz in die Aktivität mit ein und geht mit einer veränderten Kompetenzbasis dieser Aktivität wieder heraus.

Das Telekooperationsmodell ist folglich ein Modell für ziel- und zweckorientiertes Handeln von Personen. Das Prinzip des Modells liegt darin, eine Aktivität in ihren spezifizierten und in ihren nicht-spezifizierten Anteil zu zerlegen, wobei der spezifizierte Anteil in der Rolle und der nicht-spezifierte Anteil in der Person untergebracht ist. Rollen und Personen sind verschiedene Beschreibungsbestandteile eines Akteurs.

Für einen bestimmten Zeitraum verschmelzen eine permanente Person und eine permanente Rolle zu einem temporären Akteur (Abbildung 5-6). Als Akteur handelt die Person nach den Regeln der Rolle. Entscheidungen zwischen eventuellen Handlungsalternativen trifft die Person aufgrund ihrer Kompetenz und äußerer Einflüsse.

Das Telekooperationsmodell sieht eine Hierarchie von Akteuren vor, die eine Delegation von Aufgaben und die Erweiterung von Spezifikationen erlaubt. Zwei Akteure stehen in einem hierarchischen Verhältnis, wenn aus Sicht des untergeordneten Akteurs der übergeordnete Akteur die verantwortliche Person darstellt, die zwischen Handlungsalternativen entscheiden kann.

Diese Hierarchie beschreibt die Erweiterung von Spezifikationen. In der anderen Richtung (also von oben nach unten) beschreibt das Ziel der untergeordneten Aktivität ein Teilziel der übergeordneten Aktivität. Auf diese Weise können Aufgaben „nach unten“ delegiert werden. Ein übergeordneter Akteur repräsentiert gegenüber allen ihm

untergeordneten Akteuren die verantwortliche Person. Hinter dem obersten Akteur steht immer eine Person, die für all ihre Akteure und Unterakteure verantwortlich ist ([Grimm 1994], S. 95).

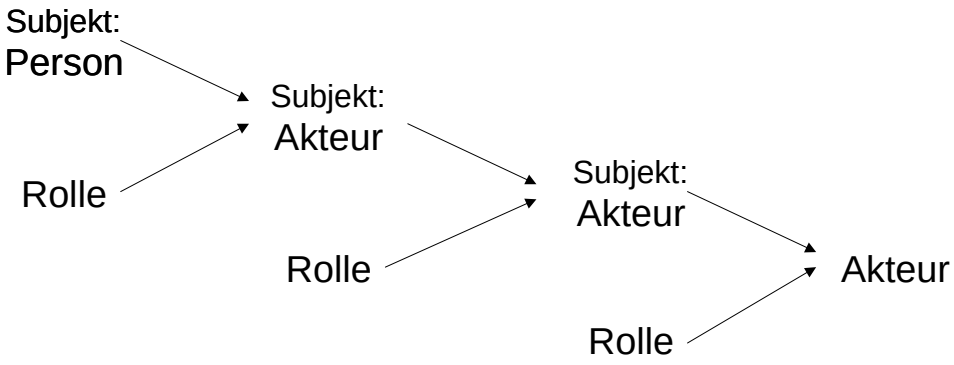


Abbildung 5-6: Hierarchie von Akteuren
Quelle: [Grimm 1994], S. 95

Die Wechselbeziehung zwischen einer Person und ihrer Außenwelt erfolgt über Akteure. Der Akteur ist die sogenannte Außenansicht einer handelnden Person. Die Schnittstelle zur Außenwelt wird über Nachrichten modelliert. Die Wechselwirkung erfolgt über Einflüsse von außen auf den Akteur (Nachrichten), und über Einwirkungen des Akteurs auf die Außenwelt (Nachrichten). Die Kooperation zweier Personen miteinander erfolgt ausschließlich über ihre Akteure.

Die Kooperation zwischen Akteuren erfolgt durch Nachrichtenaustausch über ihre Schnittstellen zur Außenwelt und folgt dem Kooperationsprinzip. Dieses besagt, daß eine Kooperationsrolle so spezifiziert sein muß, daß ein Akteur genau dann sein Kooperationsziel erreicht, wenn alle anderen Kooperationspartner auch ihre Ziele erreichen (Erfolgskopplung). Alle kooperierenden Partner haben also ein gemeinsames Kooperationsziel. Der Kooperationszweck kann verschieden, ja sogar widersprüchlich sein.

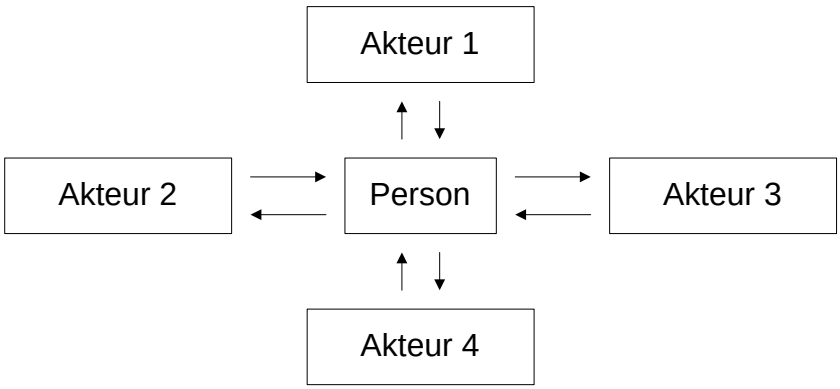


Abbildung 5-7: Koordination von Akteuren
Quelle: [Grimm 1994], S. 100

Da eine Person gleichzeitig in mehreren Rollen aktiv sein kann, muß sie diese gleichzeitig existierenden Akteure koordinieren (Abbildung 5-7). Dabei treten zwei Probleme auf, zum einen das Verträglichkeits- und zum anderen das Koordinationsproblem. Das Verträglichkeitsproblem behandelt die Frage, welche Rollen sich gegenseitig bei einer gleichzeitigen Benutzung ausschließen, also welche Rollenkombination von einer Person zu einem Zeitpunkt gewählt werden kann. Das Koordinationsproblem regelt die Frage, wie die einzelnen Aktivitäten so gesteuert werden, daß möglichst viele Aktivitäten zum Erfolg führen. Dabei können Prioritäten vergeben werden.

Als Weiterentwicklung des Telekooperationsmodells wird das Gleichgewichtsmodell ([Grimm 1994], S. 133ff) an dieser Stelle kurz vorgestellt. Die Problematik der Rechtssicherheit in offenen Systemen ist bekannt, insbesondere die grundsätzlich verschiedenen Anforderungen von Beweisbarkeit von Handlungen und Durchsetzbarkeit von Sicherheitsanforderungen. In der Literatur sind jedoch genügend Diskussionen zu finden, die zeigen, daß sich Rechtssicherheit und Anonymität nicht gezwungenermaßen ausschließen müssen ([Bürk, Pfitzmann 1990] und [Pfitzmann, Waidner, Pfitzmann 1990]).

Das Gleichgewichtsmodell beschreibt eine Einbettung von Verpflichtungen und ihren Zuständen in das allgemeine Telekooperationsmodell. Es stellt ein Verfahren, zur Konstruktion und Verifikation von allgemeinen kooperationstechnischen Anwendungen zur Verfügung, mit dem die Einhaltung von verbindlich eingegangenen Verpflichtungen überprüft werden kann ([Grimm 1994], S. 140).

5.6 Formales Datenschutzmodell

Zentral im formalen Datenschutzmodell ist die Tatsache, daß der Zugriff auf Objekte durch die momentanen Aufgaben eines Benutzers beschränkt wird [Fischer-Hübner 1994a/1994b/1995/1997].

Die zugrundeliegende Datenschutzpolitik kann folgendermaßen beschrieben werden:

> Der Zugriff auf personenbezogene Daten durch einen Benutzer soll nur dann erfolgen, wenn der Zugriff zur Erledigung der aktuellen Aufgabe erforderlich ist und der Benutzer zur Ausführung der Aufgabe autorisiert ist. Der Benutzer darf auf die Daten nur in kontrollierter Weise mit Hilfe von zertifizierten Transaktionsprozeduren zugreifen, für die die aktuelle Aufgabe des Benutzers autorisiert sind. Weiterhin muß der Zweck der aktuellen Aufgabe des Benutzers zugleich auch Erhebungszweck der personenbezogenen Daten sein oder der Betroffene, auf den sich die Daten beziehen, muß einer Verarbeitung zu diesem (Verarbeitungs-) Zweck schriftlich eingewilligt haben.

Das formale Datenschutzmodell behandelt Zustandsvariablen, Sicherheitsinvarianten und Zustandsüberführungsfunktionen, die im folgenden erläutert werden.

5.6.1 Zustandsvariablen

Unter Zustandsvariablen versteht man die Variablen, mit denen sich eine Datenschutzpolitik formal definieren läßt. Im folgenden werden die Zustandsvariablen Subjekte, Objekte, Aufgaben, Zwecke, Rechte, Zugriffe und Einwilligungen im einzelnen beschrieben:

Subjekte (S): Subjekte sind die aktiven Einheiten eines Systems, zum Beispiel Benutzer oder Prozesse.

$S = \{s_1, s_2, ...\}$ = Menge aktueller Subjekte

Objekte (O): Objekte sind passive Einheiten, die personenbezogene Daten enthalten. Personenbezogene Daten sind die Daten, mit denen man eine Person identifizieren kann. Die Frage, ob eine Person identifizierbar ist, hängt jedoch noch von zusätzlichem Wissen ab, das ein möglicher Angreifer hat. Da man über dieses zusätzliche Wissen jedoch nichts weiß, geht man davon aus, daß alle Daten personenbezogene Daten sind, da die Möglichkeit einer Reidentifizierung nicht ausgeschlossen werden kann.

$O = \{o_1, o_2, ...\}$ = Menge aktueller Objekte

Objektklassen (O_Klasse): Ein Objekt, das personenbezogene Daten enthält, kann normalerweise in bestimmte Klassen eingeteilt werden, wie zum Beispiel Patientendaten im Krankenhaus oder Bankkontodaten. Eine größere Praktikabilität kann dadurch erreicht werden, daß die Datenschutzgrundsätze wie Zweckbindung und Erforderlichkeit auf Objektklassen geprüft werden und nicht auf einzelne Objekte.

$O\_Klasse = \{o\_klasse_1, o\_klasse_2, ...\}$ = Menge der unterschiedlichen Objektklassen

$Klasse : O \rightarrow O\_Klasse$ wobei $Klasse(o_i)$ die Objektklasse des Objekt o_i ist.

Aufgaben (A): Aktionen von Subjekten auf Objekte können ausschließlich durch die Wahrnehmung bestimmter Aufgaben erfolgen, die in Abhängigkeit von der jeweiligen Anwendung zu definieren sind.

$A = \{a_1, a_2, ...\}$ = Menge von Aufgaben

Man unterscheidet Aktuelle und Autorisierte Aufgaben.

Aktuelle Aufgaben (Akt_A): Die Aktuelle Aufgabe eines Subjektes ist die Aufgabe, die ein Subjekt gerade ausführt. Jedem Subjekt wird durch folgende Funktion genau eine aktuelle Aufgabe zugeordnet: Übt ein Subjekt zur Zeit keine aktuelle Aufgabe aus, so ist der Wert der aktuellen Aufgabe Nil.

$Akt\_A : S \rightarrow A \cup Nil$ wobei $Akt\_A(s_i)$ die aktuelle Aufgabe des Subjekts s_i ist.

Autorisierte Aufgaben (Aut_A): Autorisierte Aufgaben eines Subjektes sind die Aufgaben, die ein Subjekt ausführen darf, wobei ein Subjekt mindestens eine autorisierte Aufgabe ausführen darf:

> $Aut\_A : S \rightarrow 2^A \setminus \varnothing$ wobei $Aut\_A(s_i)$ die Menge der Aufgaben ist, für die das Subjekt s_i autorisiert ist.

Benutzer, die verantwortlich für eine Aufgabe sind, dürfen diese an andere delegieren. Aus diesem Grund wird eine Funktion Verant eingeführt:

Verantwortliche Benutzer (Verant): Für jede Aufgabe kann es verantwortliche Benutzer geben, die beim Sicherheitsadministrator beantragen können, andere Personen für diese Aufgabe zu autorisieren.

> $Verant : A \rightarrow 2^S$ wobei $Verant(a_i)$ die Menge von Benutzern ist, die die Aufgabe a_i delegieren dürfen.

Benutzerrollen (Rolle): Subjekte können verschiedenen sicherheitsrelevanten Rollen zugeordnet werden, in denen sie aktiv sind.

> $Rolle : S \rightarrow ROL$ wobei $Rolle(s_i)$ die Rolle ist, in der das Subjekt s_i gerade aktiv ist.
>
> ROL besteht aus den Elementen: Benutzer, Datenschutzbeauftragter, Sicherheits-administrator, TP-Manager, ...

Zwecke (Z): Personenbezogene Daten müssen einen bestimmten Erhebungszweck haben. Außerdem müssen Aufgaben einem definierten Verwendungszweck dienen. Erhebungs- und Verwendungszweck sind in Abhängigkeit der jeweiligen Anwendung zu definieren.

> $Z = \{z_1, z_2, ...\}$ = Menge von Zwecken
>
> Zwecke und Aufgaben können hierarchisch strukturiert werden ([Bräutigam, Höller, Scholz 1990], S. 47). Zwecke können beispielsweise in unterschiedliche Teilzwecke aufgespalten werden oder zu höheren Zwecken zusammengefaßt werden. Das gleiche gilt für Aufgaben. Datenschutzaspekte und praktische Gründe müssen berücksichtigt werden, um ein angemessenes Level in dieser Hierarchie zu wählen. Mit den Zwecken dieses Levels werden die Elemente von Z definiert. Folglich dürfen nur Zwecke dieses Levels oder höherer Level mit diesen Elementen und nichtleeren Teilmengen von Z modelliert werden.
>
> Die Elemente der Menge A müssen entsprechend definiert werden. Jede Aufgabe von A muß genau einem Zweck dienen, wobei jeder Zweck durch mehrere Aufgaben erledigt werden kann und verschiedene Zwecke werden durch jeweils disjunkte Mengen von Aufgaben erledigt.

Zweckfunktion für Aufgaben (A_Zweck): Jede Aufgabe aus der Menge A muß genau einem Zweck dienen. Mit folgender Funktion wird jede Aufgabe einem Zweck zugeordnet:

$A\_Zweck : A \rightarrow Z$ wobei $A\_Zweck\ (a_i)$ den Zweck der Aufgabe a_i beschreibt.

Zweckfunktion für Objektklassen (O_Zweck): Jede Objektklasse muß für einen oder mehrere Zwecke spezifiziert sein, für den die Daten dieser Objektklasse erhoben wurden. Da eine Objektklasse aus Teilobjektklassen bestehen kann, die jeweils aus verschiedenen Zwecken erhoben wurden, wird jeder Objektklasse eine nichtleere Teilmenge von Z zugeordnet:

$O\_Zweck : O\_Klasse \rightarrow 2^Z \setminus \varnothing$ wobei $O\_Zweck\ (o\_klasse_i)$ die Menge von Zwecken ist, für die das Objekt der Objektklasse o_klasse_i erhoben wurde.

Transformations-Prozeduren (Trans): Ein Subjekt darf nicht willkürlich auf ein Objekt zugreifen. Wenn es eine Aufgabe ausübt, sollen spezielle Transformations-Prozeduren (TP) ausgeführt werden, die den Zugriff auf Objekte in einer kontrollierten Art und Weise durchführen.

$Trans = \{trans_1, trans_2, ...\}$ = Menge aller Transformations-Prozeduren

Autorisierte Transformations-Prozeduren (Aut_Trans): Während der Ausübung einer Aufgabe ist ein Subjekt autorisiert, bestimmte Transformations-Prozeduren auszuführen.

$Aut\_Trans : A \rightarrow 2^{Trans}$ wobei $Aut\_Trans\ (a_i)$ die Transformations-Prozeduren ausgibt, die für der Aufgabe a_i autorisiert sind.

Aktuelle Transformations-Prozeduren (Akt_Trans): Die Transformations-Prozedur, die aktuell ausgeführt wird, wird aktuelle Transformations-Prozedur genannt. Wenn ein Subjekt zur Zeit keine Transformations-Prozedur ausführt, wird der Wert der aktuellen Transformations-Prozedur auf Nil gesetzt.

$Akt\_Trans : S \rightarrow Trans \cup Nil$ wobei $Akt\_Trans\ (s_i)$ die aktuelle Transformations-Prozedur ist, die das Subjekt s_i gerade ausführt.

Rechte (R): Die Zugriffsrechte, die ein Subjekt auf ein Objekt ausüben darf, werden durch eine Menge von Zugriffsattributen beschrieben.

$Rechte = \{r_1, r_2, r_3, ...\}$ = Menge aller Rechte, wobei r_1 bis r_n für die verschiedenen Zugriffsmodi stehen (zum Beispiel Lesen, Schreiben, Ausführen oder Erzeugen, Verändern, Löschen, ...).

Erforderliche Zugriffe (E_Zugriff): Für jede Aufgabe muß vorab definiert werden, welche Zugriffe mit Hilfe welcher Transformations-Prozeduren auf welche Objektklassen zu ihrer Erledigung erforderlich sind.

$E\_Zugriff = \{a_i, o\_klasse_j, trans_k, r_l\}$ = Tupel der erforderlichen Zugriffe, wobei zur Erledigung der Aufgabe a_i der r_l-te Zugriff auf Objekte der Objektklasse o_klasse_j mit Hilfe der Transformations-Prozedur $trans_k$ erforderlich ist.

Aktuelle Zugriffe (A_Zugriff): Der aktuelle Zugriff, den ein Subjekt auf ein bestimmtes Objekt haben soll, wird definiert durch die Menge.

> A_Zugriff = $\{s_i, o_j, trans_k, r_l\}$ = Tupel der aktuellen Zugriffe, wobei das Subjekt s_i auf das Objekt o_j den Zugriff r_l mit Hilfe der Transformations-Prozedur $trans_k$ haben soll.

Einwilligungen (E): Nach meisten nationalen Datenschutzgesetzen ist die Verarbeitung personenbezogener Daten auch ohne gesetzliche Grundlage für einen bestimmten Zweck zulässig, falls der Betroffene eingewilligt hat.

> E = $\{z_i, o_j\}$ = Menge der Einwilligungen, wobei das Datensubjekt, auf das sich die Daten des Objektes o_j beziehen, zu einer Verarbeitung zum Zweck z_i eingewilligt hat.

5.6.2 Sicherheitsinvarianten

Um einen datenschutzkonformen Systemzustand zu erreichen, müssen Datenschutzeigenschaften definiert werden, die in formaler Weise die Datenschutzpolitik des Systems beschreiben. Für jeden Zustand, der durch die oben definierten Zustandsvariablen beschrieben ist, müssen folgende Bedingungen gelten:

Autorisierungseigenschaft für Aufgaben

Ein Subjekt darf nur solche Aufgaben ausüben, für die es autorisiert ist.

$$\forall\ s_i{:}S :$$

$$Akt\_A\ (s_i) \in\ Aut\_A\ (s_i)$$

Autorisierungseigenschaft für Transformations-Prozeduren

Die aktuelle Transformations-Prozedur eines Subjekts muß für die aktuelle Aufgabe des Subjekts autorisiert sein.

$$\forall\ s_i{:}S :$$

$$Akt\_Trans\ (s_i) \in\ Aut\_Trans\ (Akt\_A\ (s_i))$$

Erforderlichkeitseigenschaft

Ein Subjekt darf nur Zugriff durch Ausführung einer Transformations-Prozedur auf ein Objekt haben, auf dessen Objektklasse der spezifizierte Zugriff zur Erledigung einer bestimmten Aufgabe erforderlich ist.

$$\forall\ s_i{:}S,\ o_j{:}O,\ trans_k{:}Trans :$$

$$(s_i, o_j, trans_k, r_l) \in\ A\_Zugriff \Rightarrow$$

$$(Akt\_A\ (s_i), Klasse\ (o_j), trans_k, r_l) \in\ E\ \_Zugriff$$

Zweckbindungseigenschaft

Ein Subjekt darf nur auf ein Objekt zugreifen, wenn der Zweck der aktuellen Aufgabe mit dem Zweck korrespondiert, zu dem das Objekt erhoben wurde, oder wenn der Betroffene, auf den sich die Daten des Objekts beziehen, eingewilligt hat.

$$\forall\ s_i{:}S,\ o_j{:}O,\ trans_k{:}Trans :$$

$$(s_i,\ o_j,\ trans_k,\ r_l) \in A\_Zugriffe \Rightarrow$$

$$(A\_Zweck\ (Akt\_A\ (s_i)) \in O\_Zweck\ (Klasse\ (o_j)) \vee$$

$$(A\_Zweck\ (Akt\_A\ (s_i)),\ o_j) \in E)$$

5.6.3 Zustandsüberführungsfunktionen

Um zulässige Änderungen der Zustandsvariablen zu beschreiben, müssen Zustandsüberführungsfunktionen definiert werden. Die Zustandsüberführungsfunktionen werden in ihrer Funktion und ihrer Struktur beschrieben. Das Zeichen ’ nach Zustandsvariablen beschreibt den Nachfolgezustand der Zustandsvariablen.

Beispielhaft werden die Zustandsüberführungsfunktionen change_current_task () get_access (), create_object (), delet_object (), und execute_transformation_procedure () beschrieben:

change_current_task (s_i, a_j)

Subjekt s_i möchte die aktuelle Aufgabe beenden und anschließend eine neue Aufgabe a_j ausführen. Die aktuelle Aufgabe des Subjekts s_i soll also in die aktuelle Aufgabe a_j gewechselt werden.

if $a_j \in Aut\_A\ (s_i)$ and $Akt\_Trans\ (s_i) = Nil$

then $Akt\_A'\ (s_i) = a_j$

get_access (s_i, o_j, r_k)

Subjekt s_i möchte Zugriff r_k auf das Objekt o_j bekommen.

if $(Akt\_A\ (s_i),\ Klasse\ (o_j),\ Akt\_Trans\ (s_i),\ r_k) \in E\_Zugriffe$

and

[$A\_Zweck\ (Akt\_A\ (s_i)) \in O\_Zweck\ (Klasse\ (o_j))$

or

$(A\_Zweck\ (Akt\_A\ (s_i),\ o_j) \in E$]

then $A\_Zugriffe' = A\_Zugriffe \cup \{s_i,\ o_j,\ Akt\_Trans\ (s_i),\ r_k\}$

create_object (s_i, o_j, o_klasse_k,)

Subjekt s_i möchte ein Objekt o_j der Objektklasse o_klasse_k erzeugen.

if (Akt_A (s_i), o_klasse_k, Akt_Trans (s_i), r_k) ∈ E_Zugriffe
and
[A_Zweck (Akt_A (s_i)) ∈ O_Zweck (o_klasse_k)
or
(A_Zweck (Akt_A (s_i)), o_j) ∈ E]

then O' = O ∪ {o_j}
Klasse (o_j) = o_klasse_k

delete_object (s_i, o_j)

Subjekt s_i möchte Objekt o_j löschen.

if (Akt_A (s_i), Klasse (o_j), Akt_Trans (s_i), r_k) ∈ E_Zugriffe
and
[A_Zweck (Akt_A (s_i)) ∈ O_Zweck (Klasse (o_j))
or
(A_Zweck (Akt_A (s_i)), o_j) ∈ E]

then O' = O \ {o_j}

execute_transformation_procedure (s_i, $trans_j$)

Subjekt s_i möchte die Transformations-Prozedur $trans_j$ ausführen.

if $trans_j$ ∈ Aut_Trans (Akt_A (s_i))

then Akt_Trans' (s_i) = $trans_j$

Weiterhin sind privilegierte Funktionen notwendig, um Aufgaben, Zwecke, autorisierte Aufgaben für Subjekte, autorisierte Transformations-Prozeduren für Aufgaben, Objektklassen, notwendige Zugriffe und Einwilligungen zu definieren und zu verändern. Gemäß dem Vier-Augen-Prinzip darf die Definition dieser Mengen und Funktionen nur in Kooperation mit einer anderen Person erfolgen, die sich um die datenschutzgemäßen Belange der Benutzer kümmert, beispielsweise der firmeninterne Datenschutzbeauftragte oder ein Gewerkschaftsmitglied. Um dieses zu realisieren, wird eine Systemvariable für ein „Einmal-Ticket" definiert.

Ein Ticket TKT_i (Funktionstyp, Parameter-Liste) wird von einem Subjekt s_j (zum Beispiel dem Datenschutzbeauftragten) ausgestellt oder unterzeichnet und zu einer

anderen Person s_i (in der Regel der Sicherheitsadministrator) geschickt. Dies bedeutet, daß der Datenschutzbeauftragte den Sicherheitsadministrator beauftragt, eine spezielle Funktion mit bestimmten Parametern auszuführen. Mit diesem entsprechenden Ticket darf der Sicherheitsadministrator privilegierte Funktionen ausführen.

Privilegierte Funktionen haben folgende Form:

erteile_erforderlichen-Zugriff (s_i, TKT_j (funktion, a_k, o_klasse_m, $trans_n$, r_o))

Das Subjekt s_i beantragt, daß das Tupel (a_k, o_klasse_m, $trans_n$, r_o) zu der Menge der erforderlichen Zugriffe E_Zugriffe hinzugefügt wird. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

if Rolle (s_i) = Sicherheitsadministrator

and

funktion = erteile_erforderlichen-Zugriff

and

Rolle (s_j) = Datenschutzbeauftragter

then E_Zugriffe' = E_Zugriffe $\cup$ {a_k, o_klasse_m, $trans_n$, r_o}

lösche_erforderlichen_Zugriff (s_i, TKT_j (funktion, a_k, o_klasse_m, $trans_n$, r_o))

Das Subjekt s_i beantragt, daß das Tupel (a_k, o_klasse_m, $trans_n$, r_o) aus der Menge der erforderlichen Zugriffe E_Zugriffe gelöscht wird. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_Aufgabe_hinzu (s_i, TKT_j (funktion, a_k, z_m))

Das Subjekt s_i beantragt, die Aufgabe $a_{k,}$, die dem Zweck z_m dient, zu definieren. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_Aufgabe (s_i, TKT_j (funktion, a_k))

Das Subjekt s_i beantragt, die Aufgabe a_k aus der Menge der Aufgaben A zu löschen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_Objektklasse_hinzu (s_i, TKT_j (funktion, o_klasse_k, z_m))

Das Subjekt s_i beantragt, die Objektklasse $o\_klasse_{k,}$, die dem Zweck z_m,dient, zu definieren. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_Objektklasse (s_i, TKT_j (funktion, o_klasse_k))

Das Subjekt s_i beantragt, die Objektklasse o_klasse_k aus der Menge der Objektklassen zu löschen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_autorisierte_Transformations-Prozedur_hinzu (s_i, TKT_j (funktion, a_k, $trans_m$))

Das Subjekt s_i beantragt, die Transformations-Prozedur $trans_m$ für die Aufgabe a_k zu autorisieren. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_autorisierte_Transformations-Prozedur (s_i, TKT_j, (funktion, a_k, $trans_m$))

Das Subjekt s_i beantragt, die Transformations-Prozedur $trans_m$ für die Aufgabe a_k zu löschen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_Einwilligung_hinzu (s_i, TKT_j (funktion, z_k, o_m))

Das Subjekt s_i beantragt, das Tupel ($z_{k,}$, o_m) der Menge der Einwilligungen hinzuzufügen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_Einwilligung (s_i, TKT_j (funktion, z_k, o_m))

Das Subjekt s_i beantragt, das Tupel ($z_{k,}$, o_m) aus der Menge der Einwilligungen zu löschen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_Zweck_hinzu (s_i, TKT_j (funktion, z_m))

Das Subjekt s_i beantragt, den Zweck z_m der Menge der Zwecke hinzuzufügen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_Zweck (s_i, TKT_j (funktion, z_m))

Das Subjekt s_i beantragt, den Zweck z_m aus der Menge der Zwecke zu löschen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

füge_verantwortliche_Benutzer_hinzu (s_i, TKT_j (funktion, s_k, a_m))

Das Subjekt s_i beantragt, dem Benutzer $s_{k,}$ die Verantwortung für die Aufgabe a_m zu übertragen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_verantwortliche_Benutzer (s_i, TKT_j (funktion, s_k, a_m))

Das Subjekt s_i beantragt, dem Benutzer $s_{k,}$ die Verantwortung für die Aufgabe a_m zu entziehen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

In einigen Anwendungsfällen mag es hilfreich oder vernünftig sein, Aufgaben an andere Personen zu übertragen. Benutzer dürfen jedoch nicht selbst in der Lage sein, diese Aufgaben zu delegieren, da diskrete Zugriffskonzepte, bei denen der Zugriff direkt weitergegeben werden kann, verletzlich gegenüber trojanischen Pferden sind. Aus diesem Grund ist es sicherer, wenn die Delegation von Aufgaben von einem Sicherheitsadministrator kontrolliert wird. Wenn ein Benutzer eine Aufgabe delegieren möchte, muß er für diese Aufgabe verantwortlich sein und ein Ticket bei dem Sicherheitsadministrator beantragen.

füge_autorisierte_Aufgabe_hinzu (s_i, TKT_j (funktion, s_k, a_m))

Das Subjekt s_i beantragt, den Benutzer $s_{k,}$ für die Aufgabe a_m zu autorisieren. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

lösche_autorisierte_Aufgabe (s_i, TKT_j (funktion, s_k, a_m))

Das Subjekt s_i beantragt, dem Benutzer $s_{k,}$ die Autorisierung für die Aufgabe a_m zu entziehen. Das Subjekt s_i hat ein Ticket von dem Subjekt s_j, das die Ausführung legitimiert.

Das Erzeugen und Löschen von Transformations-Prozeduren darf nur von Benutzern in der Rolle des TP-Managers erfolgen. Die Funktionen dazu sehen folgendermaßen aus:

erzeuge_Transformations-Prozedur (s_i, $trans_j$)

Das Subjekt s_i erzeugt eine neue Transformations-Prozedur $trans_j$ und nimmt sie in die Menge der Transformations-Prozeduren auf.

lösche_Transformations-Prozedur (s_i, $trans_j$)

Das Subjekt s_i löscht die Transformations-Prozedur $trans_j$ aus der Menge der Transformations-Prozeduren.

5.7 Rollenbasiertes Zugriffsmodell

Aus dem Jahre 1992 stammt ein Entwurf von David F. Ferraiolo und Richard Kuhn [Ferraiolo, Kuhn 1992], über ein rollenbasiertes Konzept zur Zugriffskontrolle (Role Based Access Control, RBAC), das 1995 weiterentwickelt wurde [Ferraiolo, Cugini, Kuhn 1995].

Weitere Varianten des rollenbasierten Zugriffsmodells wurden von Sandhu und anderen formuliert [Sandhu, Coyne, Feinstein, Youman 1994/1996]. Sie beschreiben zusätzlich die administrative Seite des rollenbasierten Zugriffsmodells.

Das zentrale Ziel in einem rollenbasierten System ist der Schutz der Integrität von Informationen. Die daher gestellte Grundfrage ist: Wer kann welche Transaktionen auf welchen Informationen ausführen?

Entscheidungen über Zugriffskontrolle werden häufig von der Rolle eines individuellen Benutzers bestimmt, die dieser in einer bestimmten Organisation ausübt. Wichtige Faktoren daher sind Verpflichtung, Verantwortung und Qualifikation eines Benutzers.

Das Verfahren eines rollenbasierten Zugriffsmodells basiert auf dem Prinzip, daß Entscheidungen bezüglich der Zugriffskontrolle über die Funktionen erfolgt, die ein Benutzer innerhalb einer Organisation ausführen darf. Ein Benutzer kann keine Zugriffsrechte auf andere Benutzer übertragen, außer er ist der Sicherheitsadministrator. Der Sicherheitsadministrator ist verantwortlich für die Sicherheitspolitik und repräsentiert somit die Organisation.

Rollenbasierte Zugriffsmodelle sind dazu geeignet, individuelle Sicherheitspolitiken umzusetzen. Dies kann durch eine zentrale Sicherheitsadministration erfolgen, da die Entscheidungen über den jeweiligen Zugriff zentral, zum Beispiel in einem Regelsatz, erfolgen. Sind einmal die Rollen mit ihren zugehörigen, das heißt erlaubten, Transaktionen in einem System definiert, besteht die Hauptaufgabe der Administration darin, neue Mitglieder in Rollen aufzunehmen oder ausgeschiedene Mitglieder zu löschen. Verschiedene Rollen können zusammengesetzt werden.

Ein weiteres grundlegendes Prinzip im rollenbasierten Zugriffsmodell ist die Anforderung, daß ein Benutzer nicht mehr Rechte haben darf, als seine Aufgabe es erfordert (minimales Rechteprinzip).

Im rollenbasierten Zugriffsmodell werden zwei Entwicklungsstränge verfolgt:

- Verteilte Systeme beziehungsweise Betriebssysteme
- Datenbanksysteme

In diesem Abschnitt wird der Schwerpunkt der Betrachtung auf den Bereich der verteilten Systeme und Betriebssysteme gelegt.

5.7.1 Begriffsdefinitionen

Bevor die grundlegenden Regeln des rollenbasierten Zugriffsmodell erläutert werden, erfolgt eine Erklärung der verwendeten Begriffe.

Benutzer (b): Ein Benutzer ist eine Person.

Subjekt (s): Ein Subjekt repräsentiert den Prozeß eines Benutzers, der in einem gegebenen Kontext agieren darf.

Transaktion (t): Eine Transaktion ist eine Handlung/Prozedur in einer bestimmten Zugriffsart, die auf Objekte angewendet wird.

Rolle (r): Eine Rolle ist eine Menge von Transaktionen, die ein Benutzer oder eine Menge von Benutzern innerhalb eines Kontextes ausführen darf. Mitglieder einer Rolle sind Benutzer.

Objekt (o): Auf ein Objekt dürfen autorisierte Transaktionen ausgeführt werden.

Funktion (f): Die Definition einer (Geschäfts-) Funktion ist notwendig, um die operative Trennung von Pflichten zu realisieren.

5.7.2 Grundlegende Vereinbarungen

5.7.2.1 Benutzer, Rollen und Transaktionen

Die Beziehung zwischen Benutzern, Rollen und Transaktionen wird von den Autoren des rollenbasierten Zugriffsmodells folgendermaßen dargestellt (Entity-Relationship-Modell, Abbildung 5-8):

Transaktionen repräsentieren Handlungen auf ein oder mehrere Objekte. Sie werden Rollen zugeordnet, deren Mitglieder Benutzer sind. Der Doppelpfeil beschreibt die N-zu-M-Beziehung. Zum Beispiel kann ein einzelner Benutzer Mitglied einer oder mehrerer Rollen sein. Ebenso kann eine Rolle ein oder mehrere Mitglieder (Benutzer) haben. Äquivalent dazu kann eine Rolle eine oder mehrere Transaktionen enthalten, sowie eine Transaktion einer oder mehreren Rollen zugeordnet sein.

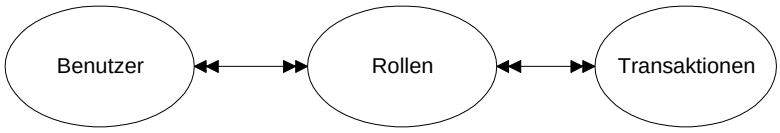


Abbildung 5-8: Beziehung zwischen Benutzern, Rollen und Transaktionen
Quelle: [Ferraiolo, Cugini, Kuhn 1995] S. 242

Transaktionen sind nicht nur Rollen zugeordnet, sondern auch Objekten, auf denen die Transaktionen ausgeführt werden dürfen (Abbildung 5-10).

Ein Subjekt repräsentiert einen aktiven Benutzerprozeß. Ein Benutzer kann ein oder mehreren Subjekten zugeordnet sein. Abbildung 5-9 beschreibt die Eins-zu-N-Beziehung zwischen Benutzer und Subjekten.

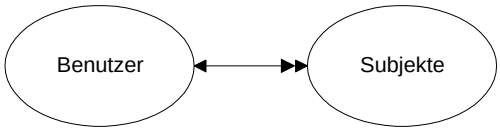


Abbildung 5-9: Beziehung zwischen Benutzern und Subjekten
Quelle: [Ferraiolo, Cugini, Kuhn 1995] S. 243

Nachfolgend werden einige Eigenschaften der Beziehungen zwischen Benutzern, Subjekten und Rollen beschrieben:

Subjekt_Benutzer (s:Subjekt) = {Der Benutzer b, der mit Subjekt s verbunden ist}

Autorisierte_Rollen (s:Subjekt) = {Autorisierte Rollen für Subjekt s}

Rollen_Mitglied (r:Rolle) = {Die Benutzer, die für die Rolle r autorisiert sind}

Benutzer_Autorisierte_Rollen (b:Benutzer) =
{Die Rollen, für die Benutzer b autorisiert ist}

Konsistentes Subjekt:

Wenn R die Menge der Rollen ist, die für Subjekt s autorisiert sind, und b der Benutzer ist, der dem Subjekt s zugeordnet ist, dann muß b den Rollen zugeordnet sein, die zur Menge R gehören.

$$\forall\ s_i\text{:Subjekt}, b_j\text{:Benutzer}, r_k\text{:Rolle}, R :$$

$$\text{Subjekt\_Benutzer}\ (s_i) = b_j \wedge \text{Autorisierte\_Rolle}\ (s_i) = R$$

$$\wedge\ b_j \in \text{Rollen\_Mitglied}\ (r_k) \Rightarrow r_k \in R$$

Rollen_Transaktionen (r: Rolle) = {Autorisierte Transaktionen,
die mit der Rolle r verbunden sind}

Transaktion_Objekte (t:Transaktion) = {Objekte, auf die die Transaktion t
angewendet werden darf}

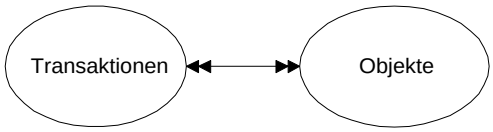


Abbildung 5-10: Beziehung zwischen Transaktionen und Objekten
Quelle: [Ferraiolo, Cugini, Kuhn 1995] S. 244

5.7.2.2 Rollen und Rollenhierarchien

(1. Regel) Rollenhierarchie:

Wenn ein Subjekt die Transaktionen einer Rolle ausführen darf und diese Rolle enthält eine weitere Rolle, dann darf das Subjekt auch die Transaktionen dieser weiteren Rolle ausführen.

$$\forall\ s_i\text{:Subjekt}, r_{j,k}\text{:Rolle}, j \neq k :$$

$$r_k \in \text{Autorisierte\_Rollen}\ (s_i) \wedge r_k > r_j \Rightarrow r_j \in \text{Autorisierte\_Rollen}\ (s_i)$$

5.7.2.3 Rollenautorisierung

Exklusive_Rollen (r:Rolle) = {Liste von Rollen , die sich mit der Rolle r
gegenseitig ausschließen}

Das rollenbasierte Zugriffsmodell unterstützt die Trennung von Aufgabenbereichen und Pflichten (separation of duty). Dies erfordert die Einhaltung des Grundsatzes, daß in einer Menge von Transaktionen kein Benutzer alle diese Transaktionen ausführen darf, sondern immer nur eine Teilmenge davon. Diese Trennung kann sowohl statisch als auch dynamisch oder operationell erfolgen. Statische Trennung von Verantwortung kann relativ einfach über die Zuordnung von Benutzern zu Rollen und Transaktionen zu Rollen geleistet werden. Dynamische Trennung von Pflichten wird mit Hilfe der 6. Regel, operationelle Trennung von Pflichten mit Hilfe der 8. Regel erreicht.

(2. Regel) Statische Trennung von Pflichten:

Wenn ein Benutzer b Mitglied einer Rolle r_i ist und Mitglied einer Rolle r_j, dann schließen sich diese beiden Rollen (r_i und r_j) nicht gegenseitig aus.

$$\forall\ b_i\text{:Benutzer}, r_{j,k}\text{:Rolle}, j \neq k :$$

$$b_i \in \text{Rollen\_Mitglied}\ (r_j) \wedge b_i \in \text{Rollen\_Mitglied}\ (r_k) \Rightarrow r_j \notin \text{Exklusive\_Rollen}\ (r_k)$$

(3. Regel) Rollenkapazität:

Die Anzahl der Mitglieder einer Rolle r darf die zulässige Mitgliederzahl der Rollen (Mitgliederbegrenzung) nicht überschreiten.

Mitglieder_Limit (r:Rolle) = {Mitgliedsbegrenzung (≥ 0) für Rolle r}

Anzahl_Mitglieder (r:Rolle) = {Anzahl der Mitglieder von Rolle r}

$$\forall\ r_i\text{:Rolle} :$$

$$\text{Mitglieder\_Limit}\ (r_i) \geq \text{Anzahl\_Mitglieder}\ (r_i)$$

5.7.2.4 Rollenaktivierung

Das Verfahren der rollenbasierten Zugriffskontrolle erfordert es, daß ein Benutzer zuerst für eine aktive Rolle autorisiert sein muß, bevor er Transaktionen ausführen darf.

Abhängig von der jeweiligen Politik des Systems oder der Organisation, in der das rollenbasierte Zugriffsmodell eingesetzt wird, werden nacheinander verschiedene Bedingungen getestet. Zuerst wird die Rolle geprüft, die zur Aktivierung vorgeschlagen ist, dann die Transaktion, die ausgeführt werden soll, und dann das Objekt, auf das zugegriffen werden soll.

Eine Rolle kann also nur dann aktiviert werden, wenn:

- der Benutzer für die vorgeschlagene Rolle autorisiert ist
- die vorgeschlagene Rolle sich nicht mit anderen aktiven Rollen des Benutzers gegenseitig ausschließt
- die vorgeschlagene Transaktion für die genannte Rolle autorisiert ist
- die Transaktion konsistent innerhalb einer vorgeschriebenen Reihenfolge von Transaktionen ist

Folgende Funktionen erlauben einem Subjekt Transaktionen auszuführen und definieren aktive Rollen für ein Subjekt:

Ausführen (s:Subjekt, t:Transaktion) = {Ist wahr, wenn Subjekt s
die Transaktion t ausführen kann}

Aktive_Rollen (s:Subjekt) = {Aktuelle Liste von aktiven Rollen für Subjekt s}

(4. Regel) Rollenautorisierung:

Die aktive Rolle eines Subjekts muß eine autorisierte Rolle für dieses Subjekt sein. Ein Subjekt kann keine aktive Rolle haben, die nicht für das Subjekt autorisiert ist.

$$\forall\ s_i\text{:Subjekt}:$$

$$\text{Aktive\_Rollen}\ (s_i) \subseteq \text{Autorisierte\_Rollen}\ (s_i)$$

(5. Regel) Rollenausführung:

Ein Subjekt kann nur dann eine Transaktion ausführen, wenn das Subjekt in einer aktiven Rolle agiert.

$$\forall\ s_i\text{:Subjekt}, t_j\text{:Transaktion}:$$

$$\text{Ausführen}\ (s_i, t_j) \Rightarrow \text{Aktive\_Rollen}\ (s_i) \neq \varnothing$$

Das Verfahren des rollenbasierten Zugriffsmodells unterstützt neben der statischen Trennung von Pflichten auch dynamische Trennung von Pflichten. Diese ist etwas komplizierter als die statische Trennung von Pflichten, erlaubt jedoch mehr Flexibilität der Transaktionen. Während der Ausführung der Transaktionen wird überprüft, ob ein und derselbe Benutzer weitere Transaktionen ausführt, die er nach diesem Grundsatz nicht ausführen darf. Dafür muß das System neben der Zugehörigkeit von Benutzer zu Rollen auch die Benutzer-Kennung (User-ID) überprüfen, um den Zugriff auf Transaktionen erlauben zu können.

Abhängig von der Politik des Systems oder der Organisation, in der das Zugriffsmodell eingesetzt wird, kann es unproblematisch sein, Mitglied in zwei verschiedenen Rollen zu sein, die keinen Interessenkonflikt bewirken, wenn sie unabhängig voneinander ausgeübt werden. Jedoch kann es problematisch sein, wenn sie zur gleichen Zeit ausgeübt werden. Dynamische Trennung von Pflichten erlaubt mehr Flexibilität in den Transaktionen, so daß ein Benutzer Mitglied in zwei Rollen sein darf, jedoch nicht gleichzeitig in ihnen agieren darf.

Exklusive_Aktive_Rollen (r:Rolle) = {Liste von aktiven Rollen, die sich gegenseitig mit Rolle r ausschließen }

(6. Regel) Dynamische Trennung von Pflichten:

Ein Subjekt kann in einer neuen Rolle aktiv werden, wenn diese Rolle sich mit keiner anderen zur Zeit aktiven Rolle gegenseitig ausschließt.

$$\forall\ s_i\text{:Subjekt}, r_{j,k}\text{:Rolle}, j \neq k:$$

$$r_j \in \text{Aktive\_Rollen}\ (s_i) \wedge r_k \in \text{Aktive\_Rollen}\ (s_i) \Rightarrow r_j \notin \text{Exklusive\_Aktive\_Rollen}\ (r_k)$$

(7. Regel) Autorisierung der Transaktion:

Ein Subjekt kann eine Transaktion ausführen, wenn die Transaktion für die Rolle autorisiert ist, in der das Subjekt gegenwärtig aktiv ist.

$$\forall\, s_i\text{:Subjekt}, t_j\text{:Transaktion} \exists\, r_k\text{:Rolle} :$$

$$\text{Ausführen}\,(s_i, t_j) \Rightarrow r_k \in \text{Aktive\_Rollen}\,(s_i) \wedge t_j \in \text{Rollen\_Transaktionen}\,(r_k)$$

5.7.2.5 Operative Trennung von Pflichten

Das rollenbasierte Zugriffsmodell kann zur Unterstützung von operativer Trennung von Pflichten genutzt werden. Dafür wird eine Funktion definiert, die einen Geschäftsablauf darstellt. Dieser Geschäftsablauf besteht aus einzelnen Transaktionen, die nacheinander ausgeführt werden.

Funktion_Transaktionen (f:Funktion) = {Menge der Transaktionen, die für eine Geschäftsfunktion f erforderlich sind}

(8. Regel) Operative Trennung von Pflichten:

Jede Transaktion ist einer Rolle zugeordnet. Die einzelnen Transaktionen einer Geschäftsfunktion dürfen nicht in einer einzigen Rolle ausgeführt werden, da sonst die Gefahr des Betrugs besteht. Eine Rolle darf nur einer Transaktion einer Geschäfts-funktion zugeordnet werden, wenn diese Rolle für den Benutzer autorisiert ist und diese Rolle nicht vorher schon einer anderen Transaktion derselben Geschäftsfunktion zugeordnet war.

Zur besseren Leserlichkeit wurde die Funktion Benutzer_Autorisierte_Rollen (b_i) mit B_R (b_i) für diese Regel abgekürzt.

$$\forall b_i\text{:Benutzer}, r_j\text{:Rolle}, f_k\text{:Funktion} :$$

$$\neg \Big(\text{Funktion\_Transaktionen}\,(f_k) \subseteq \bigcup_{r \in B\_R\,(b_i)} \text{Rollen\_Transaktionen}\,(r_j) \Big)$$

5.7.2.6 Zugriff auf Objekte

Im rollenbasierten Zugriffsmodell muß der Zugriff von Subjekten auf Objekte kontrolliert werden.

Zugriff (s:Subjekt, o:Objekt) = {Ist wahr, wenn Subjekt s auf Objekt o zugreifen kann}

(9. Regel) Autorisierung des Objektzugriffs:

Ein Subjekt kann auf ein Objekt zugreifen, wenn das Subjekt eine aktive Rolle hat, in der die entsprechende Transaktion ausgeführt werden darf, und die Transaktion für das Objekt autorisiert ist.

$$\forall\ s_i\text{:Subjekt}, o_j\text{:Objekt}:$$

$$\text{Zugriff}\ (s_i, o_j) \Rightarrow \exists\ r_k\text{:Rolle}, t_m\text{:Trans}:$$

$$r_k \in \text{Aktive\_Rolle}\ (s_i) \wedge t_m \in \text{Rollen\_Transaktion}\ (r_k) \wedge o_j \in \text{Transaktion\_Objekt}\ (t_m)$$

Seit den ersten Entwicklungen des Modells 1992 und 1995 wurde es kontinuierlich weiterentwickelt. Andere Aspekte des Rollenmodells, sowie detaillierte Untersuchungen wurden von den Autoren in den letzten Jahren veröffentlicht ([Sandhu, Coyne, Feinstein, Youman 1994/1996], [Guiri 1995]). Eine ausführliche Beschreibung des Prinzips der Trennung von Pflichten durch die Definition von Rollen, die sich gegenseitig ausschließen, wurde 1997 veröffentlicht [Kuhn 1997]. Dort wird auf die Bedeutung der Trennung von Pflichten speziell im kommerziellen Bereich eingegangen und der wesentliche Unterschied zwischen statischer und dynamischer Sichtweise erläutert. Das Rollenmodell wurde 1997 für das World Wide Web für Unix und Windows NT Server implementiert [Barkley, Cincotta, Ferraiolo, Gavrilla, Kuhn 1997].

5.8 Bewertung anhand der grundlegenden Sicherheitsanforderungen

Die hier beschriebenen Sicherheitsmodelle erfüllen unterschiedliche Sicherheitsanforderungen. Zur Erinnerung werden die in Kapitel 2.2 aufgelisteten Sicherheitsanforderungen noch einmal kurz erwähnt. Anschließend werden die einzelnen Sicherheitsmodelle anhand der Sicherheitsanforderungen bewertet.

Neben den klassischen Anforderungen, wie Vertraulichkeit, Integrität, Verfügbarkeit und Verläßlichkeit, wurden außerdem Zurechenbarkeit, Verbindlichkeit oder Nichtabstreitbarkeit, sowie informationelle Selbstbestimmung, Anonymität und Pseudonymität gefordert. Dabei ist zu beachten, daß die einzelnen Sicherheitsmodelle nicht alle Anforderungen erfüllen können, sondern nur die Basis für eine sichere Anwendungsgestaltung zur Verfügung stellen können. Anforderungen wie anonyme oder pseudonyme Benutzung lassen sich nicht durch ein Sicherheitsmodell garantieren, sondern durch die entsprechende Anwendung, der das Sicherheitsmodell zugrundeliegt. Allerdings kann das Sicherheitsmodell Mechanismen zur Verfügung stellen, die verschiedene Sicherheitsanforderungen unterstützen.

Zum Beispiel unterstützt das Prinzip der Trennung von Pflichten (separation of duty) Anforderungen, die oft an kommerzielle Systeme gestellt werden. Eine sauber strukturierte Pflichtentrennung gewährleistet Vertraulichkeit und Integrität und verhindert unberechtigten Zugriff. Sicherheitsmodelle können zusätzlich Sicherheitsmaßnahmen bezüglich Verbindlichkeit, Zurechenbarkeit oder Nichtabstreitbarkeit zur Verfügung stellen, die zum Beispiel der Beweiserstellung von Handlungen dienen.

Das Recht auf informationelle Selbstbestimmung läßt sich nur indirekt bewerten, da es keine direkte Eigenschaft eines Sicherheitsmodells ist, ob ein Subjekt selbst entscheiden kann, wem es welche Informationen gibt. Es hängt sehr stark vom jeweiligen Anwendungssystem ab, welche Daten zur Benutzung einer Anwendung erforderlich sind. Es

kann jedoch bewertet werden, ob ein Sicherheitsmodell strikt alle Aktionen eines Benutzers fest vorgibt oder ob es einen Rahmen definiert, in dem sich ein Benutzer frei bewegen kann und eigene Entscheidungen bezüglich der Benutzung der Anwendungssysteme treffen kann.

Wie bereits in Kapitel 5.2 erwähnt, unterscheidet man bei der Implementierung von Zugriffsregeln üblicherweise zwischen Discretionary Access Control (DAC), bei der der Benutzer seine Zugriffsrechte selbst festlegt und verändert und Mandatory Access Control (MAC), bei der die Zugriffsrechte zentral im System gesteuert werden. Zugriffskontrolle im Sinne von DAC und MAC beschränkt sich auf Subjekte und Objekte innerhalb eines Computer Systems, nicht jedoch auf Zugriffe auf das System, die von außen kommen. Der Zugriff von außen auf das System wird unter den Stichpunkten Benutzeridentifikation und -authentisierung abgehandelt. Vernetzte Systeme, wie sie im dargestellten Anwendungsbeispiel des elektronischen Zahlungsverkehrs genutzt werden, müssen sowohl mit internen als auch mit externen Zugriffen umgehen. Aus diesem Grund reicht ein klassisches Zugriffskonzept, das DAC und MAC kombiniert, nicht aus, um die Anforderungen in offenen Systemen zu erfüllen.

Das Modell von Bell und LaPadula (siehe Kapitel 5.3) unterstützt mit seinen Zugriffsregeln und Zugriffsmatrizen Mandatory Access Control und ist in vielen gebräuchlichen Sicherheitssystemen implementiert. Die zentrale Identifizierung beim Bell-LaPadula-Modell ist wie bei MAC und DAC ein Hauptbestandteil der Mechanismen. Die Durchsetzbarkeit der Sicherheitsmechanismen beruht dadurch auf einer zentralen Sicherheitsdomäne, der alle Beteiligten unabhängig von ihren Interessen vertrauen müssen. Schon früh wurde Kritik an dieser Art von IT-Sicherheit geäußert. Besonders David Chaum hat immer wieder darauf hingewiesen, daß die Abhängigkeit von zentralen Instanzen für ein demokratisches System politische Risiken mit sich bringt.

Besonders die zentrale Identifizierung der Benutzer stellt das Recht der Bürger auf Datenschutz und alltägliche Anonymität in Frage. David Chaum hat Kommunikationsprotokolle entwickelt, die Anonymität und gleichzeitig Rechtssicherheit zur Verfügung stellen [Chaum 1985/1987]. Der Verzicht auf die Durchsetzbarkeit von Sicherheitsanforderungen wird durch dezentrale Mechanismen kompensiert, die Betrug gar nicht erst ermöglichen oder im Nachhinein aufdecken können. Die Sicherheitspolitik im Bell-LaPadula-Modell betrifft lediglich den Vertraulichkeitsaspekt. Für umfassende Sicherheitsanforderungen ist dies jedoch nicht ausreichend. Weiterhin benötigt werden Integritätsaspekte sowie Nachweisbarkeit von Handlungen, Nichtabstreitbarkeit und Anonymität.

Das Sicherheitsmodell von Clark und Wilson (siehe Kapitel 5.4) betrachtet die Integrität aus zwei Blickwinkeln: Aus dem der internen Korrektheit und dem der externen Korrektheit. Wohldefinierte Transformationen sind an Objekte gekoppelt, schützen dadurch die interne Korrektheit. Pflichtentrennung schützt die externe Korrektheit. Es ist ein erstes Modell, das die Pflichtentrennung mit einbezieht, unter Sicherheitsaspekten jedoch nur den Aspekt der Integrität berücksichtigt. An dieselben Objekte sind jedoch immer dieselben Transformationen gekoppelt, so daß eine Unterscheidung bezüglich verschiedener Personen oder gar verschiedener Aufgaben oder Rollen nicht

möglich ist. Ein Benutzer ist nur durch wohlgeformte Transformationen in seinen Möglichkeiten der Objektbearbeitung eingeschränkt. Ein Benutzer kann dieselben Objekte immer in derselben Art und Weise bearbeiten. Das Clark-Wilson-Modell beschreibt ein geschlossenes System. Für offene Systeme ist es nicht anwendbar und kommt damit als Sicherheitsmodell für multifunktionale Chipkartenanwendungen im elektronischen Zahlungsverkehr nicht in Frage.

Das Telekooperationsmodell (siehe Kapitel 5.5) kommt den gestellten Anforderungen von Verbindlichkeit und Nichtabstreitbarkeit recht nahe. Durch die Beschreibung von Rollen und Zielen von Rollen ist eine gute Flexibilität in der Gestaltung des Systems möglich. Das Telekooperationsmodell hat eine Dimension (Rollen), in der die Gestaltung erfolgen kann. Die Rollen sind eng mit den Zielen verknüpft und lassen keine Kombination von beispielsweise Rollen mit anderen Zielen zu. Dies schränkt die Flexibilität in der Kombination von Zugriffsrechten erheblich ein.

Ein weiterer Nachteil ist die Tatsache, daß das Telekooperationsmodell nicht den Aspekt der Trennung von Pflichten (Separation of Duty) unterstützt. In offenen Systemen mit vielseitigen Anwendungsmöglichkeiten ist jedoch gerade die Trennung von Pflichten ein wesentlicher Aspekt, der in Sicherheitsmodellen berücksichtigt werden muß. Da der Trend zu immer mehr Anwendungen in einem Umfeld geht, ist es absehbar, daß immer mehr Anwendungen gleichzeitig ausgeführt werden sollen. Die parallele Bearbeitung von mehreren Anwendungen ist schon jetzt am Arbeitsplatz und privat am heimischen PC zu beobachten. Damit die gleichzeitige Ausführung von verschiedenen Anwendungen kein Sicherheitsrisiko darstellt, ist es notwendig die Trennung von Pflichten zu definieren. So kann definiert werden, welche Anwendungen oder Teilfunktionen sich mit anderen gegenseitig ausschließen, da sie sich unzulässig beeinflussen.

Im formalen Datenschutzmodell (siehe Kapitel 5.6) werden Aufgaben definiert, die ein Benutzer ausführen kann. Dabei wird bei der Ausführung streng darauf geachtet, daß die Zweckbindung und Erforderlichkeit eingehalten wird. Es sind Rollen vorgesehen, die jedoch nicht zwischen verschiedenen „Endbenutzern“ unterscheiden, sondern zwischen Sicherheitsadministratoren, Datenschutzbeauftragten und Benutzern. Das bedeutet, daß der Benutzer ausschließlich durch die Zuordnung seiner erlaubten Aufgaben organisiert wird. Ähnlich wie im Rollenmodell, das statt einer Aufgabenzuordnung eine Rollenzuordnung vorsieht, wird im formalen Datenschutzmodell nur eine Dimension in der Strukturierung erreicht.

Das Rollenmodell (siehe Kapitel 5.7) erlaubt eine flexible Gestaltung der Benutzungsrechte durch das Einrichten von Rollen mit zulässigen Transaktionen. Die Zuordnung von Benutzern zu Rollen kommt der organisatorischen Sicht von Unternehmen sehr nahe, da die Mitarbeiter eines Unternehmens meist auch spezielle Rollen innehaben. So ist die Administration der Zugriffspolitik auf die Organisationsstruktur eines Unternehmens leicht abbildbar. Rollenmodelle in verschiedenen Ausprägungen [Sandhu, Coyne, Feinstein, Youman 1996] sind Alternativen zu den traditionellen diskreten (DAC) und mandatorischen Zugriffskonzepten (MAC).

Der prinzipielle Unterschied ist, daß Rollenmodelle unternehmensspezifische Sicherheitspolitiken spezifizieren und unterstützen, die sich eng an die Unternehmensstruktur anlehnen. Rollenmodelle ermöglichen auf diese Art eine Gestaltung der Sicherheit, die sich an der Unternehmensorganisation orientiert. Einfache Versionen des Rollenmodells lassen sich mit Zugriffskontrollisten (ACL) modellieren, jedoch sind ACL nicht geeignet, komplexere Rollenmodelle darzustellen [Barkley 1997].

Positiv erwähnenswert ist im Rollenmodell das Prinzip der statischen und dynamischen Trennung von Pflichten (principle of static and dynamic separation of duty) sowie das Prinzip der minimalen Rechte (principle of least privilege).

Da in dem Rollenmodell dem Benutzer jedoch ausschließlich Rollen zugeordnet sind, besitzt es nur eine Dimension. Dies ist nachteilig, da es weniger Flexibilität in der Gestaltung der Anwendung bietet.

5.9 Folgerungen für ein neues Sicherheitsmodell

Die vorhergehende Bewertung hat verdeutlicht, daß keines der vorgestellten Sicherheitsmodelle den grundlegenden Sicherheitsanforderungen genügt. Prinzipiell gibt es zwei Ansätze, diesem Problem zu begegnen. Die eine Möglichkeit besteht darin, vorhandene Sicherheitsmodelle in einem übergeordneten Rahmenwerk zu vereinen und damit alle Einzelanforderungen zu erfüllen. Die andere Möglichkeit liegt in der Entwicklung eines neuen, spezifischen Sicherheitsmodells, das entsprechend den grundlegenden Sicherheitsanforderungen spezifiziert ist.

Für die erste Möglichkeit ist der Ansatz von Abrams und anderen prinzipiell anwendbar. Er stellt einen generellen Rahmen für unterschiedliche Sicherheitsmodelle zur Verfügung [Abrams, Eggers, LaPadula, Olson 1990]. Der Generalised Framework for Access Control (GFAC) erlaubt, je nach Anforderungen das eine oder andere Sicherheitsmodell anzuwenden, also die entsprechenden Regeln der Sicherheitspolitik umzusetzen. Dies setzt jedoch voraus, daß die eingebetteten Sicherheitsmodelle sich überlagern lassen, daß eine Vereinigung der einzelnen Sicherheitsanforderungen zu einem spezifischen Anforderungsprofil möglich ist und daß keine Widersprüche entstehen. Lassen sich die einzelnen Modelle jedoch nicht überlagern, können zu einem Zeitpunkt immer nur die Anforderungen eines gerade ausgewählten Modells erfüllt werden, womit keine Verbesserung der Situation eintritt. Die spezifischen Sicherheitsanforderungen werden somit nie gesamt, sondern nur teilweise erfüllt.

Die zweite Möglichkeit, dem oben genannten Problem zu begegnen, ist die Entwicklung eines spezifischen Sicherheitsmodells, das vollständig auf die grundlegenden Sicherheitsanforderungen eingeht. Dabei ist es sinnvoll, auf bestehende Konzepte von bereits bewährten Sicherheitsmodellen zurückzugreifen, um diese mit den spezifischen Anforderungen zu kombinieren. In dieser Arbeit wird dieser zweite Ansatz gewählt und ein spezifisches Sicherheitsmodell entwickelt.

Bei dem Sicherheitsmodell, das in Kapitel 6 vorgestellt wird, soll ausgewählt werden können, ***was*** (welche Aufgabe) und ***wie*** (in welcher Rolle) etwas ausgeführt werden soll. Es baut im wesentlichen auf dem rollenbasierten Zugriffsmodell (siehe Kapitel 5.7) auf

und erweitert es um eine zweite Dimension, nämlich die Aufgaben. Die Rollen und Aufgaben sind von einander unabhängig und können nahezu beliebig miteinander kombiniert werden, abhängig vom jeweiligen Anwendungsfall. Aus der freien Kombinierbarkeit ergibt sich die gewünschte Zweidimensionalität. Weiterhin soll der Schwerpunkt nicht auf der Kooperation zwischen Personen liegen, sondern zwischen einer Person und einem technischen System, wobei wesentlich ist, daß das informationelle Selbstbestimmungsrecht dieser Person gegenüber dem System gewahrt bleibt.

Ebenso soll die Entscheidung zur Wahl einer Rolle und Aufgabe immer bei dem Benutzer liegen. Dabei macht es keinen Unterschied, ob der Benutzer erst eine Aufgabe und dann eine passende Rolle auswählt oder zuerst eine Rolle und dann eine passende Aufgabe. Da der Benutzer ein konkretes Ziel vor Augen hat, kann er auf zwei Wegen zu diesem Ziel gelangen. Je nachdem welche Rollen bei welcher Aufgabe zugelassen sind oder welche Aufgaben in welcher Rolle erledigt werden können, kann sich eine andere Auswahl an Rollen und Aufgaben bieten. Das Ziel ist jedoch am Ende auf beiden Wegen erreicht.

Im folgenden Kapitel wird deshalb ein spezifisches Sicherheitsmodell entwickelt, das eine Kombination von Rollen und Aufgaben zuläßt und dadurch dem Benutzer eine zweidimensionale Gestaltung der Zugriffsmöglichkeiten erlaubt.

6 Rollen- und Aufgabenbasiertes Sicherheitsmodell

Im vorherigen Kapitel wurden wichtige existierende Sicherheitsmodelle erläutert und bewertet. Alle vorgestellten Modelle erfüllen spezifische Sicherheitsanforderungen (siehe Kapitel 5.8). Das Bell-LaPadula-Modell garantiert Vertraulichkeit in der Kommunikation. Das Modell von Clark und Wilson unterstützt Integrität durch wohlgeformte Transformations-Prozeduren und arbeitet nach dem Prinzip der Pflichtentrennung (Separation of Duty). Das Telekooperationsmodell formuliert Rollen für Benutzer. Jede Rolle hat ein festes Ziel. Der Benutzer kann jedoch Rollen nicht mit Zielen anderer Rollen verknüpfen und ist dadurch in seiner freien Wahl von Rollen und Zielen eingeschränkt.

Das rollenbasierte Zugriffsmodell formuliert ebenfalls Rollen für die Benutzer und hat zusätzlich noch das Prinzip der Aufgabentrennung und der minimalen Rechte. Auch hier kann der Benutzer jedoch nur die Transformations-Prozeduren ausführen, die den Rollen zugeordnet sind, und er hat dadurch eine streng hierarchische Zugriffsstruktur, die wenig flexibel ist. Das formale Datenschutzmodell definiert Aufgaben für Benutzer, die zweckgebunden und nach dem Erforderlichkeitsgrundsatz ausgeführt werden müssen. Ähnlich wie beim Rollenmodell existiert eine hierarchische Zugriffsstruktur. Allen Modellen gemeinsam ist die Tatsache, daß sie eine eindimensionale Struktur haben. Dem Benutzer sind entweder Transformations-Prozeduren, Rollen oder Aufgaben zugeordnet. Die freie Entscheidung zur Wahl einer Anwendung hat ein Benutzer jedoch dann, wenn er aus mehreren Rollen und Aufgaben beliebige Kombinationen auswählen kann und in diesen Kombinationen spezifizierten Zugriff auf Datenobjekte hat. Deshalb wird ein formales Rollen- und Aufgabenbasiertes Sicherheitsmodell (R&A-Modell) entwickelt.

In den folgenden Unterkapiteln werden nach der Beschreibung der Grundidee des R&A-Modells zunächst die Begriffe geklärt, mit denen das R&A-Modell arbeitet. Als nächstes wird auf die Bedeutung des R&A-Modells eingegangen und Eigenschaften des Modells definiert. Es folgt eine formale Beschreibung des R&A-Modells als Zustandsautomat mit formaler Spezifikation, einem Zustandsdiagramm und formaler Spezifikation der Überführungsfunktionen, unabhängig von einer möglichen Implementierung. Anschließend wird eine Beweisskizze vorgestellt, mit der die Gültigkeit der Zustände nachgewiesen wird. Für eine weitere graphische Beschreibungsmöglichkeit des R&A-Modells, wird zusätzlich die Theorie der Petrinetze herangezogen, mit der besonders der dynamische Aspekt des R&A-Modells erläutert werden kann. Es können Nebenläufigkeiten und komplexe Zusammenhänge graphisch dargestellt werden.

In Kapitel 7 der vorliegenden Arbeit wird die Anwendung des R&A-Modells (R&A-Anwendung) anhand eines Beispiels aus dem elektronischen Zahlungsverkehr detailliert erläutert.

6.1 Grundidee des R&A-Modells

In diesem Kapitel wird ein zweidimensionales, formales Sicherheitsmodell entwickelt, das in seinen Grundzügen bereits veröffentlicht wurde [Schier 1998c/1998d]. Damit ist der Benutzer in der Lage, selbstbestimmt unterschiedliche Anwendungen mit verschiedenen Sicherheitsanforderungen zu nutzen, wobei Sicherheit und Datenschutz für den Benutzer wesentlich erhöht sind. Das Sicherheitsmodell stellt Aufgaben, die ein Benutzer erledigen, und Rollen, in denen er diese erledigen kann, zur Verfügung (Rollen- und Aufgabenbasiertes Sicherheitsmodell). Es erlaubt die gleichzeitige Ausführung von konfliktfreien Anwendungen. Durch das Rollen- und Aufgabenbasierte Sicherheitsmodell (R&A-Modell), können weitestgehend die grundlegenden Sicherheitsanforderungen erfüllt werden. Dadurch kann Datenschutz realisiert und eine größtmögliche Erhaltung der Privatsphäre erreicht werden. Das R&A-Modell unterstützt die grundlegenden Sicherheitsanforderungen aus Kapitel 2.2.

Das Rollen- und Aufgabenbasierte Sicherheitsmodell (R&A-Modell) baut auf dem rollenbasierten Zugriffsmodell [Ferraiolo, Cugini, Kuhn 1995] auf und erweitert es um den Aspekt der Aufgaben. Dem bisher eindimensionalen rollenbasierten Zugriffsmodell wird auf diese Art eine zweite Dimension (die Aufgaben) hinzugefügt.

Durch die jeweilige Rollen-Aufgaben-Kombination wird nicht nur der Zugriff auf Datenobjekte geregelt, sondern auch das Nutzen von Diensten oder Ressourcen verwaltet. Ebenso regelt die Rollen-Aufgaben-Kombination die Wahl der Kommunikationspartner und die Freigabe von Daten durch den Benutzer. Die Beschreibung des Sicherheitsmodells als formales Modell im nachfolgenden Kapitel beschränkt sich jedoch im Rahmen dieser Arbeit auf den Zugriff auf anwendungsspezifische Daten. Das Verwalten von Ressourcen und Kommunikationspartnern kann integriert werden, ist aber nicht Bestandteil dieser Arbeit. Das R&A-Modell beschreibt nicht, wie und durch wen autorisierte Rollen oder autorisierte Aufgaben festgelegt werden. Die Administration einer R&A-Anwendung wird nicht durch das R&A-Modell geregelt, es beschreibt ausschließlich die Benutzung einer bereits konfigurierten R&A-Anwendung.

Weiterhin wird in diesem neuen Sicherheitsmodell das Prinzip der Trennung von Pflichten verfolgt. Ähnlich wie im Rollenmodell und im Clark-Wilson-Modell wird zwischen dem Zeitpunkt der Autorisierung, also der Konfiguration des Systems und der Ausführungszeit unterschieden. Somit ergibt sich ein statischer und dynamischer Aspekt der Trennung von Pflichten, der in getrennten Regeln modelliert wird. Für statische Pflichtentrennung werden Rollen-Aufgaben-Kombinationen definiert, die sich gegenseitig ausschließen und somit nicht für einen Benutzer autorisiert sein dürfen. Dynamische Pflichtentrennung unterstützt die gleichzeitige Ausführung von konfliktfreien Anwendungen.

Das R&A-Modell kann auf unterschiedliche Weise implementiert werden, auch in Großrechnern und PCs. In dieser Arbeit wird von der Implementierung auf einer Chipkarte ausgegangen, die als Datenbasis und Zugangsmedium verwendet wird. Sie enthält alle relevanten Daten für die vorgesehenen Anwendungen. Auf der Chipkarte sind die unterschiedlichen Rollen und Aufgaben der Anwendungen gespeichert. Die

möglichen Rollen und Aufgaben werden anwendungsbezogen und entsprechend dem Sicherheitsbedürfnis definiert (siehe Kapitel 2.5). Es werden, je nach Rolle und Aufgabe, nur erforderliche Daten auf der Chipkarte freigegeben und notwendige Dienste zugelassen. Der Benutzer kann so selbst entscheiden, wen er auf seine Daten zugreifen lassen möchte und welche Dienste er benutzen möchte. Je nach Rolle und Aufgabe werden unterschiedliche Sicherheitsanforderungen unterstützt. Der Benutzer bewegt sich so auf einem individuellen Sicherheitsniveau, wie er es heute bereits aus seinem täglichen Leben kennt.

Für den Bereich des elektronischen Zahlungsverkehrs kann die R&A-Chipkarte zum einen als anonyme elektronische Geldbörse, zum anderen als Authentisierungsmittel zu netzbasierten Finanztransaktionen und des weiteren als Zugangskarte zu Informationen zum Beispiel über Geldanlagemöglichkeiten oder Zinsentwicklungen fungieren. Das bedeutet, daß sowohl die Benutzung der elektronischen Geldbörse als auch der Zugang zu netzgestützten Zahlungsverfahren, wie zum Beispiel die Verwendung von elektronischen Schecks oder elektronischen Kreditkarten und die Ausführung von Banktransaktionen, wie Überweisungen oder Daueraufträge, durch entsprechende Rollen und Aufgaben realisiert werden müssen. Die R&A-Chipkarte wird im vorgestellten Beispiel von zwei Subjekten benutzt, zum einen von dem Karteninhaber selbst, zum anderen von der Bank für administrative Zwecke, wie zum Beispiel dem Einrichten neuer bankbezogener Aufgaben.

Die R&A-Anwendung stellt dem Karteninhaber verschiedene Aufgaben, wie zum Beispiel Bezahlen, Geld akzeptieren, Kontoführungsfunktionen und ähnliches zur Verfügung. Diese Aufgaben kann er in verschiedenen Rollen, wie zum Beispiel Geldbörsenbesitzer (GB-Besitzer), EC-Kartenbesitzer (EC-Besitzer) und Kreditkartenbesitzer (KK-Besitzer) erledigen. Je nach Rollen-Aufgaben-Kombination, zum Beispiel Bezahlen als Geldbörsenbesitzer, werden nur die erforderlichen Daten freigegeben und der Zugriff auf andere Datenbereiche unterbunden. Durch die Wahl einer Rolle und Aufgabe entscheidet der Benutzer selbst, welche Information er über sich preisgibt.

6.2 Begriffsdefinitionen

Zum besseren Verständnis werden jetzt die einzelnen Begriffe des R&A-Modells definiert.

R&A-Modell: Das R&A-Modell ist ein Rollen- und Aufgabenbasiertes Sicherheitsmodell, mit dem mehrere Anwendungen (mit multifunktionalen Chipkarten) modelliert werden können.

R&A-Anwendung: Eine Anwendung ist, wie bereits in Kapitel 2 definiert, zum Beispiel das Bezahlen mit elektronischen Kreditkarten, das Bezahlen mit elektronischem Geld oder das Nutzen von Homebanking-Diensten. Eine Anwendung, die mit dem R&A-Modell konzipiert wird, heißt R&A-Anwendung.

R&A-Chipkarte: Die Implementierung von R&A-Anwendungen kann auf einer multifunktionalen Chipkarte erfolgen. Eine Chipkarte, auf der R&A-Anwendungen implementiert sind, heißt R&A-Chipkarte.

Subjekt: Ein Subjekt beschreibt, ***wer*** etwas ausführt, und repräsentiert eine natürliche oder juristische Person. Das Subjekt existiert unabhängig von spezifizierten Rollen oder Aufgaben. Ein Subjekt heißt aktives Subjekt, wenn das Subjekt im R&A-Modell etwas auswählt oder ausführt. Ein Subjekt ist zum Beispiel der Karteninhaber.

Rolle: Eine Rolle beschreibt ***wie*** oder ***in welchem Kontext*** eine Aufgabe erledigt werden soll. Die Rolle ist die Art und Weise, in der eine Aufgabe erledigt wird. Eine Rolle wird von einem Subjekt gewählt, um eine Aufgabe in einer bestimmten Art und Weise zu erledigen. Rollen existieren unabhängig von spezifizierten Aufgaben und unabhängig davon, ob jemand in einer Rolle aktiv ist. Ein Subjekt kann zum Beispiel in der Rolle des Geldbörsenbesitzers etwas erledigen.

Aufgabe: Mit der Wahl einer Aufgabe entscheidet ein Subjekt, ***was*** es erledigen möchte. Eine Aufgabe stellt somit ein Ziel dar, das erreicht werden soll. Zur Erledigung dieses Ziels in einer bestimmten Rolle wird ein Handlungsmuster angeboten, das aus einem oder mehreren Handlungsschritten besteht. Eine Aufgabe kann von einem Subjekt in verschiedenen Rollen erledigt werden. Je nach Rollen-Aufgaben-Kombination wird ein spezielles Handlungsmuster ausgeführt. Aufgaben existieren unabhängig von spezifizierten Rollen und unabhängig davon, ob jemand in einer Aufgabe aktiv ist. Ein Subjekt kann zum Beispiel die Aufgabe Bezahlen erledigen. Eine Rollen-Aufgaben-Kombination kann zum Beispiel das Bezahlen (Aufgabe) als Geldbörsenbesitzer (Rolle) sein.

Prozedur: Eine Prozedur ist Teil eines Handlungsschrittes und beschreibt ***womit*** oder ***mit welchen Mitteln*** eine Aufgabe erledigt wird. Es existiert eine Menge von Prozeduren, die auf beliebige Objekte zugreifen können. Prozeduren sind beispielsweise Lesen, Schreiben, Löschen oder ähnliches.

Objekt: Ein Objekt ist Teil eines Handlungsschrittes und ist ein Datenobjekt, auf das mittels einer Prozedur zugegriffen wird. Ein Handlungsschritt beschreibt den Zugriff einer Prozedur auf ein Datenobjekt, also auf ***wen*** oder auf ***was*** zugegriffen wird. Dieser Handlungsschritt ist Teil eines Handlungsmusters, das durch eine Rollen-Aufgaben-Kombination festgelegt ist. Ein Datenobjekt ist zum Beispiel der Guthabenstand (der Geldbörse) oder ähnliches.

Handlungsmuster: Ein Handlungsmuster beschreibt den Ablauf einer Aufgabe in einer bestimmten Rolle. Es besteht aus einzelnen Handlungsschritten, die nacheinander ausgeführt werden. Ein Handlungsmuster beschreibt durch seine einzelnen Handlungsschritte eine geordnete Liste von Prozedur-Objekt-Paaren. Durch die Elementenordnung innerhalb der Liste wird diese Liste sequentiell abgearbeitet. Abhängig von einer Rollen-Aufgaben-Kombination ist in dieser Liste definiert, welche Prozedur auf welches Objekt zugreifen darf.

Handlungsschritt: Ein einzelner Handlungsschritt ist ein Paar bestehend aus einer Prozedur und einem Objekt. Prozeduren greifen auf Objekte zu. Ein Handlungsschrit ist zum Beispiel das Prozedur-Objekt-Paar (Lesen, Guthabenstand).

Die Definitionen des Handlungsmusters und des Handlungsschritts lehnen sich an die Definition aus dem Telekooperationsmodells an [Grimm 1994].

Zu den Definitionen von Rollen und Aufgaben gibt es zwei beschreibende Adjektive. Rollen und Aufgaben können bezüglich eines Subjekts **autorisiert** und **aktuell** sein. Eine Rolle ist eine autorisierte Rolle für ein Subjekt, wenn diese Rolle von dem Subjekt prinzipiell ausgewählt werden kann. Eine Rolle ist eine aktuelle Rolle des Subjekts, wenn das Subjekt diese Rolle ausgewählt hat. Für Aufgaben gilt dies äquivalent. Rollen und Aufgaben können zu Rollen-Aufgaben-Kombinationen zusammengefaßt werden, die für ein Subjekt autorisiert und aktuell sein können.

Die Definition der Rolle im R&A-Modell ist nicht vergleichbar mit der Definition der Rolle im rollenbasierten Modell. Dort ist eine Rolle eine Menge von Transaktionen, die auf Objekte zugreifen. Die Zuordnung von Rollen zu Transaktionen ist statisch und nicht veränderbar. Die Definition der Rolle im R&A-Modell ist ebenfalls nicht vergleichbar mit der Definition der Rolle im Telekooperationsmodells.

Die Rolle im rollenbasiertes Modell und im Telekooperationsmodell sind eher vergleichbar mit einer Rollen-Aufgaben-Kombination im R&A-Modell. Das R&A-Modell stellt durch die feine Granularität der Begriffe die gewünschte Zweidimensionalität zur Verfügung, die sowohl im rollenbasierten Modell als auch im Telekooperationsmodell nicht gegeben ist.

6.3 Bedeutung des R&A-Modells

Das R&A-Modell ist ein universelles Sicherheitsmodell, das für viele verschiedene Anwendungen eingesetzt werden kann. Die in Kapitel 6.4 beschriebenen Zustandsvariablen, Regeln und Überführungsfunktionen stellen einen Rahmen für den generellen Einsatz des R&A-Modells dar. Es hängt von der jeweiligen R&A-Anwendung ab, ob alle Möglichkeiten ausgeschöpft werden, die das Modell bietet. Das bedeutet, daß unter dem Gesichtspunkt der begrenzten Ressourcen oder der einfachen Benutzbarkeit die Universalität des R&A-Modells eingeschränkt werden kann. Zum Beispiel kann in einer multifunktionalen R&A-Chipkarte auf die gleichzeitige Ausführung von R&A-Anwendungen verzichtet werden, wenn die Ressourcen der Chipkarte dies nicht ermöglichen.

Durch die Kombination von Rollen und Aufgaben, die der Benutzer nahezu beliebig miteinander kombinieren kann, handelt er je nach Rollen-Aufgaben-Kombination auf unterschiedlich sicherem Niveau. Das gewünschte Sicherheitsniveau wird durch die Wahl einer Rolle und einer Aufgabe implizit mitgewählt. Wird das R&A-Modell direkt in der Chipkarte verankert, ist die Gefahr gering, daß das gewählte Sicherheitsniveau von außen unterlaufen wird. Die Sicherheit wird in der Chipkarte, also im Endgerät der Kommunikationskette, festgelegt und realisiert. Der Benutzer kann durch die Wahl der Rolle und der Aufgabe das jeweilige Sicherheitsniveau selbst bestimmen. Die Abstufung der einzelnen Sicherheitsniveaus erfolgt rollen- und aufgabenbezogen.

Folgende Eigenschaften einer R&A-Anwendung werden erarbeitet:

- Der Benutzer kann aufgrund der individuellen Rollen und Aufgaben sein informationelles Selbstbestimmungsrecht wahrnehmen und selbst entscheiden, wem er welche Daten über sich zur Verfügung stellt. Dies setzt entweder eine aktive Beteiligung im Umgang mit der Chipkarte oder eine sinnvolle Vorkonfiguration im Sinne des Benutzers voraus.
- Jede der R&A-Anwendungen auf einer multifunktionalen R&A-Chipkarte hat definierte Rollen und Aufgaben. Je nachdem, was der Benutzer ausführen möchte, wählt er aus einer Menge von Rollen und Aufgaben eine Kombination (Rollen-Aufgaben-Kombination) aus. Jede Rollen-Aufgaben-Kombination beschreibt ein bestimmtes Handlungsmuster.
- Ein Benutzer kann gleichzeitig mehrere Rollen-Aufgaben-Kombinationen ausführen, wenn sich diese Kombinationen nicht gegenseitig ausschließen.
- Eine Voreinstellung der Rollen und Aufgaben für Standardanwendungen ist vorgesehen. Für technisch wenig interessierte Benutzer soll die R&A-Chipkarte eine Rollenvoreinstellung haben, die es dem Benutzer ermöglicht, alle R&A-Anwendungen auf einem jeweils angemessenen Standardsicherheitsniveau zu benutzen. Die Benutzung der R&A-Chipkarte soll keine aktive Administration *erfordern*, sondern gegebenenfalls *ermöglichen*.
- Es soll eine öffentliche vertrauenswürdige Instanz geben, die den Benutzer bei der Administration und Benutzung der Chipkarte unterstützt. Wenn es erforderlich ist, soll diese Instanz dem Benutzer zur Verfügung stehen. Diese Instanz soll informieren und in technischen, administrativen und rechtlichen Fragen praktische Hilfe leisten. Ihre Aktivitäten und Zugriffe werden sicher und nachvollziehbar für den Benutzer protokolliert. Sie soll unabhängig von den Anwendungsanbietern oder sonstigen beteiligten Parteien sein. Vorstellbar ist eine Institution, die vergleichbar mit einer Mischung aus Verbraucherberatung und Datenschutzbehörde ist.

Im folgenden wird das R&A-Modell formal als Zustandsautomat beschrieben. Dazu werden Variablen definiert, die einen Zustand beschreiben. Weiterhin werden Konsistenzregeln definiert, die in den einzelnen Zuständen gültig sind, sowie Überführungsfunktionen, die von einem Zustand in den nächsten überführen.

6.4 Formale Beschreibung des R&A-Modells

Das R&A-Modell wird nun formal dargestellt und exemplarisch als R&A-Anwendung im elektronischen Zahlungsverkehr beschrieben. Eine Implementierung auf einer Chipkarte (R&A-Chipkarte) ist denkbar. Bei der Beschreibung der R&A-Anwendung wird jedoch nicht auf Implementierungsdetails eingegangen.

Das R&A-Modell wird in Kapitel 7 anhand eines Anwendungsbeispiels aus dem elektronischen Zahlungsverkehr erläutert. Im folgenden wird kurz die R&A-Anwendung

beschrieben, um danach auf die formale Spezifikation des R&A-Modells einzugehen. Wie schon in Kapitel 2 ausführlich beschrieben wurde, verlagert sich der Zahlungsverkehr von konventioneller auf elektronische Zahlungsweise. Chipkarten gewinnen immer stärkere Bedeutung, sowohl als Zahlungsmedium, als auch als Zugangsmedium. Fast alle Zahlungsmodalitäten werden in Zukunft elektronisch verfügbar sein. Die Zusammenfassung verschiedener Anwendungen auf einer multifunktionalen Chipkarte führt zu verschiedenen Problemen für den Benutzer (siehe Kapitel 2). Eine multifunktionale Chipkarte, wie sie in Kapitel 2.5 beschrieben wurde, soll als Beispiel für die Anwendung des R&A-Modells verwendet werden. Eine detaillierte Beschreibung des Anwendungsbeispiels erfolgt in Kapitel 7.

Das R&A-Modell unterstützt die statische und dynamische Trennung von Pflichten (Static and Dynamic Separation of Duty). Der Benutzer kann mehrere autorisierte Rollen und Aufgaben haben, wenn sich diese Rollen und Aufgaben nicht gegenseitig ausschließen. Das Prinzip der statischen Pflichtentrennung (Static Separation of Duty) überprüft, daß nur dann mehrere Rollen oder Aufgaben oder Rollen-Aufgaben-Kombinationen für einen Benutzer autorisiert sein können, wenn sich diese nicht gegenseitig ausschließen. Zum Beispiel können die Rollen EC-Besitzer und Bankadministrator nicht für denselben Benutzer autorisiert sein.

Die dynamische Trennung von Pflichten (Dynamic Separation of Duty) regelt die gleichzeitige Ausführung von Rollen-Aufgaben-Kombinationen. Zum Beispiel kann es für einen Benutzer unkritisch sein, mehrere autorisierte Rollen oder Aufgaben oder Rollen-Aufgaben-Kombinationen zu haben, jedoch die gleichzeitige Ausführung derselben ist nicht erlaubt. Ein Benutzer kann genau dann verschiedene Rollen oder Aufgaben oder Kombinationen davon gleichzeitig ausführen, wenn sich diese nicht gegenseitig ausschließen. Zum Beispiel sind die Rollen-Aufgaben-Kombinationen (GB-Besitzer, Bezahlen) und (GB-Besitzer, Geld akzeptieren) für einen Benutzer autorisiert. Wenn sie jedoch gleichzeitig ausgeführt würden, könnten Inkonsistenzen im Guthabenstand der Geldbörse auftreten, da Abbuchen und Aufbuchen nicht gleichzeitig erfolgen dürfen. Unkritisch hingegen ist das gleichzeitige Bezahlen als GB-Besitzer und als KK-Besitzer, da hier auf unterschiedliche Datenbereiche zugegriffen wird.

Das R&A-Modell läßt sich leicht administrieren. Es ermöglicht eine einfache Erweiterung durch Definition neuer Rollen und Aufgaben und deren Integration. Dies darf in der Administratorrolle durchgeführt werden, die von den Benutzerrollen getrennt ist. Für den Benutzer entsteht so kein Administrationsaufwand. Der Benutzer kann jedoch auf eigenen Wunsch die Benutzung seiner Chipkarte konfigurieren, so daß zum Beispiel nur bestimmte Rollen oder Aufgaben verfügbar sind. Diese Konfiguration ermöglicht dann die aktive Beteiligung des Benutzers, sie ist jedoch auf keinen Fall für eine prinzipielle Benutzung erforderlich.

6.4.1 Zustandsvariablen und Funktionen

Das R&A-Modell läßt sich mit Hilfe von Zustandsvariablen beschreiben, die im nachfolgenden erläutert werden. Die Zustandsvariablen teilen sich in Subjekte, Aufgaben, Rollen, Prozeduren und Objekte auf. Jede Zustandsvariable ist als Menge

ihrer Elemente definiert. Der Zugriff auf diese Variablen erfolgt über spezielle Funktionen. Diese Funktionen liefern die Informationen, welche Elemente unter bestimmten Bedingungen zu der Menge einer Variablen gehören. Das Ergebnis jeder Funktion ist also die Menge derjenigen Elemente, für die eine angegebene Bedingung oder Fragestellung zutrifft, also eine Teilmenge der Zustandsvariablenmengen. Jeder Teilmenge können weitere Elemente zugefügt werden. Wählt ein Subjekt beispielsweise eine weitere aktuelle Aufgabe, obwohl es schon eine aktuelle Aufgabe hat, wird der Teilmenge der aktuellen Aufgaben ein weiteres Element aus der Menge Aufgaben zufügt.

Die Mengen der Rollen und Aufgaben stellen zwei voneinander unabhängige Mengen dar, deren Elemente theoretisch beliebig miteinander kombiniert werden können. Es hängt von der jeweiligen R&A-Anwendung ab, welche Aufgaben in welchen Rollen erledigt werden können.

Die Darstellung der Variablen und Funktionen im Text erfolgt kursiv. Formeln und formale Beschreibungen sind zentriert dargestellt und mit einer Randnummer versehen.

In Abbildung 6-1 wird die Mengenbeziehung zwischen *Subjekten*, *Rollen* und *Aufgaben* beschrieben. Pfeile stellen Funktionen dar, die nachfolgend erläutert werden. *AutRollen* (s_1) beschreibt die Menge der autorisierten Rollen für das angegebene Subjekt s_1. Die Menge der autorisierten Rollen ist eine Teilmenge der Menge aller Rollen. Es kann auch Rollen geben, die zur Zeit für kein Subjekt autorisiert sind.

AutAufgaben (s_1) beschreibt die Menge der autorisierten Aufgaben für das Subjekt s_1. Die Menge *AutAufgaben* (s_1) ist eine Teilmenge der Menge aller Aufgaben. Weitere Elemente der Menge *Aufgaben* gehören zu anderen Teilmengen, nämlich zu den autorisierten Aufgaben für das Subjekt s_2. Es kann auch Aufgaben geben, die zur Zeit für kein Subjekt autorisiert sind.

AutRollenUndAufgaben (s_1) beschreibt die Menge der autorisierten Rollen-Aufgaben-Kombinationen für das angegebene Subjekt. In dieser Teilmenge sind alle Kombinationen von Rollen und Aufgaben enthalten, die für ein angegebenes Subjekt s_1 autorisiert sind. Das bedeutet nicht unbedingt, daß die Menge der autorisierten Rollen-Aufgaben-Kombinationen automatisch alle Kombinationsmöglichkeiten der Elemente der Mengen von autorisierten Rollen und autorisierten Aufgaben darstellt.

Prinzipiell ist es nicht von Belang, ob zuerst eine Aufgabe zur Erledigung ausgewählt wird und danach die Rolle, in der die Aufgabe erledigt werden soll oder umgekehrt, da in beiden Fällen das Ergebnis das gleiche ist. Allerdings kann es semantische Unterschiede bei verschiedenen Reihenfolgen geben, so daß eine Wahl eventuell „mehr“ Auswahlmöglichkeiten bietet als die andere. Dies kann je nach Anwendung vorteilhaft oder nachteilig angesehen werden. Abbildung 6-1 beschreibt beide Reihenfolgen in Form einer Mengenabbildung.

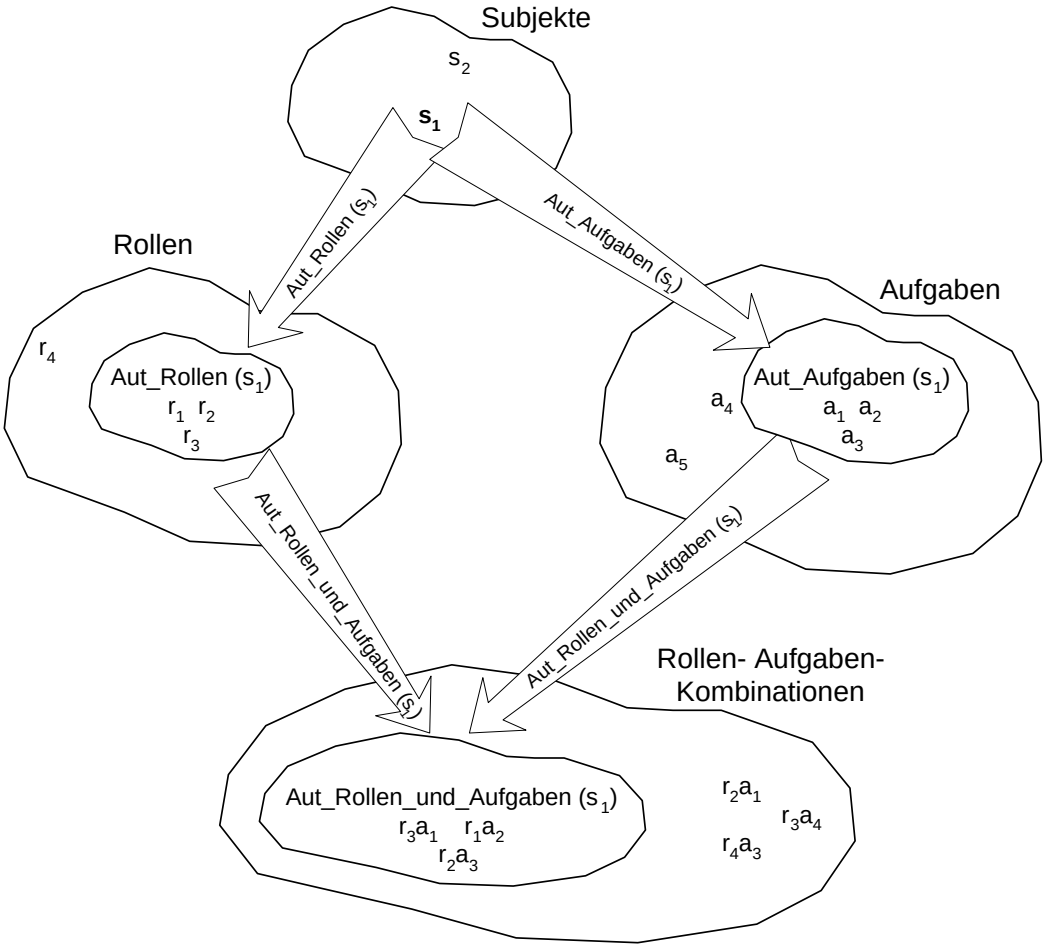


Abbildung 6-1: Autorisierte Aufgaben und Rollen für Subjekt s_1

Folgende Variablen sind zur formalen Spezifikation notwendig und können mit Hilfe der angegebenen Funktionen beschrieben werden. Funktionen beschreiben Teilmengen von Variablen.

Subjekte: *Subjekte* sind aktive Einheiten eines Systems, sie repräsentieren natürliche oder juristische Personen.

$$\text{Subjekte} = \{s_1, ..., s_n\} \quad (1)$$

Rollen: Ein Subjekt kann in mehreren *Rollen* agieren. Eine Rolle beschreibt, wie eine Aufgabe erledigt werden soll.

$$\text{Rollen} = \{r_1, ..., r_m\} \quad (2)$$

Prinzipiell gilt im R&A-Modell folgende Beziehung zwischen Subjekten und Rollen (Abbildung 6-2). Die Darstellung der Eins-zu N-Beziehung erfolgt als Entity-Relationship-Modell, wie auch in Kapitel 5.7 verwendet.

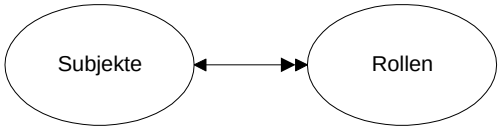


Abbildung 6-2: Beziehung zwischen Subjekten und Rollen im R&A-Modell

Für jedes Subjekt sind bestimmte Rollen autorisiert. Die Funktion *AutRollen (s_i)* beschreibt die Menge von Rollen, die für das angegebene Subjekt s_i unabhängig von einer Aufgabe autorisiert sind.

$$\text{AutRollen}(s_i) \subseteq \text{Rollen} \tag{3}$$

Die Vereinigungsmenge aller autorisierten Rollen für alle Subjekte ist eine Teilmenge der Menge aller Rollen. Es gilt:

$$\bigcup_{i = 1..n} \text{AutRollen}(s_i) \subseteq \text{Rollen} \tag{4}$$

Die Funktion *AktRollen (s_i)* beschreibt die aktuellen Rollen eines Subjekts s_i unabhängig von einer noch zu wählenden Aufgabe.

$$\text{AktRollen}(s_i) \subseteq \text{AutRollen}(s_i) \subseteq \text{Rollen} \tag{5}$$

Aufgaben: Subjekte können verschiedene *Aufgaben* auswählen, die sie erledigen möchten. Jedes Subjekt kann eine oder mehrere aktuelle Aufgaben zu einem Zeitpunkt haben, die es aus mehreren autorisierten Aufgaben auswählen kann.

$$\text{Aufgaben} = \{a_1, ..., a_k\} \tag{6}$$

Prinzipiell gilt im R&A-Modell folgende Beziehung zwischen Subjekten und Aufgaben (Abbildung 6-3).

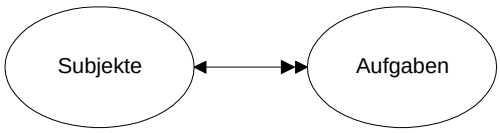


Abbildung 6-3: Beziehung zwischen Subjekten und Aufgaben im R&A-Modell

Ein Subjekt kann verschiedene Aufgaben erledigen. Die Funktion *AutAufgaben (s_i)* beschreibt die Menge der Aufgaben, die das angegebene Subjekt s_i unabhängig von einer noch zu wählenden Rolle erledigen darf.

$$\text{AutAufgaben}(s_i) \subseteq \text{Aufgaben} \tag{7}$$

Die Vereinigungsmenge aller autorisierten Aufgaben für alle Subjekte ist eine Teilmenge der Menge aller Aufgaben. Es gilt:

$$\bigcup_{i = 1..n} \text{AutAufgaben}\,(s_i) \subseteq \text{Aufgaben} \qquad (8)$$

Die Funktion *AktAufgaben (s_i)* beschreibt die Menge der aktuellen Aufgaben, die das angegebene Subjekt s_i unabhängig von einer noch zu wählenden Rolle zur Erledigung ausgewählt hat.

$$\text{AktAufgaben}\,(s_i) \subseteq \text{AutAufgaben}\,(s_i) \subseteq \text{Aufgaben} \qquad (9)$$

Rollen-Aufgaben-Kombinationen: Nach Wahl einer Rolle muß immer eine Aufgabe und nach Wahl einer Aufgabe immer eine Rolle gewählt werden. Erst nach Auswahl einer Rollen-Aufgaben-Kombination kann diese durch Ausführung des zugehörigen Handlungsmusters erledigt werden. Es können durchaus mehrere Rollen-Aufgaben-Kombinationen nacheinander gewählt werden. Um nach Wahl einer Rolle eine Aufgabe zu bestimmen, beziehungsweise nach Wahl einer Aufgabe eine Rolle, muß auch diese Rollen-Aufgaben-Kombination für das Subjekt autorisiert sein. Die Funktion *AutRollenUndAufgaben (s_i)* beschreibt die Kombinationen von Rollen und Aufgaben, die für das angegebene Subjekt s_i autorisiert sind.

$$\begin{aligned}\text{AutRollenUndAufgaben}\,(s_i) &\subseteq \text{AutRollen}\,(s_i) \times \text{AutAufgaben}\,(s_i)\\ &\subseteq \text{Rollen} \times \text{Aufgaben}\end{aligned} \qquad (10)$$

Die Elemente der Mengen von autorisierten Rollen und der Menge von autorisierten Aufgaben lassen sich in einer Matrix darstellen. Die Einträge in folgender Matrix sind willkürlich gewählt (Tabelle 6-1). Ein Eintrag in der Matrix an der Stelle j, n bedeutet, daß die Aufgabe a_j in der Rolle r_n für das Subjekt s_i autorisiert ist.

AutRollen(s_i) / **AutAufgaben(s_i)**	**(r_1)**	**(r_2)**	**(r_3)**	**(r_m)**
(a_1)	✓	✓	✓	✓
(a_2)	✓			
(a_3)		✓		✓
(a_4)	✓			✓
(a_k)		✓	✓	

Tabelle 6-1: Autorisierte Rollen und Aufgaben für ein Subjekt

Für die übrigen Einträge gilt es äquivalent. Kein Eintrag in der Matrix bedeutet, daß diese Rollen-Aufgaben-Kombination nicht für das Subjekt autorisiert ist. Die Autorisierung wird positiv definiert. Es wird also explizit definiert, was erlaubt ist. Alles andere ist automatisch verboten. Für jedes Subjekt existiert eine solche Matrix.

Die Funktion *AktRollenUndAufgaben (s_i)* beschreibt die Menge der aktuellen Rollen-Aufgaben-Kombinationen, die das angegebene Subjekt s_i zu diesem Zeitpunkt ausgewählt hat.

$$\text{AktRollenUndAufgaben}(s_i) \subseteq \text{AktRollen}(s_i) \times \text{AktAufgaben}(s_i)$$
$$\subseteq \text{Rollen} \times \text{Aufgaben} \qquad (11)$$

Autorisierung: Statische Trennung von Pflichten (Static Separation of Duty):

Ein Subjekt kann prinzipiell mehrere autorisierte Rollen haben. Ein Subjekt darf jedoch nicht dann zwei oder mehrere autorisierte Rollen haben, wenn diese Rollen sich gegenseitig ausschließen. Die Ausschlußbedingung ist symmetrisch. Die Funktion *AusschlRollen (r_j)* beschreibt alle Rollen, die sich mit der angegebenen Rolle ausschließen.

$$\text{AusschlRollen}(r_j) \subseteq \text{Rollen} \qquad (12)$$

Ein Subjekt kann prinzipiell ebenso mehrere autorisierte Aufgaben haben. Ein Subjekt darf jedoch nicht zwei oder mehrere autorisierte Aufgaben haben, die sich gegenseitig ausschließen. Die Funktion *AusschlAufgaben (a_m)* beschreibt alle Aufgaben, die sich mit der angegebenen Aufgabe a_m ausschließen.

$$\text{AusschlAufgaben}(a_m) \subseteq \text{Aufgaben} \qquad (13)$$

Für ein Subjekt können prinzipiell mehrere Rollen-Aufgaben-Kombinationen autorisiert sein. Es dürfen jedoch nur dann mehrere Rollen-Aufgaben-Kombinationen für ein Subjekt autorisiert sein, wenn sich diese Kombinationen nicht gegenseitig ausschließen.

Die Funktion *AusschlRollenUndAufgaben (r_j, a_m)* beschreibt alle Kombinationen, die sich mit der angegebenen Kombination (r_j, a_m) ausschließen.

$$\text{AusschlRollenUndAufgaben}(r_j, a_m) \subseteq \text{Rollen} \times \text{Aufgaben} \qquad (14)$$

Ausführung: Dynamische Trennung von Pflichten (Dynamic Separation of Duty):

Das gleichzeitige Agieren in mehreren Rollen ist durch die dynamische Trennung von Pflichten geregelt. Wenn sich zwei oder mehrere autorisierte Rollen nicht gegenseitig ausschließen, kann ein Subjekt auch in mehreren aktuellen Rollen gleichzeitig agieren. Nach Wahl einer aktuellen Rolle kann das Subjekt s_i eine weitere Rolle wählen, in der es gleichzeitig agiert. Die Funktion *AusschlAktRollen* (r_j) beschreibt alle Rollen, die sich mit der angegebenen aktuellen Rolle r_j unabhängig von einer noch zu wählenden Aufgabe gegenseitig ausschließen.

$$\text{AusschlAktRollen}\,(r_j) \subseteq \text{AutRollen}(r_j) \subseteq \text{Rollen} \qquad (15)$$

Wenn sich zwei oder mehrere autorisierte Aufgaben für ein Subjekt nicht gegenseitig ausschließen, kann ein Subjekt diese Aufgaben zu aktuellen Aufgaben machen und damit gleichzeitig erledigen. Nach Wahl einer Aufgabe kann das Subjekt s_i eine weitere Aufgabe wählen, die es gleichzeitig erledigt. Die Funktion *AusschlAktAufgaben* (a_m) schreibt alle Aufgaben, die sich mit der angegebenen aktuellen Aufgabe a_m unabhängig von einer noch zu wählenden Rolle gegenseitig ausschließen.

$$\text{AusschlAktAufgaben}\,(a_m) \subseteq \text{AutAufgaben}(a_m) \subseteq \text{Aufgaben} \qquad (16)$$

Es wird definiert, welche gleichzeitige Erledigung von Aufgaben in welchen Rollen unkritisch ist. Dabei ist es unerheblich, in welcher Reihenfolge die Rollen und Aufgaben ausgewählt wurden.

Die Funktion *AusschlAktRollenUndAufgaben* (r_j, a_m) beschreibt alle Rollen-Aufgaben-Kombinationen, die sich mit der angegebenen aktuellen Rollen-Aufgaben-Kombination für ein Subjekt zu diesem Zeitpunkt ausschließen. Das bedeutet, daß ein Subjekt in einer Rolle r_j und einer Aufgabe a_m die angegebenen Rollen-Aufgaben-Kombinationen nicht gleichzeitig erledigen darf.

$$\text{AusschlAktRollenUndAufgaben}\,(r_j, a_m) \subseteq \text{Rollen} \times \text{Aufgaben} \qquad (17)$$

Prozeduren: Prozeduren sind Teile der definierten Handlungsschritte einer Aufgabe und greifen auf Datenobjekte in kontrollierter Weise zu.

$$\text{Prozeduren} = \{p_1, ..., p_p\} \qquad (18)$$

Objekte: Objekte sind Datenobjekte und passive Einheiten im System.

$$\text{Objekte} = \{o_1, ..., o_q\} \qquad (19)$$

Zugriffe: Der Zugriff eines Subjekts in einer bestimmten Rolle mit einer bestimmten Aufgabe erfolgt über Prozeduren auf Datenobjekte. Zu jeder Anwendung ist definiert, welche Prozeduren in welchen Rollen-Aufgabenkombinationen auf welche Objekte angewendet werden dürfen.

Es wird positiv definiert, das heißt welche Zugriffe erlaubt sind. Die gegenteilige negative Definition, das heißt die Definition von verbotenen Zugriffen, wird in diesem Modell nicht verwendet. Der Vorteil bei der positiven Definition liegt eindeutig im klareren Verständnis. „Es ist alles erlaubt, was definiert ist“, ist einfacher zu verstehen als, „es ist alles erlaubt, was nicht definiert ist“.

Ein weiterer Vorteil liegt in der einfachen Erweiterbarkeit. Es können einfach weitere Erlaubnisse hinzugefügt werden, ohne daß die Gefahr besteht, daß ungewollte Erlaubnisse entstehen, weil nichts per Prinzip erlaubt ist. Die Menge der erlaubten Zugriffe besteht aus einer Menge von Listen von Paaren von Prozeduren und zugehörigen Objekten. Die Liste stellt eine geordnete Menge dar, beschreibt also eine Reihenfolge der Elemente.

$$\text{Zugriffe} = \{[(p_p, o_q)]_1, [(p_r, o_s)]_2, ..., [(p_t, o_u)]_v\} \quad (20)$$

Die Funktion *AutZugriff* (s_i, r_j, a_m) beschreibt den autorisierten Zugriff eines Subjektes in einer bestimmten Aufgabe und Rolle über spezielle Prozeduren auf zugehörige Datenobjekte. Das Ergebnis ist eine Liste von Prozedur-Objekt-Paaren, die aus mindestens einem Element besteht. Diese entspricht dem oben erwähnten Handlungsmuster, die Elemente der Liste den Handlungsschritten.

$$\text{AutZugriff}\ (s_i, r_j, a_m) = [(p_p, o_q)] \quad (21)$$

Die Funktion *ErlaubterZugriff*(s_i, r_j, a_m) ist genau dann wahr, wenn der Zugriff des Subjekts s_i in der Rolle r_j mit der Aufgabe a_m erlaubt ist. Entsprechend der positiven Definition der Erlaubnisse, erfolgt auch die Abfrage in positiver Form.

$$\text{ErlaubterZugriff}\ (s_i, r_j, a_m) = \{\text{wahr, wenn für das Subjekts } s_i \text{ in der Rolle } r_j \text{ mit der Aufgabe } a_m \text{ der Zugriff erlaubt ist.}\} \quad (21)$$

6.4.2 Konsistenzregeln

Die oben beschriebenen Variablen beschreiben die Zustände des R&A-Modells. Im folgenden werden Regeln definiert, die in jedem Systemzustand gelten, der durch die Variablen beschrieben wird. Einige Regeln teilen sich in Unterregeln auf, da bewußt die Freiheit der Wahl gelassen wurde, ob zuerst eine Aufgabe und dann eine Rolle gewählt wird oder umgekehrt (Abbildung 6-1).

1. Regel: Rollenautorisierung

Ein Subjekt kann nur dann in einer Rolle agieren, also eine aktuelle Rolle haben, wenn diese Rolle zu den für das Subjekt autorisierten Rollen gehört.

$$\forall\, s_i \in \text{Subjekte}:$$

$$\text{AktRollen}(s_i) \subseteq \text{AutRollen}(s_i) \tag{22}$$

2. Regel: Aufgabenautorisierung

Ein Subjekt kann nur dann eine Aufgabe erledigen, also eine aktuelle Aufgabe haben, wenn diese Aufgabe zu den für das Subjekt autorisierten Aufgaben gehört.

$$\forall\, s_i \in \text{Subjekte}:$$

$$\text{AktAufgaben}(s_i) \subseteq \text{AutAufgaben}(s_i) \tag{23}$$

3. Regel: Rollen- und Aufgabenautorisierung

Wenn bereits eine Rolle ausgewählt wurde, kann das Subjekt nur dann eine Aufgabe auswählen, wenn diese Aufgabe mit der zugehörigen Rolle zu den für das Subjekt autorisierten Rollen-Aufgaben-Kombinationen gehört. Ebenso kann das Subjekt nur dann eine Rolle, bei bereits gewählter Aufgabe, auswählen, wenn diese Rollen-Aufgaben-Kombination zu seinen autorisierten Kombinationen gehört.

$$\forall\, s_i \in \text{Subjekte}:$$

$$\text{AktRollenUndAufgaben}(s_i) \subseteq \text{AutRollenUndAufgaben}(s_i) \tag{24}$$

4. Regel: Statische Trennung von Pflichten (Static Separation of Duty)

Statische Trennung von Pflichten bezieht sich neben der Autorisierung von Aufgaben für Subjekte und der Autorisierung von Rollen für Subjekte auch auf die Autorisierung von Rollen-Aufgaben-Kombinationen für Subjekte.

4.1: Für Rollen gilt, daß eine Rolle nur für ein Subjekt autorisiert sein darf, wenn sich diese Rolle mit keiner anderen Rolle ausschließt, die für das Subjekt autorisiert ist.

$$\forall\, s_i \in \text{Subjekte},\ r_{j,k} \in \text{Rollen},\ j \neq k:$$

$$r_j \in \text{AutRollen}(s_i) \wedge r_k \in \text{AutRollen}(s_i)$$

$$\Rightarrow r_j \notin \text{AusschlRollen}(r_k) \tag{25}$$

4.2: Bezüglich der Aufgaben kann eine Aufgabe nur für ein Subjekt autorisiert sein, wenn sich diese Aufgabe mit keiner anderen Aufgabe ausschließt, die für das Subjekt autorisiert ist. Zwei Aufgaben, die für ein Subjekt autorisiert sind, dürfen sich in dieser Regel nicht gegenseitig ausschließen.

$$\forall\, s_i \in \text{Subjekte}, a_{m,n} \in \text{Aufgaben}, m \neq n :$$
$$a_m \in \text{AutAufgaben}\,(s_i) \wedge a_n \in \text{AutAufgaben}\,(s_i)$$
$$\Rightarrow a_m \notin \text{AusschlAufgaben}\,(a_n) \qquad (26)$$

4.3: Für Rollen-Aufgaben-Kombinationen gilt, daß eine Rollen-Aufgaben-Kombination nur für ein Subjekt autorisiert sein darf, wenn sich diese Rollen-Aufgaben-Kombination mit keiner anderen Kombination ausschließt, die für das Subjekt autorisiert ist. Zwei Rollen-Aufgaben-Kombinationen dürfen sich nicht gegenseitig ausschließen.

$$\forall\, s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, a_{m,n} \in \text{Aufgaben}, (r_j, a_m) \neq (r_k, a_n):$$
$$(r_j, a_m) \in \text{AutRollenUndAufgaben}\,(s_i) \wedge (r_k, a_n) \in \text{AutRollenUndAufgaben}\,(s_i)$$
$$\Rightarrow (r_j, a_m) \notin \text{AusschlRollenUndAufgaben}\,(r_k, a_n) \qquad (27)$$

5. Regel: Dynamische Trennung von Pflichten (Dynamic Separation of Duty)

Für die dynamische Trennung von Pflichten gilt sowohl die gleichzeitige Erledigung von Aufgaben als auch das gleichzeitige Agieren in mehreren Rollen. Zusätzlich wird in dieser Regel geprüft, welche aktuelle Rollen-Aufgaben-Kombinationen sich mit welcher anderen Rollen-Aufgaben-Kombination gegenseitig ausschließt.

5.1: Für Rollen gilt, daß das gleichzeitige Agieren in mehreren Rollen nur dann erlaubt ist, wenn sich die zuletzt gewählte aktuelle Rolle mit keiner der bereits aktuellen Rollen des Subjektes ausschließt.

$$\forall\, s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, j \neq k :$$
$$r_j \in \text{AktRollen}\,(s_i) \wedge r_k \in \text{AktRollen}\,(s_i)$$
$$\Rightarrow r_j \notin \text{AusschlAktRollen}\,(r_k) \qquad (28)$$

5.2: Ein Subjekt kann zu einem Zeitpunkt nur dann eine weitere aktuelle Aufgabe auswählen, wenn sich diese Aufgabe mit keiner der anderen zu diesem Zeitpunkt für das Subjekt aktuellen Aufgaben gegenseitig ausschließt. Zwei aktuelle Aufgaben, die ein Subjekt erledigen kann, dürfen sich nicht gegenseitig ausschließen.

$$\forall\, s_i \in \text{Subjekte}, a_{m,n} \in \text{Aufgaben}, m \neq n :$$
$$a_m \in \text{AktAufgaben}\,(s_i) \wedge a_n \in \text{AktAufgaben}\,(s_i)$$
$$\Rightarrow a_m \notin \text{AusschlAktAufgaben}\,(a_n) \qquad (29)$$

5.3: Wenn das Subjekt bereits eine Rolle gewählt hat und eine Aufgabe zu dieser Rolle wählt, darf sich diese Rollen-Aufgaben-Kombination mit keiner aktuellen Rollen-Aufgaben-Kombination ausschließen. Ebenso gilt für die umgekehrte Reihenfolge, daß das Subjekt bei bereits gewählter Aufgabe nur dann eine Rolle dazu wählen darf, wenn sich die ergebenden Rollen-Aufgaben-Kombination mit keiner bereits aktuellen Rollen-Aufgaben-Kombination ausschließt. Ein Subjekt darf keine Rollen-Aufgaben-Kombination mehrfach zur gleichen Zeit ausführen. Dies wird dadurch geregelt, daß sich jede Rollen-Aufgaben-Kombination prinzipiell mit selbst ausschließt.

$$\forall\, s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, a_{m,n} \in \text{Aufgaben}, \; (r_j, a_m) \neq (r_k, a_n) :$$

$$(r_j, a_m) \in \text{AktRollenUndAufgaben}\,(s_i) \wedge (r_k, a_n) \in \text{AktRollenUndAufgaben}\,(s_i)$$

$$\Rightarrow (r_j, a_m) \notin \text{AusschlAktRollenUndAufgaben}\,(r_k, a_n) \qquad (30)$$

6. Regel: Zugriffserlaubnis

Für die Anwendung werden die Zugriffe der Prozeduren auf die Objekte definiert. Diese Prozeduren werden in Abhängigkeit von der aktuellen Rolle und der aktuellen Aufgabe des Subjektes ausgeführt.

$$\forall\, s_i \in \text{Subjekte}, r_j \in \text{Rollen}, a_m \in \text{Aufgaben} :$$

$$\text{ErlaubterZugriff}\,(s_i, r_j, a_m) \Rightarrow$$

$$(r_j, a_m) \in \text{AktRollenUndAufgaben}\,(s_i) \wedge$$

$$\exists\, p_p \in \text{Prozeduren}, o_q \in \text{Objekte} :$$

$$[(p_p, o_q)] \in \text{AutZugriff}\,(s_i, r_j, a_m) \qquad (31)$$

6.4.3 Überführungsfunktionen

Die Überführungsfunktionen überführen von einem Zustand in den nächsten, wobei in jedem Zustand die entsprechenden Regeln aus Kapitel 6.4.2 erfüllt sein müssen. Die folgenden Überführungsfunktionen können vom Benutzer durch Wahl einer Aufgabe, einer Rolle oder durch Ausführen eines Handlungsmusters ausgelöst werden.

Die Zustandsüberführungsfunktionen werden in einer Art Pseudocode beschrieben, der einen Hinweis auf eine mögliche Implementierung geben soll, sich jedoch nicht wortwörtlich so programmieren läßt. Die Schlüsselwörter der Befehle des Pseudocodes sind in der Schriftart `COURIER` dargestellt. Der Begriff „Fehlerbehandlung“ steht symbolisch für eine nicht weiter ausgeführte Art der Behandlung im Falle eines Fehlers. Zum Beispiel bedeutet die Fehlerbehandlung, daß bereits in früheren Funktionen gesetzte Variablen (AktRollen oder AktAufgaben) wieder zurückgesetzt werden, falls die Prüfung der dynamischen Trennung von Pflichten nicht erfolgreich verläuft. Der Begriff „Ausführen“ steht symbolisch für das Abarbeiten eines Handlungsmusters. Das Folgeergebnis einer Funktion wird durch einen * hinter dieser Funktion beschrieben.

WähleAufgabe (s_i, a_m)

Ein Subjekt darf eine Aufgabe zur Erledigung wählen, wenn diese gewünschte Aufgabe zu der Menge von autorisierten Aufgaben für das Subjekt gehört. Das Subjekt darf nur dann eine weitere Aufgabe zur Erledigung auswählen, wenn das Subjekt zur Zeit keine aktuelle Aufgabe hat oder wenn sich diese neue Aufgabe mit keiner der bereits aktuellen Aufgaben gegenseitig ausschließt.

IF $a_m \in$ AutAufgaben (s_i)

AND

{

AktAufgaben (s_i) = $\emptyset$

OR

$\forall\ a_n \in$ AktAufgaben (s_i): $a_n \notin$ AusschlAktAufgaben (a_m)

}

THEN AktAufgaben* (s_i) = AktAufgaben (s_i) $\cup$ {a_m}

ELSE „Fehlerbehandlung“

WähleRolleNachAufgabe (s_i, r_j, a_m)

Die Auswahl einer Rolle, nachdem eine aktuelle Aufgabe gewählt wurde, ist nur dann erlaubt, wenn diese Rolle mit der aktuellen Aufgabe für das Subjekt autorisiert ist. Weiterhin darf sich diese Rolle und die bereits gewählte Aufgabe mit keiner der zur Zeit aktuellen Rollen-Aufgaben-Kombinationen des Subjektes ausschließen.

IF (r_j, a_m) $\in$ AutRollenUndAufgaben (s_i)

AND

{

AktRollenUndAufgaben (s_i) = $\emptyset$

OR

$\forall\ (r_k, a_n) \in$ AktRollenUndAufgaben (s_i), $j \neq k$:

(r_k, a_n) $\notin$ AusschlAktRollenUndAufgaben (r_j, a_m)

}

THEN AktRollen* (s_i) = AktRollen (s_i) $\cup$ {r_j}

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) $\cup$ {(r_j, a_m)}

ELSE „Fehlerbehandlung“

WähleRolle (s_i, r_j)

Ein Subjekt kann zuerst eine Rolle auswählen, in der es eine Aufgabe erledigen möchte, und anschließend die Aufgabe auswählen. Die Auswahl einer Rolle ist nur dann erlaubt, wenn diese Rolle für das Subjekt autorisiert ist. Das Subjekt darf nur dann eine weitere aktuelle Rolle wählen, wenn das Subjekt noch keine aktuelle Rolle hat oder wenn sich diese neue Rolle mit keiner der bereits aktuellen Rollen des Subjekts ausschließt.

IF $r_j \in$ Aut Rollen (s_i)

AND

{

Akt Rollen (s_i) = ∅

OR

$\forall r_k \in$ AktRollen (s_i): $r_k \notin$ AusschlAktRollen (r_j)

}

THEN AktRollen* (s_i) = AktRollen (s_i) ∪ {r_j}

ELSE „Fehlerbehandlung“

WähleAufgabeNachRolle (s_i, r_j, a_m)

Die Auswahl einer Aufgabe, nachdem eine aktuelle Rolle gewählt wurde, ist nur dann erlaubt, wenn diese Aufgabe mit der aktuellen Rolle für das Subjekt autorisiert ist. Weiterhin darf sich diese Aufgabe und die bereits gewählte Rolle mit keiner der aktuellen Rollen-Aufgaben-Kombinationen des Subjektes ausschließen.

IF (r_j, a_m)$\in$ AutRollenUndAufgaben (s_i)

AND

{

AktRollenUndAufgaben (s_i) = ∅

OR

$\forall$ (r_k, a_n) $\in$ AktRollenUndAufgaben (s_i), $m \neq n$:

(r_k, a_n)$\notin$ AusschlAktRollenUndAufgaben (r_j, a_m)

}

THEN AktAufgaben* (s_i) = AktAufgaben (s_i) ∪ {a_m}

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) ∪ {(r_j, a_m)}

ELSE „Fehlerbehandlung“

FührеAus (s_i, r_j, a_m)

Diese Funktion führt ein Handlungsmuster aus, das für ein Subjekt in einer Rollen-Aufgaben-Kombination festgelegt ist. Ein Subjekt kann eine Aufgabe in einer Rolle dann erledigen, wenn dieser Zugriff erlaubt ist, also wenn die Rolle und Aufgabe zu den für das Subjekt aktuellen Rollen-Aufgaben-Kombinationen gehören. Weiterhin muß zu der aktuellen Rollen-Aufgaben-Kombination ein Handlungsmuster in Form einer geordneten Liste von Prozedur-Objekt-Paaren existieren. Dies wird gemäß Konsistenzregel 6 durch die Funktion ErlaubterZugriff (s_i, r_j, a_m) geprüft. Nach erfolgreichem Ausführen werden die Elemente r_j aus der Menge der aktuellen Rollen, a_m aus der Menge der aktuellen Aufgaben und (r_j, a_m) aus der Menge der aktuellen Rollen-Aufgaben-Kombinationen gelöscht.

IF ErlaubterZugriff (s_i, r_j, a_m)

THEN „Ausführen“

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) \ {(r_j, a_m)}

AktRollen* (s_i) = AktRollen (s_i) \ {r_j}

AktAufgaben* (s_i) = AktAufgaben (s_i) \ {a_m}

ELSE „Fehlerbehandlung“

Die Überführungsfunktion FührеAus (s_i, r_j, a_m) führt ein spezielles Handlungsmuster aus, das für das Subjekt in der Rollen-Aufgaben-Kombination festgelegt ist. Hat ein Subjekt mehrere aktuelle Rollen-Aufgaben-Kombinationen ausgewählt, wird die Überführungsfunktion für jede einzelne Rollen-Aufgaben-Kombination ausgeführt. Damit diese Überführungsfunktionen gleichzeitig ausgeführt werden können, muß bei der Implementierung eine übergeordnete Funktion definiert werden, die für alle Rollen-Aufgaben-Kombinationen diese Überführungsfunktion parallel ausführt. Da dies jedoch von der jeweiligen Implementierung abhängt, wird auf die Definition einer solchen Funktion hier verzichtet.

Die Endlichkeit des R&A-Modells ist dadurch gegeben, daß nur eine begrenzte Anzahl von Rollen (n) und eine begrenzte Anzahl von Aufgaben (m) existiert. So ergibt sich selbst bei maximal möglicher Kombination von Rollen und Aufgaben nur eine Anzahl von (n * m) Kombinationen.

Eine mehrfache Auswahl derselben Rollen-Aufgaben-Kombination ist nicht erlaubt. Die Überführungsfunktionen prüfen diese Tatsache und lassen sie nicht zu (Prüfung der Regel dynamische Trennung von Pflichten). Weiterhin lassen sich in der jeweiligen Anwendung inhaltliche Schranken festlegen, die ebenso eine Endlichkeit garantieren. Zum Beispiel kann bei Geldabhebefunktionen am Bankautomaten durch den maximalen Abhebebetrag ein zusätzliches Limit eingerichtet sein.

6.4.4 Beweisskizze

Das R&A-Modell wurde in Kapitel 6.4 formal spezifiziert. Dazu wurden Variablen definiert, die einen Zustand beschreiben, und ebenso Regeln und Überführungsfunktionen. Es kann eine Beweisskizze geführt werden, in der gezeigt wird, daß das R&A-Modell in allen Zuständen die jeweiligen Regeln erfüllt, es sich also um gültige Zustände handelt. Weiterhin kann gezeigt werden, daß die Überführungsfunktionen von einem gültigen Zustand nur in einen wieder gültigen Zustand überführen.

In der Literatur ist es nicht üblich, Sicherheitsmodelle zu beweisen. Nur sehr wenige Sicherheitsmodelle wurden vollständig bewiesen [Bell, LaPadula 1977]. Für das rollenbasierte Zugriffsmodell wurde ein Teilaspekt des formal spezifizierten Modells ansatzweise bewiesen [Gavrila, Barkley 1998]. Auch aus der bereits in Kapitel 3.5.1 erwähnten US-Norm FIPS Pub 140 [FIPS 140-1], in der Sicherheitsanforderungen an die Erstellung von Systemen mit kryptographischen Modulen gestellt werden, geht hervor, daß selbst in der höchstmöglichen Sicherheitsstufe ein Beweis des zugrundeliegenden Zustandsautomaten nicht erforderlich ist. Die formale Spezifikation des Zustandsautomaten mit seinen Variablen, Konsistenzregeln und Überführungsfunktionen, sowie ein detailliertes Zustandsdiagramm ist ausreichend.

Ein vollständiger Beweis des R&A-Modells ist demnach nicht notwendig, es soll jedoch an dieser Stelle trotzdem eine Beweisskizze durchgeführt werden, um die Idee des Beweises zu verdeutlichen. Die hier geführte Beweisskizze geht ähnlich vor wie der Beweisansatz von Gavrila und Barkley für das rollenbasierte Modell.

Die Beweisskizze wird nach dem Prinzip der vollständigen Induktion geführt. Dies erfolgt in drei Schritten [Bronstein, Semendjajew 1989]:

1) Induktionsanfang: Zu zeigen ist, daß der Anfangszustand ein gültiger Zustand ist.
2) Induktionsannahme: Annahme, daß ein Zustand z_s gültig ist.
3) Induktionsschritt: Zu zeigen ist, daß der Nachfolgezustand z_{s+1} ein gültiger Zustand ist.

Die Menge der Zustände wird folgendermaßen beschrieben:

$$\text{Zustände} = \{z_1, ..., z_s\}$$

Ein Zustand z_s wird durch alle Zustandsvariablen beschrieben:

z_s = {Subjekte, Rollen, Aufgaben,
AutRollen, AktRollen,
AutAufgaben, AktAufgaben,
AutRollenUndAufgaben, AktRollenUndAufgaben,
AusschlAufgaben, AusschlRollen,
AusschlRollenUndAufgaben,
AusschlAktRollen, AusschlAktAufgaben,
AusschlAktRollenUndAufgaben,
Prozeduren, Objekte,
Zugriffe, AutZugriff, ErlaubterZugriff}

Der Anfangszustand z_0 hat folgende Form:

z_0 = {Subjekte, Rollen, Aufgaben,
AutRollen, AktRollen,
AutAufgaben, AktAufgaben,
AutRollenUndAufgaben, AktRollenUndAufgaben,
AusschlAufgaben, AusschlRollen,
AusschlRollenUndAufgaben,
AusschlAktRollen, AusschlAktAufgaben,
AusschlAktRollenUndAufgaben,
Prozeduren, Objekte,
Zugriffe, AutZugriff, ErlaubterZugriff }

Für den Anfangszustand z_0 gilt

$$\forall s_i \in \text{Subjekte} :$$

$$\text{AktRollen}(s_i) = \varnothing$$

$$\text{AktAufgaben}(s_i) = \varnothing$$

$$\text{AktRollenUndAufgaben}(s_i) = \varnothing$$

Alle anderen Mengen sind nicht leer.

6.4.4.1 Induktionsanfang

Es ist zu zeigen, daß der Anfangszustand z_0 ein gültiger Zustand ist. Ein Zustand ist dann gültig, wenn die Konsistenzregeln 1 bis 6 (siehe Kapitel 6.4.2) erfüllt sind.

Konsistenzregel 1: Rollenautorisierung

$$\forall s_i \in \text{Subjekte} :$$

$$\text{AktRollen}(s_i) \subseteq \text{AutRollen}(s_i),$$

$$\text{weil AktRollen}(s_i) = \varnothing$$

Konsistenzregel 2: Aufgabenautorisierung

$$\forall\, s_i \in \text{Subjekte}:$$
$$\text{AktAufgaben}(s_i) \subseteq \text{AutAufgaben}(s_i),$$
$$\text{weil AktAufgaben}(s_i) = \varnothing$$

Konsistenzregel 3: Rollen- und Aufgabenautorisierung

$$\forall\, s_i \in \text{Subjekte}:$$
$$\text{AktRollenUndAufgaben}(s_i) \subseteq \text{AutRollenUndAufgaben}(s_i),$$
$$\text{weil AktRollenUndAufgaben}(s_i) = \varnothing$$

Konsistenzregel 4: Statische Trennung von Pflichten (Static Separation of Duty)

4.1 Rollen:

$$\forall\, s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, j \neq k:$$
$$r_j \in \text{AutRollen}(s_i) \wedge r_k \in \text{AutRollen}(s_i)$$
$$\Rightarrow r_j \notin \text{AusschlRollen}(r_k),$$
$$\text{per definitionem der Funktion AusschlRollen (r)}$$

4.2 Aufgaben:

$$\forall\, s_i \in \text{Subjekte}, a_{m,n} \in \text{Aufgaben}, m \neq n:$$
$$a_m \in \text{AutAufgaben}(s_i) \wedge a_n \in \text{AutAufgaben}(s_i)$$
$$\Rightarrow a_m \notin \text{AusschlAufgaben}(a_n),$$
$$\text{per definitionem der Funktion AusschlAufgaben (a)}$$

4.3 Rollen-Aufgaben-Kombinationen:

$$\forall\, s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, a_{m,n} \in \text{Aufgaben}, (r_j, a_m) \neq (r_k, a_n):$$
$$(r_j, a_m) \in \text{AutRollenUndAufgaben}(s_i) \wedge$$
$$(r_k, a_n) \in \text{AutRollenUndAufgaben}(s_i)$$
$$\Rightarrow (r_j, a_m) \notin \text{AusschlRollenUndAufgaben}(r_k, a_n),$$
$$\text{per definitionem der Funktion AusschlRollenUndAufgaben (r, a)}$$

Konsistenzregel 5: Dynamische Trennung von Pflichten (Dynamic Separation of Duty)

5.1 Rollen:

$$\forall s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, j \neq k :$$
$$r_j \in \text{AktRollen}(s_i) \wedge r_k \in \text{AktRollen}(s_i)$$
$$\Rightarrow r_j \notin \text{AusschlAktRollen}(r_k),$$
$$\text{weil AktRollen}(s_i) = \varnothing$$

5.2 Aufgaben:

$$\forall s_i \in \text{Subjekte}, a_{m,n} \in \text{Aufgaben}, m \neq n :$$
$$a_m \in \text{AktAufgaben}(s_i) \wedge a_n \in \text{AktAufgaben}(s_i)$$
$$\Rightarrow a_m \notin \text{AusschlAktAufgaben}(a_n),$$
$$\text{weil AktAufgaben}(s_i) = \varnothing$$

5.3 Rollen-Aufgaben-Kombinationen:

$$\forall s_i \in \text{Subjekte}, r_{j,k} \in \text{Rollen}, a_{m,n} \in \text{Aufgaben}, (r_j, a_m) \neq (r_k, a_n):$$
$$(r_j, a_m) \in \text{AktRollenUndAufgaben}(s_i) \wedge$$
$$(r_k, a_n) \in \text{AktRollenUndAufgaben}(s_i)$$
$$\Rightarrow (r_j, a_m) \notin \text{AusschlAktRollenUndAufgaben}(r_k, a_n),$$
$$\text{weil AktRollenUndAufgaben}(s_i) = \varnothing$$

Konsistenzregel 6: Zugriffserlaubnis

$$\forall s_i \in \text{Subjekte}, r_j \in \text{Rollen}, a_m \in \text{Aufgaben} :$$
$$\text{ErlaubterZugriff}(s_i, r_j, a_m) \Rightarrow$$
$$(r_j, a_m) \in \text{AktRollenUndAufgaben}(s_i) \wedge$$
$$\exists p_p \in \text{Prozeduren}, o_q \in \text{Objekte} :$$
$$[(p_p, o_q)] \in \text{AutZugriff}(s_i, r_j, a_m) \quad ,$$
$$\text{weil AktRollenUndAufgaben}(s_i) = \varnothing, \text{ ist ErlaubterZugriff} = \{\text{falsch}\}$$
$$\text{das heißt, der Zugriff ist nicht erlaubt}$$

Damit ist der Anfangszustand z_0 im Sinne der sechs Konsistenzregeln gültig.

6.4.4.2 Induktionsannahme

Annahme, der Zustand z_s ist im Sinne der sechs Konsistenzregeln gültig.

6.4.4.3 Induktionsschritt

Es ist zu zeigen, daß der Folgezustand z_{s+1} im Sinne der sechs Konsistenzregeln gültig ist.

Für diesen Schritt wird eine allgemeine Überführungsfunktion (ÜF) definiert. Diese Überführungsfunktion führt ganz allgemein von einem Zustand in den nächsten, wobei die Menge der Operationen die Menge der Überführungsfunktionen aus Kapitel 6.4.3 darstellt und $2^{Parameter}$ die Menge der dafür benötigten Eingabeparameter ist.

$$ÜF : Zustände \times Operationen \times 2^{Parameter} \rightarrow Zustände$$

Die Menge Operationen besteht aus den bereits spezifizierten Überführungsfunktionen (WähleRolle, WähleAufgabeNachRolle, WähleAufgabe, WähleRolleNachAufgabe, FühreAus) und sieht folgendermaßen aus:

Operationen = $\{op_1, ..., op_x\}$

Die Menge Parameter besteht aus den Eingabeparametern, die für die jeweiligen Überführungsfunktionen erforderlich sind.

Parameter = $\{para_1, ..., para_y\}$

Es muß gezeigt werden, daß die allgemeine Überführungsfunktion ÜF die sechs Konsistenzregeln erfüllt.

Annahme: Wenn z_s ein gültiger Zustand ist und die Parameter $para_w, .., para_y$ der Operation op_x die Vorbedingungen aus der Operation erfüllen, dann gilt

$$\forall\ z_s \in Zustände,\ op_x \in Operationen,\ para_w, .., para_y \in Parameter:$$

$$z_{s+1} = ÜF\ (z_s, op_x, para_w, .., para_y)$$

und z_{s+1} ist ein gültiger Zustand im Sinne der sechs Konsistenzregeln.

Der Induktionsschritt wird nun anhand einer Überführungsfunktion aus Kapitel 6.4.3 gezeigt. Die Gültigkeit des Zustands z_{s+1} wird am Beispiel der Überführungsfunktion WähleRolle gezeigt. Der Beweis ist für die restlichen Überführungsfunktionen äquivalent.

WähleRolle

Parameter: s_i, r_j

Aktion: AktRollen* (s_i) = AktRollen (s_i) $\cup$ {r_j}

Vorbedingung 1: $r_j \in$ AutRollen (s_i)

Vorbedingung 2: AktRollen (s_i) = $\varnothing \vee$
$\forall r_k \in$ AktRollen (s_i) : $r_k \notin$ AusschlAktRollen (r_j)

Es wird angenommen, daß die Parameter $s_i \in$ Subjekte und $r_j \in$ Rollen die beiden Vorbedingungen der Überführungsfunktion erfüllen.

Es wird nun gezeigt, daß nach Ausführung der Funktion WähleRolle die Konsistenzregeln weiterhin erfüllt sind.

Konsistenzregel 1: Es ist zu zeigen, daß nach Ausführung von WähleRolle gilt:

$$\text{AktRollen*}\ (s_i) \subseteq \text{AutRollen}\ (s_i)$$

Man kann zwei Fälle unterscheiden: Entweder hat das Subjekt bisher keine aktuellen Rollen (AktRollen (s_i) = $\varnothing$) oder es hat bereits aktuelle Rollen.

1. Im ersten Fall ist die Konsistenzregel erfüllt, da $r_j \in$ AutRollen (s_i) gilt und AktRollen (s_i) genau dieses Element r_j enthält.

2. Im zweiten Fall hat die Menge AktRollen (s_i) bereits Elemente. Da man davon ausgeht, daß im Zustand z_s die Konsistenzregeln gelten, gilt

$$\text{AktRollen}\ (s_i) \subseteq \text{AutRollen}\ (s_i)$$

Es wird nun der Menge AktRollen (s_i) das Element r_j hinzugefügt. Vorbedingung 1 stellt sicher, daß $r_j \in$ AutRollen (s_i). Damit ist diese Konsistenzregel gültig und es gilt die Aussage:

$$\text{AktRollen*}\ (si) \subseteq \text{AutRollen}\ (s_i)$$

Die Konsistenzregeln 2, 3, 4, 5 und 6 hängen in keiner Weise von der Überführungsfunktion WähleRolle ab. Wenn diese Regeln im Ausgangszustand z_s gültig sind, gelten sie ebenso nach Ausführung von WähleRolle in dem Nachfolgezustand z_{s+1}.

q.e.d.

6.4.5 Zustandsdiagramm

Neben der formalen Beschreibung des R&A-Modells soll eine graphische Darstellung erfolgen. Durch eine graphische Darstellung können die Zustände und die Abläufe im R&A-Modell verständlich und transparent erläutert werden. Dabei erfolgt die

Darstellung mit Hilfe eines Zustandsdiagramms. Es werden die einzelnen Zustände und Überführungsfunktionen für die Wahl einer einzigen Rollen-Aufgaben-Kombination und deren Ausführung graphisch dargestellt. Nachfolgendes Zustandsdiagramm (Abbildung 6-4) verdeutlicht die Zustände und Überführungsfunktionen des R&A-Modells, die in Kapitel 6.4 definiert wurden. Dabei wird zwischen erfolgreich ausgeführten und nicht erfolgreich ausgeführten Zustandsüberführungsfunktionen unterschieden. Der Fall einer nicht erfolgreich ausgeführten Überführungsfunktion unterteilt sich in unterschiedliche Fehlerfälle und wird in der formalen Beschreibung der Überführungsfunktionen in Kapitel 6.4.3 mit dem Hinweis „Fehlerbehandlung“ nicht weiter unterschieden. Die „Fehlerbehandlung“ wird als Teil der Implementierung angesehen und aus diesem Grund nicht weiter vertieft.

Das Zustandsdiagramm beginnt mit dem Startzustand, in dem das Subjekt bereits authentisiert ist. Der Prozeß der Authentisierung wird nicht im Zustandsdiagramm beschrieben. Vor dem Startzustand sind neben der Authentisierung noch weitere Zustände denkbar, die die Initialisierung und Personalisierung der Chipkarte beschreiben (siehe Kapitel 3.4). Diese Phasen sollen hier jedoch nicht dargestellt werden. Durch das Zustandsdiagramm wird exemplarisch die Auswahl einer Rollen-Aufgaben-Kombination und deren Ausführung dargestellt, unabhängig davon, ob zuerst eine Rolle und danach eine Aufgabe gewählt wird oder umgekehrt. Der Startzustand ist gleichzeitig auch der Endzustand, der nach der Ausführung eines Handlungsmusters wieder erreicht wird.

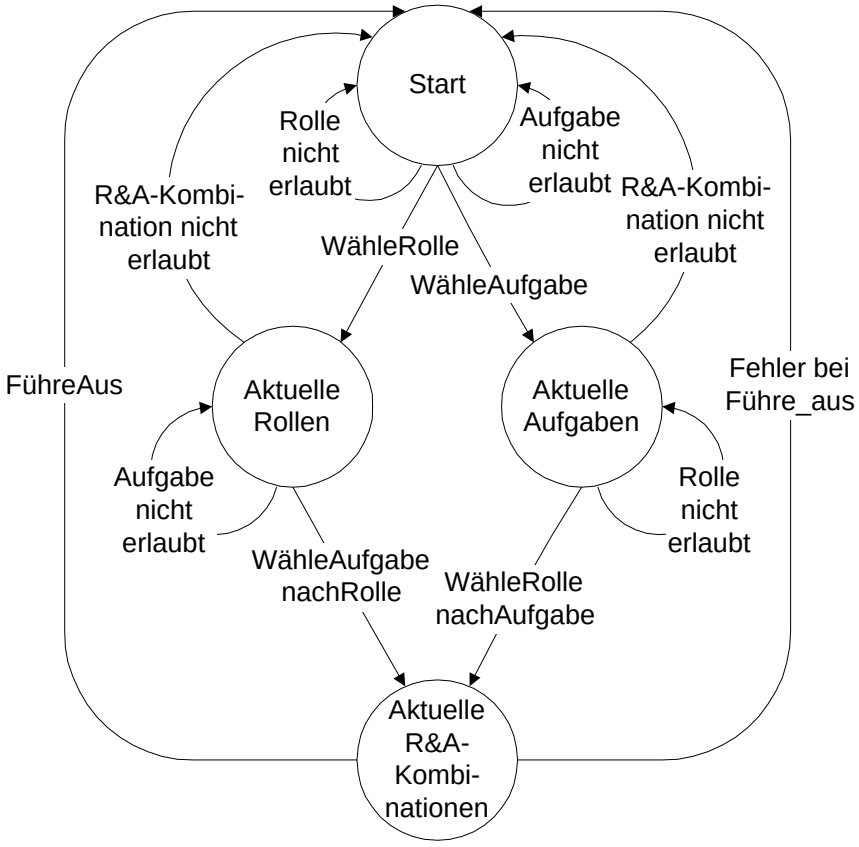


Abbildung 6-4: R&A-Modell als Zustandsdiagramm

In jedem Zustand ist es möglich, den Vorgang abzubrechen und in den Startzustand zurückzukehren, dies ist in der Graphik aus Gründen der Übersichlichkeit nicht eingezeichnet. Das Subjekt kann dann entweder den Vorgang ganz abbrechen oder eine neue Auswahl von Rollen und Aufgaben treffen.

Das Zustandsdiagramm wird zur Verdeutlichung des prinzipiellen Ablaufs einer Auswahl von einer Rolle und einer Aufgabe verwendet. Es beschreibt ausschließlich die Wahl einer einzigen Rollen-Aufgaben-Kombination und deren Ausführung. Es beschreibt nicht die Wahl mehrerer Kombinationen, die dann gleichzeitig ausgeführt werden können.

Die graphische Darstellungsmöglichkeiten von Zustandsdiagrammen sind mit einer solchen Darstellung weitestgehend erschöpft. Komplexere Darstellungen, wie das Auswählen weiterer Rollen und Aufgaben, sind mit Zustandsdiagrammen schwer möglich. Besonders schwierig darzustellen ist das gleichzeitige Ausführen von mehreren Handlungsmustern, nachdem verschiedene Rollen-Aufgaben-Kombinationen gewählt wurden.

In der graphischen Darstellung ist es weiterhin nur schwer möglich, Bedingungen darzustellen, die an einen Übergang geknüpft sind. Weiterhin können die Mengen nicht explizit in der Graphik beschrieben werden, die sich durch den Übergang von einem Zustand in den nächsten ergeben.

Die Tatsache, daß das Zustandsdiagramm nur eine begrenzte Zahl von Informationen bereitstellt, trägt jedoch wesentlich zur Übersichtlichkeit bei. Aus eben diesem Grund wird diese Darstellungsmethode für diesen einfachen Fall verwendet. Für die Darstellung der Auswahl einer einzigen Rollen-Aufgaben-Kombination ist das Zustandsdiagramm völlig ausreichend und wesentlich übersichtlicher als Petrinetze, die durch ihre komplexe Darstellungsweise mehr Informationen graphisch beschreiben können.

Um komplexere Zusammenhänge graphisch darzustellen, können Petrinetze verwendet werden. Ebenfalls kann das Auswählen mehrerer Rollen-Aufgaben-Kombinationen und das gleichzeitige Ausführen von Handlungsmustern (Nebenläufigkeit) mit Petrinetzen graphisch dargestellt werden. Zusätzlich zu der Darstellung des Zustandsdiagramms soll das R&A-Modell hier nun als Petrinetz graphisch dargestellt werden, wobei kein Anspruch auf eine vollständige Darstellung erhoben wird. Das Petrinetz ist lediglich eine weitere Darstellungsmöglichkeit und dient nicht als Beweisunterstützung des formalen R&A-Modells. In weiterführenden Arbeiten können Simulationswerkzeuge für Petrinetze eingesetzt werden, mit denen das R&A-Modell verifiziert werden kann. Eine Simulation hätte den Rahmen dieser Arbeit gesprengt.

6.5 Zusätzliche Darstellung des R&A-Modells als Petrinetz

Petrinetze bieten eine komplexe Beschreibungsmöglichkeit. Ein Petrinetz ist ein abstraktes Modell des Informations- oder Objektflusses, mit dessen Hilfe Systeme und Prozesse auf unterschiedlicher Abstraktionsebene beschrieben werden können

[Starke 1990]. Die Vorteile von Petrinetzen bestehen darin, daß durch ihre graphische Darstellungsmethodik die Anschaulichkeit erhöht wird und daß neben der Modellierung von Abhängigkeiten auch die Nebenläufigkeit von Handlungen visualisiert werden kann. Die dabei entstehenden Modelle können durch einfache Verfeinerungs- oder Vergröberungsoperationen miteinander verbunden werden.

Mit Hilfe der Petrinetztheorie können Modelle verifiziert werden und mit bestehenden Softwarepaketen simuliert und analysiert werden. Analysewerkzeuge für Petrinetze erlauben es, Deadlocks zu erkennen und damit Aussagen über die Lebendigkeit von Petrinetzen zu treffen. Für die Implementierung können Simulationswerkzeuge helfen, Petrinetze zu überprüfen und Abläufe zu demonstrieren. Petrinetze können als Proptotypen in der Softwareentwicklung verwendet werden.

Für die vorliegende Arbeit wird der Schwerpunkt jedoch ausschließlich auf die graphische Darstellungsweise gelegt. Diese Darstellungsweise dient als Ergänzung zur Beschreibung des R&A-Modells als Zustandsdiagramm, sowie der Vertiefung des Verständnisses. Aus der graphischen Darstellung der Petrinetze lassen sich Analyse-aussagen treffen, die direkt aus der Grafik abgelesen werden können. So läßt sich beispielsweise im Vergleich zum Zustandsdiagramm nicht nur der mögliche Ablauf skizzieren, sondern es können die einzelnen Zustände mit ihren Bedingungen bereits in der Grafik aussagekräftig beschrieben werden.

Ein Petrinetz-System besteht aus Elementen, die mit den Zustandsvariablen vergleichbar sind ([Starke 1990] S. 21ff). Diese Elemente werden Stellen oder Plätze genannt, die graphisch als Kreise dargestellt werden. Diese enthalten Marken, die den jeweiligen Systemzustand beschreiben. Ferner besteht das Netzmodell aus Elementen, die die Aktivitäten des Systems (Zustandsübergänge) darstellen. Sie werden Transitionen genannt und graphisch als Rechtecke dargestellt. Transitionen haben nur lokale Auswirkungen auf das System, beeinflussen also nur einige Zustandsparameter des Systems.

Die Beziehung zwischen Stellen und Transitionen wird im Netzmodell durch die Flußrelation dargestellt, graphisch als Bogen. Ein Bogen kann nur von einer Stelle zu einer Transition gehen oder von einer Transition zu einer Stelle. Bei korrekter Modellierung gibt es keine direkten Abhängigkeiten zwischen zwei Transitionen oder zwei Stellen, so daß ein Bogen niemals zwei Stellen oder zwei Transitionen miteinander verbindet.

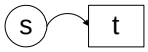
Ein Bogen von einer Stelle s zu einer Transition t zeigt an, daß s eine Vorbedingung oder eine Vorstelle von t ist. Die Ausführbarkeit von t hängt vom Erfülltsein aller Vorbedingungen von t ab. Ein Bogen von einer Transition t zu einer Stelle s bezeichnet s als die Nachbedingung oder Nachstelle von t.

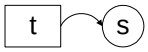
Formal wird ein Petrinetz wie folgt definiert ([Jantzen, Valk 1998] S. 313).

Definition 6.1 (Klassisches Petrinetz)

Ein Petrinetz ist ein gerichteter Graph mit zwei Sorten von Knoten, den Stellen oder Plätzen und den Transitionen. Die Kanten dürfen stets nur entweder von Stellen zu Transitionen oder von Transitionen zu Stellen gezeichnet werden. Es gilt:

1) $N = (S, T, F)$ heißt (Petri-) Netz, wobei
2) S Menge von Stellen und
3) T Menge von Transitionen ist, mit $S \cap T = \varnothing$ und $S \cup T \neq \varnothing$.
4) $F \subseteq (S \times T) \cup (T \times S)$ ist die Flußrelation, die angibt, welche Pfeile (gerichtete Kanten) existieren.

   $(s, t) \in F$ wird notiert als: 

   $(t, s) \in F$ wird notiert als: 

Eine Markierung eines Netzes N ist eine Abbildung m, die jeder Stelle s eine Anzahl von Marken m (s) zuordnet.

Aufbauend auf dieser Basisdefinition lassen sich gefärbte Petrinetze (GPN) beschreiben, die als „höhere“ Netztypen bezeichnet werden können ([Starke 1990] S. 222ff). Sie unterscheiden sich von den klassischen Petrinetzen dadurch, daß sie in ihren Stellen Marken verschiedener Sorten beinhalten können. Die Unterscheidungsmerkmale für Marken werden Farben genannt. Bei einem gefärbten Netz ist jeder Stelle s eine endliche Menge von Markensorten (Stellenfarben) und jeder Transition t eine endliche Menge von Schaltmodi (Transitionsfarben) zugeordnet.

Eine Menge, deren Elemente mehrfach verwendet werden können, nennt man Multimenge. Die Funktion M beschreibt die Häufigkeit der Elemente in der Menge X, wobei die Häufigkeit ein Element der Menge der natürlichen Zahlen $\mathbb{N}$ ist:

$$M : X \rightarrow \mathbb{N}$$

wobei: $M(x_1) = n$

mit $x_1 \in X$

und $n \in \mathbb{N}$

Definition 6.2 (Gefärbtes Petrinetz)

Das Tupel $GPN = [S, T, F, C, V, m_0]$ wird gefärbtes Netz genannt, wenn

1) (S, T, F) ein Netz ist,
2) C eine Abbildung ist, die jedem Knoten $x \in S \cup T$ eine endliche nicht leere Menge $C(x)$ von Farben zuordnet,
3) V eine Abbildung ist, die jeder Kante $f \in F$ eine Abbildung $V(f)$ von $C(t)$ in die Menge aller Multimengen über $C(s)$ zuordnet, wobei s die Stelle und t die Transition an der Kante f ist, und

4) m_0 eine Abbildung ist, die jeder Stelle s eine Multimenge, die Anfangsmarkierung, m_0 (s) über C (s) zuordnet.

Für die Farbfunktion C (s) gilt:

C_M (s) beschreibt die Multimenge der Farbfunktion über der Stelle s.

Die Spezifikation des Petrinetzes erfolgt in enger Anlehnung an die Beschreibung des R&A-Modells als Zustandsdiagramm, jedoch mit der vereinfachten Schreibweise, daß das Handlungsmuster, das im Zustandsdiagramm als geordnete Liste von Prozedur-Objekt-Paaren beschrieben wird, hier kurz als Menge (HMuster) von Handlungsmustern (h) wiedergegeben wird.

$$HMuster = \{h_1, ..., h_r\}$$

Dadurch vereinfacht sich die Darstellung des Petrinetzes erheblich. Ein Element der Menge HMuster kann somit dargestellt werden als $h_a = [(p_p, o_q)]$.

Die Abbildung 6-6 und beschreibt das R&A-Modell in Form eines gefärbten Petrinetzes, wobei sich die Stellen und Transitionen an der formalen Beschreibung aus Kapitel 6.4 orientieren. Prinzipiell werden im Text *Stellen* und *Transitionen* kursiv dargestellt. Wenn ein Pfeil zwei Pfeilspitzen hat, bedeutet das, daß das Element aus der Stelle herausgeholt und anschließend wieder hineingelegt wird. Zur einfacheren Darstellung wird anstelle von zwei Pfeilen nur ein Pfeil jedoch mit zwei Pfeilspitzen verwendet.

Die Phase der Authentisierung läßt sich im Petrinetz relativ einfach im Vergleich zum Zustandsdiagramm darstellen (Abbildung 6-5). Eine Semaphore in Form der Stelle *Subjekt in Aktion* stellt sicher, daß immer nur ein Subjekt zu einem Zeitpunkt aktiv ist. Dies wird in nachfolgender Abbildung dargestellt.

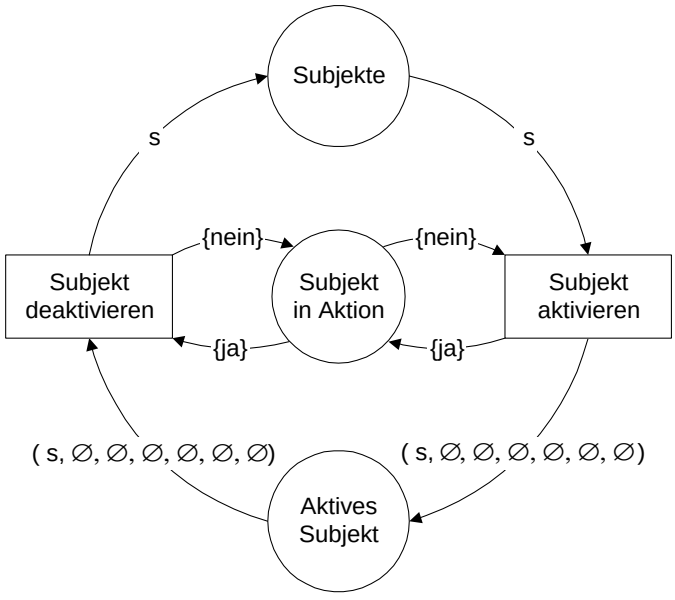


Abbildung 6-5: Authentisierung eines Subjekts

Die Stelle *Aktives Subjekt* ist die Schnittstelle zwischen Abbildung 6-5 und Abbildung 6-6. Das Petrinetz beginnt ebenso wie das Zustandsdiagramm mit seiner Darstellung damit, daß ein Subjekt bereits authentisiert, also aktiv ist (Abbildung 6-6). Die Stelle *Aktives Subjekt* hat somit immer genau ein Element, das aktive Subjekt.

Bevor auf das eigentliche Petrinetz eingegangen wird, erfolgt eine kurze Beschreibung der verwendeten Stellen, Anfangsmarkierungen und Transitionen aus Abbildung 6-5.

Stellen der Authentisierung:

Subjekte: *Subjekte* beschreibt die Menge aller Subjekte, die sich im System befinden. Die Farbfunktion der Stelle *Subjekte* lautet: C (*Subjekte*) = $\{s_1, ..., s_n\}$ und beschreibt alle Elemente der Menge.

Subjekt in Aktion: *Subjekt in Aktion* ist eine Stelle, die verhindert, daß mehr als eine Marke die Stelle Subjekte verläßt. Die Markierung der Stelle besteht aus den beiden Marken „{nein}“ und „{ja}“, die Farbfunktion lautet demnach C (*Subjekt in Aktion*) = {nein, ja}.

Die Stelle *Aktives Subjekt* wird im Zusammenhang mit der Petrinetz-Darstellung des R&A-Modells erläutert (Abbildung 6-6). Im Vergleich zum Zustandsautomat vereint die Stelle *Aktives Subjekt* die drei Zustände des Zustandsdiagramms Aktuelle Rollen, Aktuelle Aufgaben und Aktuelle Rollen-Aufgaben-Kombinationen. Die unterschiedlichen Zustände werden in den Farben der Stelle *Aktives Subjekt* vewaltet.

Anfangsmarkierung der Authentisierung

- Die Elemente der Menge *Subjekte* sind alle Subjekte, die im Modell vorhanden sind.

  m_0 (*Subjekte*) := $1*s_1 + 1*s_2 + ... + 1*s_n$

- Die Stelle *Subjekt in Aktion* gibt an, ob bereits ein Subjekt aktiv ist. Diese Menge enthält das Element {nein}, da zu Beginn kein Subjekt aktiv ist.

  m_0 (*Subjekt in Aktion*) := {nein}

Invariante der Authentisierung

Invariante: Die Stelle *Subjekt in Aktion* hat entweder die Marke {nein}oder {ja} und damit ist die Anzahl der Marken immer eins.
|m (*Subjekt in Aktion*)| = 1

Transitionen der Authentisierung

Transition A-1: Subjekt aktivieren

Ein Subjekt darf nur dann in den Prozeß eintreten, wenn kein anderes Subjekt zu dem Zeitpunkt aktiv ist. Dies wird durch die Stelle *Subjekt in Aktion* sichergestellt. Ist noch

kein Subjekt aktiv, ist die Marke der Stelle *Subjekt in Aktion* {nein}. Nur dann schaltet die Transition *Subjekt aktivieren.*

Die Guard-Funktion stellt sicher, daß:

$s \in m$ (*Subjekte*) $\wedge$ {nein} $\in m$ (*Subjekt in Aktion*)

Transition A-2: Subjekt deaktivieren

Hat ein Subjekt den Prozeß vollständig durchlaufen, begibt es sich wieder in die Stelle *Subjekte* zurück, aus der es gestartet ist. Die Stelle *Subjekt in Aktion* hat zu diesem Zeitpunkt die Marke {ja}, nur dann schaltet die Transition *Subjekt deaktivieren.*

Die Guard-Funktion stellt sicher, daß:

$s \in m$ (*Subjekte*) $\wedge$ {ja} $\in m$ (*Subjekt in Aktion*) $\wedge$
m (*Aktives Subjekt*) = {(s, ∅, ∅, ∅, ∅, ∅, ∅)}

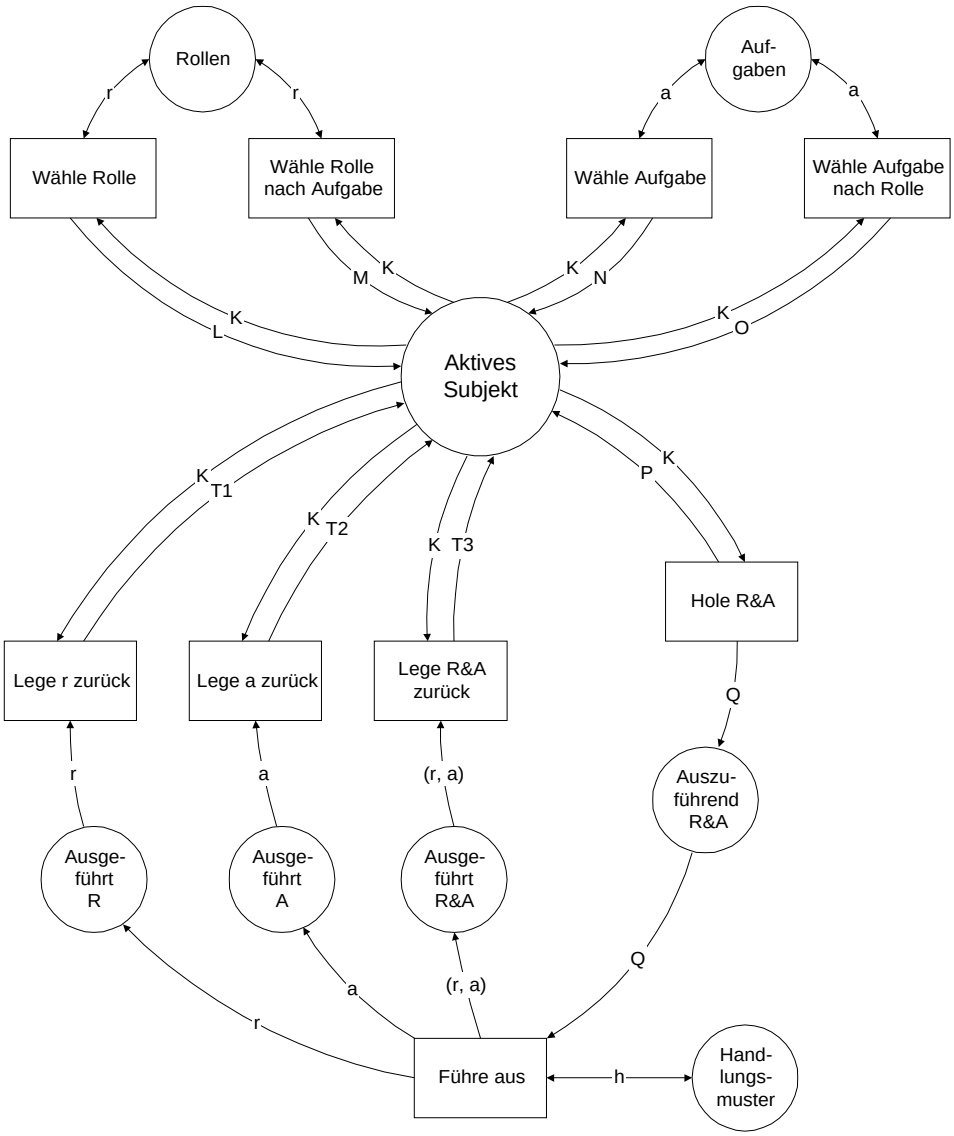
Im folgenden wird nun das Petrinetz nach Beginn der Authentisierung des Subjekts beschrieben (Abbildung 6-6). Es beginnt in seiner Darstellung mit der Stelle *Aktives Subjekt*. Das aktive Subjekt befindet sich in der Stelle *Aktives Subjekt* und hat prinzipiell folgende Möglichkeiten zur Auswahl: Entweder wählt es eine Rolle (*Wähle Rolle, Wähle Rolle nach Aufgabe*) oder es wählt eine Aufgabe (*Wähle Aufgabe, Wähle Aufgabe nach Rolle*) oder es stößt die Ausführung der Handlungsmuster durch Übertragung der Rollen-Aufgaben-Kombinationen an (*Hole R&A*).

Die Stelle *Aktives Subjekt* kann feststellen, ob das Subjekt nur eine Rolle gewählt hat und noch eine Aufgabe benötigt oder ob das Subjekt erst eine Aufgabe gewählt hat und noch eine Rolle benötigt. Je nachdem, was das Subjekt bereits gewählt hat, schaltet die Transition *Wähle Rolle nach Aufgabe* oder *Wähle Aufgabe nach Rolle.*

Weiterhin kann die Stelle *Aktives Subjekt* prüfen, ob das Subjekt nun zur Ausführung von der Transition *Führe aus* berechtigt ist. Dies ist nur der Fall, wenn das Subjekt mindestens eine aktuelle Rollen-Aufgaben-Kombination ausgewählt hat.

Die Stelle Aktives Subjekt enthält neben dem aktiven Subjekt verschiedene Mengen. Die Menge Akt_R beschreibt die aktuellen Rollen des Subjekts. Die Menge Akt_A beschreibt die aktuellen Aufgaben des Subjekts. Die Menge Akt_RA beschreibt die aktuellen Rollen-Aufgaben-Kombinationen des Subjekts. Die Menge Kop_RA enthält eine Kopie von Akt_RA. Weiterhin gibt es die Menge R, die die zuletzt gewählte Rolle enthält, sowie die Menge A, die die zuletzt gewählte Aufgabe des Subjekts enthält.

Wird eine Marke aus der Stelle *Rollen* oder aus der Stelle *Aufgaben* gewählt, wird die Rolle oder Aufgabe der Menge Akt_R oder Akt_A hinzugefügt und anschließend wird die Marke wieder zurück in die Stelle *Rollen* oder in die Stelle *Aufgaben* gelegt. Bei den Mengen Akt_A und Akt_R handelt es sich demnach um Multimengen, in denen die Elemente mehrfach vorkommen können.



Legende

K : (s, Akt_R, Akt_A, Akt_RA, Kop_RA, R, A)

L : (s, Akt_R + {r}, Akt_A, Akt_RA, Kop_RA, R + {r}, A)

M : (s, Akt_R + {r}, Akt_A, Akt_RA + {(r, a)},
Kop_RA + {(r, a)}, R, A - {a})

N : (s, Akt_R, Akt_A + {a}, Akt_RA, Kop_RA, R, A + {a})

O : (s, Akt_R, Akt_A + {a}, Akt_RA + {(r, a)},
Kop_RA + {(r, a)}, R - {r}, A)

P: (s, Akt_R, Akt_A, Akt_RA, Kop_RA - {(r, a)}, R, A)

Q : (s, {(r, a)})

T1 : (s, Akt_R - {r}, Akt_A, Akt_RA, Kop_RA, R, A)

T2 : (s, Akt_R, Akt_A - {a}, Akt_RA, Kop_RA, R, A)

T3 : (s, Akt_R, Akt_A, Akt_RA - {(r, a)}, Kop_RA, R, A)

Abbildung 6-6: R&A-Modell als gefärbtes Petrinetz

Zusätzlich wird die gewählte Rolle oder Aufgabe in die Menge R oder in die Menge A geschrieben. Dort wird immer die zuletzt gewählte Rolle oder Aufgabe gespeichert und entsprechend wird die Wahl einer weiteren Rolle oder Aufgabe zugelassen oder verboten. Zu Beginn sind beide Mengen leer.

Wird zum Beispiel eine Rolle gewählt, enthält die Menge R genau diese Rolle. Eine weitere Rolle kann nur dann gewählt werden, wenn diese Menge leer ist. So wird

sichergestellt, daß nach Wahl einer Rolle nur eine Aufgabe gewählt werden kann. Eine Aufgabe kann dann nur gewählt werden, wenn die Menge A leer ist. Ist dies geschehen und ist diese Rollen-Aufgaben-Kombination eine gültige Kombination, schließt sich also mit keiner anderen aktuellen Kombination aus, wird die Menge R wieder geleert, so daß weitere Rollen und Aufgaben gewählt werden können. Die Menge R und die Menge A sind danach beide leer. Nach Leeren der Menge R wird die aktuelle Rollen-Aufgaben-Kombination der Mengen Akt_RA und Kop_RA hinzugefügt.

Die Mengen Akt_RA und Kop_RA sind keine Multimenge, da eine mehrfache Erledigung derselben Rollen-Aufgaben-Kombination nicht erlaubt ist. Dies wird schon beim Versuch verboten, ein und dieselbe Rollen-Aufgaben-Kombination erneut auszuwählen (Dynamische Trennung von Pflichten). Dies wird durch die zugehörige Guard-Funktionen der Transitionen *Wähle Rolle*, *Wähle Aufgabe nach Rolle*, *Wähle Aufgabe* und *Wähle Rolle nach Aufgabe* sichergestellt. Guard-Funktionen stellen sicher, daß die Vorbedingungen für die einzelnen Transitionen erfüllt sind.

Stellen

Das gefärbte Petrinetz (GPN) hat folgende Stellen (Plätze), die nun im Detail beschrieben werden:

Aktives Subjekt: Diese Stelle beschreibt das Subjekt s, das durch die Transition *Subjekt aktivieren* weitergeschaltet wurde. *Aktives Subjekt* beschreibt weiterhin die Menge der aktuellen Rollen (Akt_R), die Menge der aktuellen Aufgaben (Akt_A), die Menge der aktuellen Rollen-Aufgaben-Kombinationen (Akt_RA), sowie eine Kopie derselben (Kop_RA) und die Mengen der zuletzt gewählten Rolle (R) und der zuletzt gewählten Aufgabe (A). Für die einzelnen Mengen gilt:

$s \in C$ (*Subjekte*),
$Akt\_R \in C_M$ (*Rollen*),
$Akt\_A \in C_M$ (*Aufgaben*),
$Akt\_RA \in (C$ (*Rollen*) $\times C$ (*Aufgaben*)),
$Kop\_RA \in (C$ (*Rollen*) $\times C$ (*Aufgaben*)),
$R \in C$ (*Rollen*),
$A \in C$ (*Aufgaben*)

Diese Mengen sind direkt nach der Transition *Subjekt aktivieren* leer, da das Subjekt noch keine Rolle oder Aufgabe wählen konnte. Damit das Subjekt mit der Transition *Subjekt deaktivieren* wieder in die Stelle *Subjekt* zurückkehren kann, müssen all diese Mengen wiederum leer sein. Damit wird sichergestellt, daß alle Handlungsmuster ausgeführt wurden. Die Farbfunktion lautet

$$C(\textit{Aktives Subjekt}) \in C(\textit{Subjekte}) \times C_M(\textit{Rollen}) \times C_M(\textit{Aufgaben}) \times (C(\textit{Rollen}) \times C(\textit{Aufgaben})) \times (C(\textit{Rollen}) \times C(\textit{Aufgaben})) \times C(\textit{Rollen}) \times C(\textit{Aufgaben}).$$

Rollen: In dieser Stelle stehen alle Rollen, die für Subjekte autorisiert sind und damit ausgewählt werden können. Die Farbfunktion lautet C (*Rollen*) = $\{r_1, ..., r_m\}$.

Aufgaben: In dieser Stelle stehen alle Aufgaben, die für Subjekte autorisiert sind und damit ausgewählt werden können. Die Farbfunktion lautet C (*Aufgaben*) = $\{a_1, ..., a_k\}$.

HMuster: Die Stelle *HMuster* beschreibt alle Handlungsmuster, die prinzipiell ausgeführt werden können. Ein Subjekt kann in Abhängigkeit einer bestimmten Rollen-Aufgaben-Kombination genau ein Handlungsmuster ausführen. Die Farbfunktion lautet C (*HMuster*) = $\{h_1, ..., h_r\}$.

Auszuführend R&A: Diese Stelle beschreibt alle Rollen-Aufgaben-Kombinationen, die von der Transition *Führe aus* ausgeführt werden sollen. Die Transition *Hole R&A* holt sich aus der Stelle *Aktives Subjekt* alle aktuellen Rollen-Aufgaben-Kombinationen aus der Menge Kop_RA und legt sie in der Stelle *Auszuführend R&A* ab. Dort wird auch die Information gespeichert, welches Subjekt die Kombination ausführen möchte. Die Farbfunktion lautet C (*Auszuführend R&A*) = $\{(s, (r, a))_1, ..., (s, (r, a))_x\}$.

Ausgeführt R: Diese Stelle beschreibt die abgearbeiteten Rollen, nachdem die Transition *Führe aus* ausgeführt wurde. Dort werden alle Rollen gesammelt, die in einer Rollen-Aufgaben-Kombination verwendet wurden. Rollen dürfen mehrfach vorkommen. Von dieser Stelle aus werden die abgearbeiteten Rollen durch die Transition *Lege r zurück* aus der Menge der aktuellen Rollen (Akt_R) in der Stelle *Aktives Subjekt* gelöscht. Die Farbfunktion lautet C (*Ausgeführt R*) = $\{r_1, ..., r_y\}$.

Ausgeführt A: Diese Stelle beschreibt die abgearbeiteten Aufgaben, nachdem die Transition *Führe aus* ausgeführt wurde. Dort werden alle Aufgaben gesammelt, die in einer Rollen-Aufgaben-Kombination verwendet wurde. Aufgaben dürfen mehrfach vorkommen. Von dieser Stelle aus werden die abgearbeiteten Aufgaben durch die Transition *Lege a zurück* aus der Menge der aktuellen Aufgaben (Akt_A) in der Stelle *Aktives Subjekt* gelöscht. Die Farbfunktion lautet C (*Ausgeführt A*) = $\{a_1, ..., a_z\}$.

Ausgeführt R&A: Diese Stelle beschreibt die abgearbeiteten Rollen-Aufgaben-Kombinationen, nachdem die Transition *Führe aus* ausgeführt wurden. Dort werden alle Rollen-Aufgaben-Kombinationen gesammelt, deren Handlungsmuster ausgeführt wurden. Von dieser Stelle aus werden die abgearbeiteten Rollen-Aufgaben-Kombinationen durch die Transition *Lege R&A zurück* aus der Menge der aktuellen Rollen-Aufgaben-Kombinationen (Akt_RA) in der Stelle *Aktives Subjekt* gelöscht. Die Farbfunktion lautet C (*Ausgeführt R&A*) = $\{(r, a)_1, ..., (r, a)_w\}$.

Anfangsmarkierungen

Das Petrinetz hat zu Beginn folgende Anfangsmarkierungen:

- Zu Beginn des Ablaufs ist noch kein Subjekt aktiv, deshalb enthält die Stelle *Aktives Subjekt* nur leere Mengen.

  m_0 (*Aktives Subjekt*) := {(∅, ∅, ∅, ∅, ∅, ∅, ∅)}

- Die Elemente der Stelle *Rollen* sind alle Rollen, die im Modell vorhanden sind.

  m_0 (*Rollen*) := $1*r_1 + 1*r_2 + ... + 1*r_m$

- Die Elemente der Stelle *Aufgaben* sind alle Aufgaben, die im Modell vorhanden sind.

  m_0 (*Aufgaben*) := $1*a_1 + 1*a_2 + ... + 1*a_k$

- Die Elemente der Stelle *HMuster* sind alle Handlungsmuster, die für Rollen-Aufgaben-Kombinationen existieren.

  m_0 (*HMuster*) := $1*h_1 + 1*h_2 + ... + 1*h_r$

- Zu Beginn des Ablaufs existiert noch keine aktuelle Rollen-Aufgaben-Kombination, deshalb enthält die Stelle *Auszuführend R&A* die leere Menge.

  m_0 (*Auszuführend R&A*) := ∅

- Zu Beginn des Ablaufs wurde noch kein Handlungsmuster ausgeführt und somit noch keine Rolle abgearbeitet. Die Stelle *Ausgeführt R* enthält deshalb die leere Menge.

  m_0 (*Ausgeführt R*) := ∅

- Zu Beginn des Ablaufs wurde noch kein Handlungsmuster ausgeführt und somit noch keine Aufgabe abgearbeitet. Die Stelle *Ausgeführt A* enthält deshalb die leere Menge.

  m_0 (*Ausgeführt A*) := ∅

- Zu Beginn des Ablaufs wurde noch kein Handlungsmuster ausgeführt und somit noch keine Rollen-Aufgaben-Kombination abgearbeitet. Die Stelle *Ausgeführt R&A* enthält deshalb die leere Menge.

  m_0 (*Ausgeführt R&A*) := ∅

Für eine Analyse werden keine speziellen Analysewerkzeuge verwendet, sondern aufgrund der graphischen Darstellung des Petrinetzes Aussagen getroffen, die aus der Spezifikation ersichtlich sind. Diese Aussagen betreffen mögliche Invarianten des Petrinetzes, Nebenläufigkeiten und die Guard-Funktionen der Transitionen. Eine solche Analyse des Petrinetzes läßt folgende Aussagen zu:

Invarianten

1. Invariante: Da nur ein Subjekt zu einem Zeitpunkt einen Prozeß des Petrinetzes durchlaufen darf, ist die Summe der Anzahl der Marken der Stelle *Aktives Subjekt*, bezüglich des Subjektes immer eins.
   |m (*Aktives Subjekt*)| = 1

Nebenläufigkeit

Ein Vorteil der Darstellung mit Petrinetzen ist die Möglichkeit, Nebenläufigkeiten zu visualisieren. Ein aktives Subjekt kann verschiedene Rollen und Aufgaben zur Benutzung auswählen. Wenn sich diese Rollen-Aufgaben-Kombinationen nicht gegenseitig ausschließen, können mehrere Rollen-Aufgaben-Kombinationen gleichzeitig ausgeführt werden. Dieses gleichzeitige Ausführen wird als Nebenläufigkeit bezeichnet. Jedes einzelne Handlungsmuster läuft als eigenständiger Prozeß unabhängig von den anderen Handlungsmustern ab.

Im Petrinetz wird die Nebenläufigkeit an der gleichzeitigen und voneinander unabhängigen Ausführung der Transition *Führe aus* deutlich. Die Transition *Führe aus* benötigt als Eingabe das Subjekt und die Rollen-Aufgaben-Kombination, die erledigt werden soll. Diese Informationen werden, noch sequentiell, in die Stelle Auszuführende Rollen und Aufgaben (*Auszuführend R&A*) mit der Transition *Hole R&A* aus der Stelle *Aktives Subjekt* geholt. Nachdem alle Rollen-Aufgaben-Kombinationen mit der Information des aktiven Subjekts in dieser Stelle vorhanden sind, kann die Transition *Führe aus* als nebenläufiger Prozeß abgewickelt werden. Jede Transition *Führe aus* holt sich aus der Stelle *HMuster* das entsprechende Handlungsmuster, das ausgeführt werden soll. Anschließend legt die Transition *Führe aus* die abgearbeitete Rolle in die Stelle ausgeführte Rollen (*Ausgeführt R*) und die abgearbeitete Aufgabe in die Stelle der ausgeführten Aufgaben (*Ausgeführt A*) und die abgearbeitete Rollen-Aufgaben-Kombination in die Stelle der ausgeführten Rollen-Aufgaben-Kombinationen (*Ausgeführt R&A*) zurück.

Transitionen

Vergleichbar zu den Überführungsfunktionen des formalen R&A-Modells (siehe Kapitel 6.4) existieren Transitionen mit Guard-Funktionen. Die Überführungsfunk-

tionen werden sinngemäß auf die Transitionen übertragen. Zusätzlich zu den Überführungsfunktionen werden jedoch weitere Transitionen definiert. Um die Nebenläufigkeit graphisch darzustellen, werden vier weitere Transitionen benötigt (*Hole R&A, Lege r zurück, Lege a zurück, Lege R&A zurück*). Die Transition *Hole R&A* legt alle aktuellen Rollen-Aufgaben-Kombinationen aus der Menge Kop_RA, der Kopie der aktuellen Rollen-Aufgaben-Kombinationen mit dem zugehörigen Subjekt in die Stelle *Auszuführend R&A*. Nach *Führe aus* werden alle abgearbeiteten Rollen in die Stelle *Ausgeführt R*, alle abgearbeiteten Aufgaben in die Stelle *Ausgeführt A* und alle abgearbeiteten Rollen-Aufgaben-Kombination in die Stelle *Ausgeführt R&A* gelegt. Von dort werden sie mit den Transitionen *Lege r zurück*, *Lege a zurück* und *Lege R&A zurück* aus den Mengen der aktuellen Rollen, der aktuellen Aufgaben und der aktuellen Rollen-Aufgaben-Kombinationen der Stelle *Aktives Subjekt* gelöscht.

Transition 1: Wähle Rolle

Ein Subjekt wählt zuerst eine Rolle aus, in der es eine Aufgabe erledigen möchte, danach wählt es eine Aufgabe. Die Auswahl einer Rolle ist nur dann erlaubt, wenn diese Rolle für das Subjekt autorisiert ist. Das Subjekt darf nur dann eine weitere aktuelle Rolle wählen, wenn das Subjekt noch keine aktuelle Rolle hat oder wenn sich diese neue Rolle mit keiner der bereits aktuellen Rollen des Subjekts ausschließt. Die Mengen AutRollen und AusschlAktRollen sind von der Spezifikation des formalen R&A-Modells übernommen und beschreiben zum einen die autorisierten Rollen für das Subjekt und zum anderen die Rollen, die sich mit der aktuellen Rolle des Subjekts ausschließen. Ist der erste Teil der Guard-Funktion der Transition erfüllt, wird die gewählte Rolle zu der Menge der aktuellen Rollen (Akt_R) und zu der Menge der zuletzt gewählten Rolle (R) hinzugefügt.

Wurde zuletzt eine Rolle gewählt, schaltet die Transition *Wähle Rolle* nicht.

Die Guard-Funktion stellt sicher, daß:

1. Fall: $R = \emptyset \wedge A = \emptyset$
$s \in m$ (*Subjekte*) $\wedge\ r \in$ AutRollen (s) $\wedge$
[$Akt\_R = \emptyset \vee$
$\forall r' \in$ Akt_R: $r' \notin$ AusschlAktRollen (r)]

Transition 2: Wähle Rolle nach Aufgabe

Die Auswahl einer Rolle, nachdem eine aktuelle Aufgabe gewählt wurde, ist nur dann erlaubt, wenn die entstehende Rollen-Aufgaben-Kombination für das Subjekt autorisiert ist und wenn sich diese Kombination mit keiner bereits aktuellen Rollen-Aufgaben-Kombination des Subjekts ausschließt. Die Menge AutRollenUndAufgaben ist von der Spezifikation formalen R&A-Modells übernommen und beschreibt die Rollen-Aufgaben-Kombinationen, die für das Subjekt autorisiert sind. Die Menge AusschlAktRollenUndAufgaben ist ebenso von der Spezifikation des formalen R&A-Modells übernommen und beschreibt die Rollen-Aufgaben-Kombinationen, die sich mit der aktuellen Rollen-Aufgaben-Kombination des Subjekts ausschließen. Ist der zweite Teil der Guard-Funktion der Transition erfüllt, wird die gewählte Rolle mit der zuletzt gewählten Aufgabe aus (A) zu der Menge der aktuellen Rollen-Aufgaben-

Kombinationen (Akt_RA) hinzugefügt. Die zuletzt gewählte Aufgabe wird aus der Menge der zuletzt gewählten Aufgabe (A) gelöscht, um die Wahl weiterer Rollen-Aufgaben-Kombinationen zu ermöglichen.

Wurde zuletzt eine Rolle gewählt, schaltet die Transition *Wähle Rolle nach Aufgabe* nicht.

Die Guard-Funktion stellt sicher, daß:

2. Fall: $R = \emptyset \wedge A \neq \emptyset$
$s \in m(\textit{Subjekte}) \wedge (r, a) \in AutRollenUndAufgaben(s) \wedge$
$[\, Akt\_RA = \emptyset \vee$
$\forall (r', a') \in Akt\_RA :$
$(r', a') \notin AusschlAktRollenUndAufgaben(r, a)\ ]$

Transition 3: Wähle Aufgabe

Ein Subjekt darf eine Aufgabe zur Erledigung wählen, wenn diese gewünschte Aufgabe zu der Menge von autorisierten Aufgaben für das Subjekt gehört. Das Subjekt darf nur dann eine weitere Aufgabe zur Erledigung auswählen, wenn es zur Zeit keine aktuelle Aufgabe hat oder wenn sich diese neue Aufgabe mit keiner der bereits aktuellen Aufgaben gegenseitig ausschließt. Die Menge AutAufgaben und AusschlAktAufgaben sind aus der Spezifikation des formalen R&A-Modells übernommen und beschreiben zum einen die autorisierten Aufgaben für das Subjekt und zum anderen die Aufgaben, die sich mit der aktuellen Aufgabe ausschließen. Ist der erste Teil der Guard-Funktion der Transition erfüllt, wird die gewählte Aufgabe zu der Menge der aktuellen Aufgaben (Akt_A) und zu der Menge der zuletzt gewählten Aufgabe (A) hinzugefügt.

Wurde zuletzt eine Aufgabe gewählt, schaltet die Transition *Wähle Aufgabe* nicht.

Die Guard-Funktion stellt sicher, daß:

1. Fall: $R = \emptyset \wedge A = \emptyset$
$s \in m(\textit{Subjekte}) \wedge a \in AutAufgaben(s) \wedge$
$[\, Akt\_A = \emptyset \vee$
$\forall a' \in Akt\_A : a' \notin AusschlAktAufgaben(a)\ ]$

Transition 4: Wähle Aufgabe nach Rolle

Wurde bereits eine aktuelle Rolle gewählt, kann nur dann eine Aufgabe dazu gewählt werden, wenn die entstehende Rollen-Aufgaben-Kombination für das Subjekt autorisiert ist und wenn sich diese Kombination mit keiner der bereits aktuellen Rollen-Aufgaben-Kombination des Subjekts ausschließt. Die Menge AutRollenUndAufgaben ist von der Spezifikation des formalen R&A-Modells übernommen und beschreibt die Rollen-Aufgaben-Kombinationen, die für das Subjekt autorisiert sind. Die Menge AusschlAktRollenUndAufgaben ist ebenso von der Spezifikation des formalen R&A-Modells übernommen und beschreibt die Rollen-Aufgaben-Kombinationen, die sich mit der aktuellen Rollen-Aufgaben-Kombination des Subjekts ausschließen. Ist der zweite Teil der Guard-Funktion der Transition erfüllt, wird die gewählte Aufgabe mit der zuletzt gewählten Rolle aus (R) zu der Menge der aktuellen Rollen-Aufgaben-Kombinationen (Akt_RA) hinzugefügt. Die zuletzt gewählte Rolle wird aus der Menge

der zuletzt gewählten Rolle (R) gelöscht, um die Wahl weiterer Rollen-Aufgaben-Kombinationen zu ermöglichen.

Wurde zuletzt eine Aufgabe gewählt, schaltet die Transition *Wähle Aufgabe nach Rolle* nicht.

Die Guard-Funktion stellt sicher, daß:

2. Fall: $R \neq \emptyset \wedge A = \emptyset$
$s \in m$ (*Subjekte*) $\wedge$ (r, a) $\in$ AutRollenUndAufgaben (s) $\wedge$
[Akt_RA = $\emptyset \vee$
$\forall$ (r', a') $\in$ Akt_RA :
(r', a') $\notin$ AusschlAktRollenUndAufgaben (r, a)]

Transition 5: Hole R&A

Diese Transition holt nacheinander alle aktuellen Rollen-Aufgaben-Kombinationen aus der Menge Kop_RA aus der Stelle *Aktives Subjekt* in die Stelle *Auszuführend R&A*. Es wird auch eine Kopie des Subjekts s angefertigt, da jede Transition *Führe aus* neben der Rolle und der Aufgabe auch das Subjekt benötigt. Von der Stelle *Auszuführend R&A* kann dann die Transition *Führe aus* alle Rollen-Aufgaben-Kombinationen gleichzeitig und unabhängig voneinander als nebenläufige Prozesse ausführen.

Die Guard-Funktion stellt sicher, daß:

$s \in m$ (*Subjekte*) $\wedge$ Akt_RA $\neq \emptyset$

Transition 6: Führe aus

Diese Transition kann nebenläufig ausgeführt werden, nachdem alle Rollen-Aufgaben-Kombination aus der Stelle Aktives Subjekt in die Stelle *Auszuführend R&A* geholt wurden. Ein Subjekt kann eine Aufgabe in einer Rolle dann erledigen, wenn die Rollen-Aufgaben-Kombination zu den für das Subjekt aktuellen Rollen-Aufgaben-Kombinationen gehören. Weiterhin muß zu der aktuellen Rollen-Aufgaben-Kombination ein Handlungsmuster existieren.

Dies wird alles in der Funktion ErlaubterZugriff aus dem formalen R&A-Modell geprüft, die hier verwendet wird. Die Transition legt anschließend die Rolle in die Stelle *Ausgeführt R,* die Aufgabe in die Stelle *Ausgeführt A* und die Rollen-Aufgaben-Kombination in die Stelle *Ausgeführt R&A.*

Die Guard-Funktion stellt sicher, daß:

$s \in m$ (*Subjekte*) $\wedge$ ErlaubterZugriff (s, r, a)

Transition 7: Lege r zurück

Die Transition *Lege r zurück* holt aus der Menge der ausgeführten Rollen (*Ausgeführt R*) eine Rolle heraus und löscht sie aus der Menge der aktuellen Rollen (Akt_R) des Subjekts aus der Stelle *Aktives Subjekt*. Solange noch Elemente in der Menge *Ausgeführt R* sind, schaltet die Transition *Lege r zurück*.

Nach Schalten der letzten Transition *Lege r zurück* wurden alle Rollen aus die Menge ausgeführte Rollen (*Ausgeführt R*) herausgenommen und aus der Menge Akt_R der

Stelle *Aktives Subjekt* gelöscht, so daß nach dem Ausführen des letzten Handlungsmusters keine aktuelle Rolle mehr für das Subjekt existiert.

Die Guard-Funktion stellt sicher, daß:

m (*Ausgeführt R*) ≠ ∅

Transition 8: Lege a zurück

Die Transition *Lege a zurück* holt aus der Menge der ausgeführten Aufgaben (*Ausgeführt A*) eine Aufgabe heraus und löscht sie aus der Menge der aktuellen Aufgaben (Akt_A) des Subjekts aus der Stelle *Aktives Subjekt*. Solange noch Elemente in der Menge *Ausgeführt A* sind, schaltet die Transition *Lege a zurück*. Nach Schalten der letzten Transition *Lege a zurück* wurden alle Aufgaben aus die Menge ausgeführte Aufgaben (*Ausgeführt A*) herausgenommen und aus der Menge Akt_A der Stelle *Aktives Subjekt* gelöscht, so daß nach dem Ausführen des letzten Handlungsmusters keine aktuelle Aufgabe mehr für das Subjekt existiert.

Die Guard-Funktion stellt sicher, daß:

m (*Ausgeführt A*) ≠ ∅

Transition 9: Lege R&A zurück

Die Transition *Lege R&A zurück* holt aus der Menge der ausgeführten Rollen-Aufgaben-Kombinationen (*Ausgeführt R&A*) eine Rollen-Aufgaben-Kombination heraus und löscht sie aus der Menge der aktuellen Rollen-Aufgaben-Kombinationen (Akt_RA) des Subjekts aus der Stelle *Aktives Subjekt*. Solange noch Elemente in der Menge *Ausgeführt R&A* sind, schaltet die Transition *Lege R&A zurück*. Nach Schalten der letzten Transition *Lege R&A zurück* wurden alle Rollen-Aufgaben-Kombinationen aus der Menge der ausgeführten Rollen-Aufgaben-Kombinationen (*Ausgeführt R&A*) herausgenommen und aus der Menge Akt_RA der Stelle *Aktives Subjekt* gelöscht, so daß nach dem Ausführen des letzten Handlungsmusters keine aktuelle Aufgabe mehr für das Subjekt existiert.

Die Guard-Funktion stellt sicher, daß:

m (*Ausgeführt R&A*) ≠ ∅

Endlichkeit

Die Überprüfbarkeit der Guard-Funktionen und der Invarianten wird durch die Endlichkeit des R&A-Modells gewährleistet. Durch die begrenzte Anzahl von Rollen und Aufgaben ergibt sich ebenso nur eine begrenzte Zahl von Rollen-Aufgaben-Kombinationen, nämlich bei n Aufgaben und m Rollen maximal n*m Kombinationen von erlaubten Rollen und Aufgaben. Da keine doppelten Kombinationen zugelassen werden, ist dadurch die Endlichkeit des Modells garantiert.

6.6 Bewertung anhand der grundlegenden Sicherheitsanforderungen

Das formale R&A-Modell wird anhand eines Vergleichs zum rollenbasierten Modell bewertet, da das formale R&A-Modell auf dem rollenbasierten Modell (RBAC) aufbaut. Das formale R&A-Modell stellt eine Erweiterung des rollenbasierten Modells um eine zweite Dimension (Aufgaben) dar. Es unterstützt alle Eigenschaften des RBAC und stellt zusätzlich weitere Sicherheitsvorteile für den Benutzer zur Verfügung.

Das formale R&A-Modell unterstützt folgende Eigenschaften:

- Zugriffsrechte nach MAC: Das formale R&A-Modell ist prinzipiell nach MAC konzipiert. Es ermöglicht dem Benutzer jedoch eine eigenständige Konfiguration der Rollen und Aufgaben. Der Benutzer kann bestehende Rollen und Aufgaben in ihren Möglichkeiten begrenzen und damit eine eingeschränkte Benutzung seiner R&A-Chipkarte für andere Benutzer (zum Beispiel den eigenen Kinder) zulassen (eingeschränktes DAC). Der Benutzer kann jedoch keine Rechte an andere weitergeben, über die er selbst nicht verfügt.
- Leichte Administration: Das formale R&A-Modell ermöglicht eine leichte Administration durch einfaches Hinzufügen von neuen Rollen und Aufgaben. Da die Zugriffe in positiver Weise definiert sind, kann durch Definition einer weiteren Rollen-Aufgaben-Kombination leicht ein neuer Zugriff (Handlungsmuster) zu der Menge der bestehenden Handlungsmuster hinzugefügt werden. Es können auch nur Rollen oder nur Aufgaben hinzugefügt werden, die in die entsprechende Liste von autorisierten Rollen oder Aufgaben hinzugenommen werden.
- Pflichtentrennung: Statische und dynamische Trennung von Pflichten ermöglicht klare Zugriffsregelungen und strikte Trennung von Zuständigkeiten. Eine saubere Modellierung von statischer und dynamischer Trennung von Pflichten gewährleistet Vertraulichkeit und Integrität und verhindert unberechtigten Zugriff.
- Gleichzeitige Ausführung von konfliktfreien Anwendungen: Die dynamische Trennung von Pflichten ermöglicht das gleichzeitige Ausführen von konfliktfreien Anwendungen mit unterschiedlichen Sicherheitsanforderungen.
- Minimierung der notwendigen Rechte: Durch das Prinzip der statischen Trennung von Pflichten werden nur die notwendigen Rechte zur Erledigung einer Rollen-Aufgaben-Kombination erteilt.
- Zugriff über wohldefinierte Prozeduren auf Objekte: Das formale R&A-Modell erlaubt den Zugriff auf Datenobjekte nur über wohldefinierte Prozeduren.

Das R&A-Modell stellt zusätzliche Sicherheitseigenschaften für den Benutzer zur Verfügung:

- Zweidimensionale Struktur: Die zweidimensionale Struktur erlaubt eine bessere Abbildung der realen Anwendungen auf das R&A-Modell. Durch Spezifikation von Rollen und Aufgaben können Anwendungen genauer beschrieben und für den Benutzer transparenter abgebildet werden.
- Feine Granularität der Zugriffe: Das formale R&A-Modell ermöglicht durch die Minimierung der Zugriffsrechte und die Zweidimensionalität eine genauere und feinere Detaillierung im Zugriff. So wird sichergestellt, daß kein unberechtigter oder unbefugter Zugriff auf sensitive Datenobjekte erfolgen kann.
- Vertraulichkeit und Integrität: Das formale R&A-Modell ist für die Anwendung auf einer R&A-Chipkarte vorgesehen. Durch Kombination von Chipkarten mit kryptographischen Modulen wird Vertraulichkeit und Integrität gewahrt. Kryptographische Algorithmen unterstützen durch Verschlüsselung und digitale Signaturen die Vertraulichkeit und Integrität.
- Informationelle Selbstbestimmung: Das formale R&A-Modell gibt nicht strikt alle Aktionen des Benutzers vor, sondern es definiert einen Rahmen, in dem sich der Benutzer frei bewegen kann. Dadurch wird die informationelle Selbstbestimmung des Benutzers gewahrt. Er kann selbst entscheiden, welche Informationen er wann über sich freigibt.
- Erhöhung der Akzeptanz: Durch die Möglichkeit der vollständigen Kontrolle, die der Benutzer über das R&A-Modell hat, kann die Akzeptanz eines entsprechenden Systems mit R&A-Chipkarten erhöht werden.

Durch die genannten zusätzlichen Sicherheitseigenschaften erhöht sich die Sicherheit und der Datenschutz für den Benutzer. Die grundlegenden Sicherheitsanforderungen aus Kapitel 2.2 sind erfüllt, soweit sie von einem Sicherheitsmodell erfüllt werden können. Die Akzeptanz einer R&A-Chipkarte kann durch den Einsatz des R&A-Modells wesentlich erhöht werden, da der Benutzer die Kontrolle über die Benutzung behält und selbst entscheiden kann, welche Rollen und Aufgaben er zu welchem Zeitpunkt nutzen möchte.

Einer Bewertung des R&A-Modells ist hinzuzufügen, daß prinzipiell Fehler bei der Abbildung von der Realität auf das Modell geschehen können. Dies ist ein prinzipielles Problem beim Modellieren realer Ereignisse. Der Abbildungsprozeß kann nie hundertprozentig erfolgen, da es kein Modell der realen Welt geben kann. Die Fehler können eingegrenzt werden, in dem das R&A-Modell so früh wie möglich in den gesamten Entstehungsprozeß eines Anwendungssystems eingebunden wird.

6.7 Erweiterung des R&A-Modells

Im Hinblick auf eine einfache Handhabung kann es von Vorteil sein, mit Voreinstellung von Rollen oder Aufgaben zu arbeiten. Das bedeutet, daß der Benutzer zu Beginn einer

Nutzung entscheidet, daß er ausschließlich in einer Rolle agieren möchte oder nur eine Aufgabe (jedoch in unterschiedlichen Rollen) erledigen möchte.

Zum Beispiel kann man sich vorstellen, daß ein Benutzer vor Beginn eines Einkaufsbummels entscheidet, daß er an diesem Tag nur mit der GeldKarte bezahlen möchte. Selbstverständlich möchte er die GeldKarte jedoch wieder aufladen, wenn sie leer ist. Das bedeutet, daß er alle Aktionen an diesem Tag in der Rolle des Geldbörsenbesitzers durchführen möchte. Aus diesem Grund ist es sinnvoll, die Voreinstellung von einer Rolle (oder einer Aufgabe) zu ermöglichen, um den Benutzer eine vereinfachte Auswahl zur Verfügung zu stellen.

Dies erfordert eine Erweiterung des formalen R&A-Modells um die Mengen *ExklusiveRolle* und *ExklusiveAufgabe*.

ExklusiveRolle = $\{r_j\}$, wobei r_j die Rolle angibt, die der Benutzer als exklusive Rolle gewählt hat.

ExklusiveAufgabe = $\{a_m\}$, wobei a_m die Aufgabe angibt, die der Benutzer als exklusive Aufgabe gewählt hat.

Der Benutzer kann eine exklusive Rolle oder eine exklusive Aufgabe wählen. Hat der Benutzer eine exklusive Rolle gewählt, ist die Wahl einer weiteren Rolle nicht erlaubt, entsprechendes gilt für Aufgaben. Bei Wahl einer exklusiven Rolle und einer exklusiven Aufgabe ergibt sich genau eine Rollen-Aufgaben-Kombination. Für diese Kombination gibt es genau ein Handlungsmuster, das der Benutzer ausführen kann.

Nachdem ein Handlungsmuster ausgeführt wurde, bleibt die exklusive Rolle oder Aufgabe solange erhalten, bis der Benutzer eine neue exklusive Rolle oder Aufgabe wählt oder die Exklusivität aktiv zurücksetzt.

Die Überführungsfunktion FührеAus aus der Definition des formalen R&A-Modells (siehe Kapitel 6.4.3) muß entsprechend der Erweiterung verändert werden und sieht anschließend folgendermaßen aus.

FührеAus (s_i, r_j, a_m)

IF ErlaubterZugriff (s_i, r_j, a_m)

THEN

IF $r_j \in$ ExklusiveRolle $\wedge$ $a_m \in$ ExklusiveAufgabe

THEN „Ausführen“

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) \ {(r_j, a_m)}

ELSE IF $r_j \in$ ExklusiveRolle $\wedge$ ExklusiveAufgabe = $\varnothing$

THEN „Ausführen“

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) \ {(r_j, a_m)}

AktAufgaben* (s_i) = AktAufgaben (s_i) \ {a_m}

ELSE IF ExklusiveRolle = ∅ ∧ a_m ∈ ExklusiveAufgabe

THEN „Ausführen“

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) \ {(r_j, a_m)}

AktRollen* (s_i) = AktRollen (s_i) \ {r_j}

ELSE „Ausführen“

AktRollenUndAufgaben* (s_i) =

AktRollenUndAufgaben (s_i) \ {(r_j, a_m)}

AktRollen* (s_i) = AktRollen (s_i) \ {r_j}

AktAufgaben* (s_i) = AktAufgaben (s_i) \ {a_m}

ELSE „Fehlerbehandlung“

Eine zusätzliche Erweiterung des formalen R&A-Modells kann in einer anderen Form der Voreinstellung von Rollen und Aufgaben liegen. Es kann eine sinnvolle Erweiterung darstellen, bestimmte Rollen-Aufgaben-Kombinationen über längere Zeit aktuell zu halten, um sie immer wieder auszuführen. Zwischendurch sollen diese Kombinationen jeoch „abschaltbar“ sein, um danach wieder aktuell zu werden.

Bei der Benutzung einer R&A-Chipkarte ist es denkbar, daß der Benutzer über einen bestimmten Zeitraum, zum Beispiel eine ganzen Tag lang, eine bestimmte Rollen-Aufgaben-Kombination ständig verfügbar haben möchte. Beispielsweise soll das Bezahlen mit der Geldbörse die ganze Zeit als aktuelle Rollen-Aufgaben-Konbination möglich sein. Zwischendurch möchte der Benutzer jedoch eine andere Rollen-Aufgaben-Kombination ausführen, die er nicht mit der bereitsgewählten R&A-Kombination gleichzeitig aktuell haben möchte.

Zum Beispiel möchte der Benutzer zwischendurch Überweisungen tätigen. Für diese Ausführung soll das Bezahlen mit der Geldbörse nicht möglich sein. Nach Abschluß der Überweisungstransaktion kann sich die Bezahlfunktion mit der Geldbörse automatisch wieder als aktuelle Rollen-Aufgaben-Kombination einschalten. Dies erfordert einige Erweiterungen des formalen R&A-Modells, die hier im Detail nicht spezifiziert werden sollen. Die sehr informelle Beschreibung einer möglichen Erweiterung sollte lediglich zukünftige Erweiterungen des R&A-Modells deutlich machen.

Wird eine R&A-Chipkarte zur Implementierung gewählt, kann prinzipiell zwischen zwei Chipkartenvarianten bezüglich der Ein- und Ausgabemöglichkeiten unterschieden werden. Zum einen können herkömmliche Chipkarten verwendet werden, wie sie zur Zeit als GeldKarte oder auch in Form der Telefonkarte im Einsatz sind. Bei diesen Chipkarten ist eine Ein- beziehungsweise Ausgabe nur mit einem zusätzlichen Gerät möglich. Dieses Gerät befindet sich in der Regel nicht im Besitz des Benutzers und kann deshalb nicht als vertrauenswürdig angenommen werden. Deshalb benötigt der Benutzer ein weiteres Ein-/Ausgabegerät im Taschenformat, mit dem er seine Auswahl von Rollen und Aufgaben in vertrauenwürdiger Umgebung treffen kann.

Zum anderen ist jedoch der Einsatz von Chipkarten denkbar, die bereits eine integrierte Ein-/Ausgabemöglichkeit in Form eines kleinen Displays und einer Tastatur besitzt. Diese Variante stellt für den Benutzer die praktischste und sicherste Lösung dar, da zur Kommunikation mit seiner R&A-Chipkarte kein zusätzliches Gerät benötigt und somit immer eine vertrauenwürdige Kommunikation gewährleistet ist.

An beide Varianten sind hohe, wenn auch unterschiedliche Anforderungen an die Handhabung und Benutzungsoberfläche zu stellen. In beiden Fällen muß durch das begrenzte Display die Anzeige sehr knapp aber trotzdem aussagekräftig sein. Ebenso muß die Gestaltung der Tastatur oder sonstigen Eingabefeldern nach ergonomischen Gesichtspunkten erfolgen. Es darf für den Benutzer keine wesentliche Schwierigkeit darstellen, eine R&A-Chipkarte zu benutzen.

Zur Veranschaulichung des R&A-Modells wird im nachfolgenden Kapitel 7 die Anwendung einer R&A-Chipkarte für ein komplettes Anwendungsbeispiel (R&A-Anwendung) modelliert und exemplarisch werden verschiedene Abläufe durchgespielt. Es erfolgt eine detaillierte Beschreibung der einzelnen Variablen, Regeln und Überführungsfunktionen des formalen R&A-Modells. Die Darstellung erfolgt als Zustandsdiagramm. In dem Beispiel wird in exemplarischen Zuständen die Korrektheit der aufgestellten Regeln und Überführungsfunktionen gezeigt.

Neben diesen möglichen Erweiterungen des R&A-Modells, sind noch weitere Problembereiche zu formulieren, bevor eine konkrete Anwendung mit dem R&A-Modell realisiert werden kann. Diese werden im einzelnen im abschließenden Kapitel dieser Arbeit (siehe Kapitel 8.2) angesprochen.

7 Modellierung von R&A-Anwendungen

Im vorigen Kapitel wurde das R&A-Modell formal als Zustandsautomat beschrieben. Dafür wurde es graphisch als Zustandsdiagramm dargestellt. Eine zusätzliche Form der graphischen Darstellung des R&A-Modells erfolgte mit gefärbten Petrinetzen, mit denen komplexe Zusammenhänge beschrieben werden können. Um nun die Anwendbarkeit des R&A-Modells zu verdeutlichen, wird es in diesem Kapitel auf eine Anwendung im elektronischen Zahlungsverkehr abgebildet.

Auf der Basis des formalen R&A-Modells werden R&A-Anwendungen für eine multifunktionale Chipkarte (R&A-Chipkarte) im elektronischen Zahlungsverkehr modelliert. Zuerst wird der statische Aspekt des R&A-Modells präsentiert (siehe Kapitel 7.1), anschließend der dynamische Aspekt (siehe Kapitel 7.2). Der statische Aspekt definiert im einzelnen die möglichen Rollen und Aufgaben, die ein Benutzer auswählen und erledigen kann. Weiterhin werden die Subjekte des Systems und die Datenobjekte definiert, die für die einzelnen Rollen und Aufgaben notwendig sind, und auf die mittels Prozeduren zugegriffen werden kann. Es wird festgelegt, welche Rollen und Aufgaben für einen Benutzer autorisiert sein können (Regel der statischen Trennung von Pflichten).

Der dynamische Aspekt des R&A-Modells beschreibt eine mögliche Nutzung der R&A-Chipkarte und wird anhand von zwei Benutzungszenarien im elektronischen Zahlungsverkehr verdeutlicht. Hier werden die Funktionen eingebunden, die eine Benutzung der einzelnen Zustandsvariablen beschreiben. Beispielsweise werden die aktuellen Rollen oder aktuellen Aufgaben eines Subjektes beschrieben, die zum Zeitpunkt der Benutzung gewählt wurden. Ebenso werden die aktuellen Rollen-Aufgaben-Kombinationen beschrieben. Weiterhin werden die Rollen und Aufgaben beschrieben, die sich bei einer gleichzeitigen Ausführung gegenseitig ausschließen (Regel der dynamischen Trennung von Pflichten). Anhand eines gewählten Beispiels wird mit Hilfe der Überführungsfunktionen das R&A-Modell von einem Zustand in den nächsten überführt. Dabei wird überprüft, ob die Regeln (siehe Kapitel 6.4.2) für die einzelnen Zustände erfüllt sind.

R&A-Anwendungen sind entsprechend dem R&A-Modell konzipiert, das heißt, dem Benutzer stehen Rollen und Aufgaben zur Verfügung. Er kann wahlweise zuerst eine Rolle und dann eine Aufgabe wählen. Alternativ kann er in umgekehrter Reihenfolge erst eine Aufgabe wählen und dann die Rolle, in der die Aufgabe erledigt werden soll. Für die Modellierung der Beispielsanwendung macht es jedoch keinen Unterschied, wenn in umgekehrter Reihenfolge vorgegangen wird.

In dem hier vorgestellten Anwendungsbeispiel vereinigt die R&A-Chipkarte folgende schon bestehende Karten in sich. Die elektronische Geldbörse, die EC- beziehungsweise Kundenkarte und die Kreditkarte. Damit integriert sie unterschiedliche Zahlungsfunktionen einzelner Chip- beziehungsweise Magnetstreifenkarten. Ferner unterstützt die R&A-Chipkarte den Zugang zu einzelnen Bankgeschäften (Homebanking) sowie deren Durchführung.

Die R&A-Anwendung definiert Rollen und Aufgaben. Nicht jede Aufgabe kann in jeder Rolle erledigt werden. Die erlaubten Rollen-Aufgaben-Kombinationen für die einzelnen Benutzer werden im folgenden erläutert (siehe Kapitel 7.1.1).

In diesem Anwendungsbeispiel kann die Chipkarte an unterschiedlichen Orten eingesetzt werden. Diese können sein: Ein SB-Terminal, der Kassenschalter einer Bank, der eigene Computer mit einer Netzverbindung zum Beispiel zur eigenen Bank, die eigene Chipkarte (sofern es sich um eine Chipkarte mit integriertem Anzeige- und Tastaturfeld handelt), ein persönliches Lese-/Schreibgerät oder bankenunabhängige Zahlungsterminals, beispielsweise in Geschäften. Die möglichen Orte sind abhängig von der gerade gewählten Rolle und Aufgabe. Bei Kontozugriffen in der Rolle des EC-Besitzers muß immer eine Verbindung zu dem ausgewählten Konto vorhanden sein, während bei Aktionen, die nur die Daten der Chipkarte betreffen, eine solche Verbindung nicht notwendig ist. Demzufolge kann das Anschauen des Guthabenstandes der Geldbörse mit der Chipkarte selbst oder dem persönlichen Lesegerät erfolgen, jedoch muß eine Aufladung der Geldbörse vom Konto am SB-Terminal oder über eine Netzverbindung zur Bank erfolgen.

Für zukünftige Anwendungen sind weitere zahlreiche Rollen und Aufgaben denkbar. Um die Anwendbarkeit des R&A-Modells zu demonstrieren, wird jedoch ein übersichtliches Anwendungsbeispiel gewählt, das jederzeit erweitert werden kann. Im Anwendungsbeispiel gibt es zwei Subjekte, zum einen den Karteninhaber und zum anderen den Vertreter der kartenausgebenden Stelle, im folgenden kurz Bank genannt.

Für das nachfolgende Beispiel sowie für die generelle Benutzung werden einige funktionale Annahmen getroffen. Beispielsweise wird vorausgesetzt, daß das R&A-Modell alle wesentlichen Elemente enthält, die zur Beschreibung eines konkreten Anwendungsfalles notwendig sind. Weiterhin wird von einem fehlertoleranten Gesamtsystem ausgegangen. Das bedeutet, daß Fehler abgefangen werden, die durch das Gesamtsystem entstehen können. Dazu gehören beispielsweise Fehler durch den Ausfall des Hintergrundrechners einer R&A-Anwendung. Diese Art von Fehlern und Problemen wird in diesem Beispiel vernachlässigt und ist nicht relevant für die exemplarische Darstellung des formalen R&A-Modells.

Einen Überblick über das Anwendungsbeispiel liefert Kapitel 7.1. In den darauffolgenden Unterkapiteln wird der statische Zustand des R&A-Modells, also die Variablen des R&A-Modells (Subjekte, Objekte, Rollen und Aufgaben, Prozeduren und Zugriffe) und die Funktionen, die auf diese Variablen zugreifen, beschrieben.

Der dynamische Aspekt des R&A-Modells, also eine beipielhafte Benutzung wird in Kapitel 7.2 beschrieben. Dort wird gezeigt, daß die Überführungsfunktionen bei jedem Zustandsübergang sicherstellen, daß die Regeln erfüllt sind.

7.1 Statischer Aspekt des R&A-Modells

Wie schon in Kapitel 6 kurz erwähnt, wird eine R&A-Anwendung modelliert, die verschiedene Rollen und Aufgaben im Anwendungsbereich des elektronischen Zahlungsverkehrs zur Verfügung stellt (Abbildung 7-2 und Abbildung 7-3).

Bei den nachfolgenden Grafiken werden Symbole nach Abbildung 7-1 verwendet. Die Variablen Subjekte, Rollen und Aufgaben des formalen R&A-Modells werden im Text *kursiv* dargestellt. Die Darstellung in Abbildungen oder Tabellen erfolgt jedoch nicht kursiv. Die Abkürzung für Rollen und Aufgaben erfolgt wie in Kapitel 6.4, dabei bedeutet r_1 Rolle 1 und a_2 Aufgabe 2.

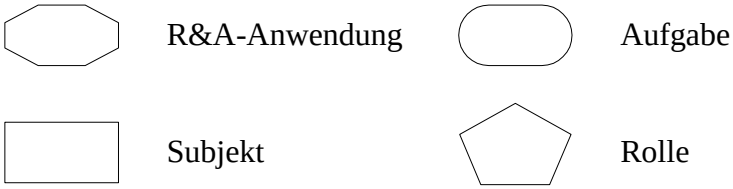


Abbildung 7-1: Verwendete Symbole

7.1.1 Subjekte, Rollen und Aufgaben

Es gibt zwei Subjekte: Den *Karteninhaber* und den Vertreter der kartenausgebenden Stelle, im folgenden kurz *Bank* genannt.

Subjekte	**Beschreibung**
Karteninhaber	Inhaber beziehungsweise der Benutzer der Chipkarte.
Bank	Vertreter der kartenausgebenden Stelle (Bankangestellter).

Tabelle 7-1: Subjekte der R&A-Anwendung

Die R&A-Chipkarte stellt für den Benutzer die Rollen als Geldbörsenbesitzer (*GB-Besitzer*), als EC-Kartenbesitzer (*EC-Besitzer*) und als Kreditkartenbesitzer (*KK-Besitzer*) zur Verfügung. In diesen Rollen können in diesem Beispiel unter anderem die Aufgaben Bezahlen, Geld akzeptieren, Konto führen und Transaktionslimit setzen erledigt werden. Abbildung 7-2 beschreibt die Rollen in diesem Beispiel für die beiden existierenden Subjekte:

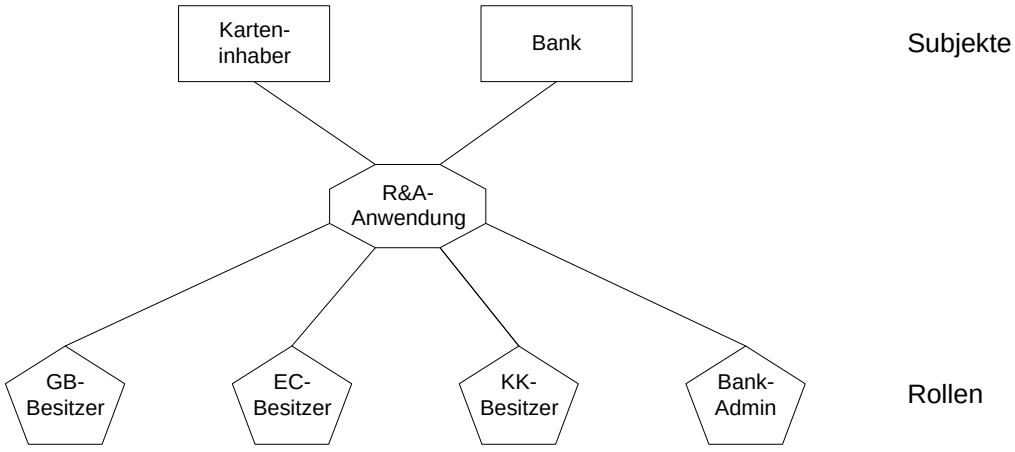


Abbildung 7-2: Subjekte und Rollen der R&A-Anwendung

Die Subjekte können in den Rollen unterschiedliche Aufgaben erledigen. Es hängt jedoch von der jeweiligen Anwendung ab, ob jede Aufgabe in jeder Rolle erledigt werden kann. Aus diesem Grund gibt es in jeder Anwendung zulässige Rollen-Aufgaben-Kombinationen, die im folgenden erläutert werden. Nachfolgende Aufgabenbereiche (Abbildung 7-3) existieren in diesem Anwendungsbeispiel für den *Karteninhaber* und die *Bank*.

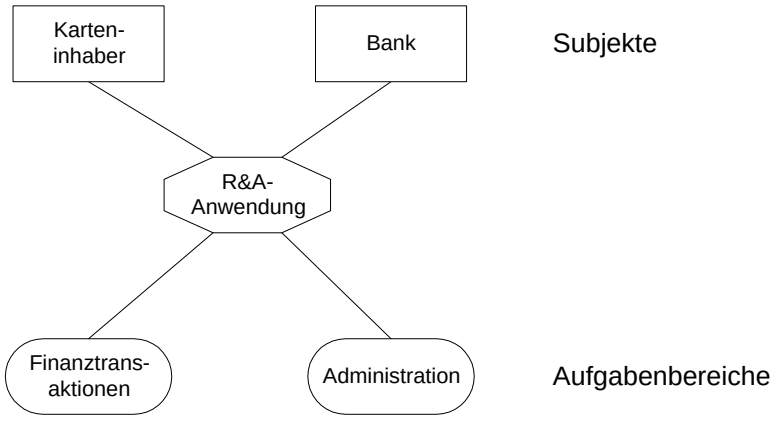


Abbildung 7-3: Subjekte und Aufgabenbereiche der R&A-Anwendung

Um die Übersichtlichkeit der einzelnen Abbildungen zu garantieren, werden einzelne Aufgaben zu inhaltlichen Aufgabenbereichen zusammengefaßt. Ein Aufgabenbereich ist eine graphische Zusammenfassung einzelner inhaltlich zusammenpassender Aufgaben. Diese Zusammenfassung von Aufgaben zu Aufgabenbereichen hat jedoch nur Auswirkungen auf die graphische Darstellung, nicht jedoch auf das R&A-Modell. Im folgenden werden die einzelnen Aufgaben der Aufgabenbereiche mit den zugelassenen Rollen erläutert.

Aufgabenbereich „Finanztransaktionen“:

Der Aufgabenbereich Finanztransaktionen besteht aus den Aufgaben *Bezahlen, Geld akzeptieren, Geld transferieren, Konto Info* und *Konto führen* (Abbildung 7-4). Die Aufgaben *Bezahlen, Geld transferieren und Konto Info* können in den drei Rollen Geldbörsenbesitzer (*GB-Besitzer*), EC-Kartenbesitzer (*EC-Besitzer*) und Kreditkartenbesitzer (*KK-Besitzer*) erledigt werden. Die Aufgabe *Geld akzeptieren* kann in der Rolle *GB-Besitzer* und die Aufgabe *Konto führen* in der Rolle *EC-Besitzer* erledigt werden.

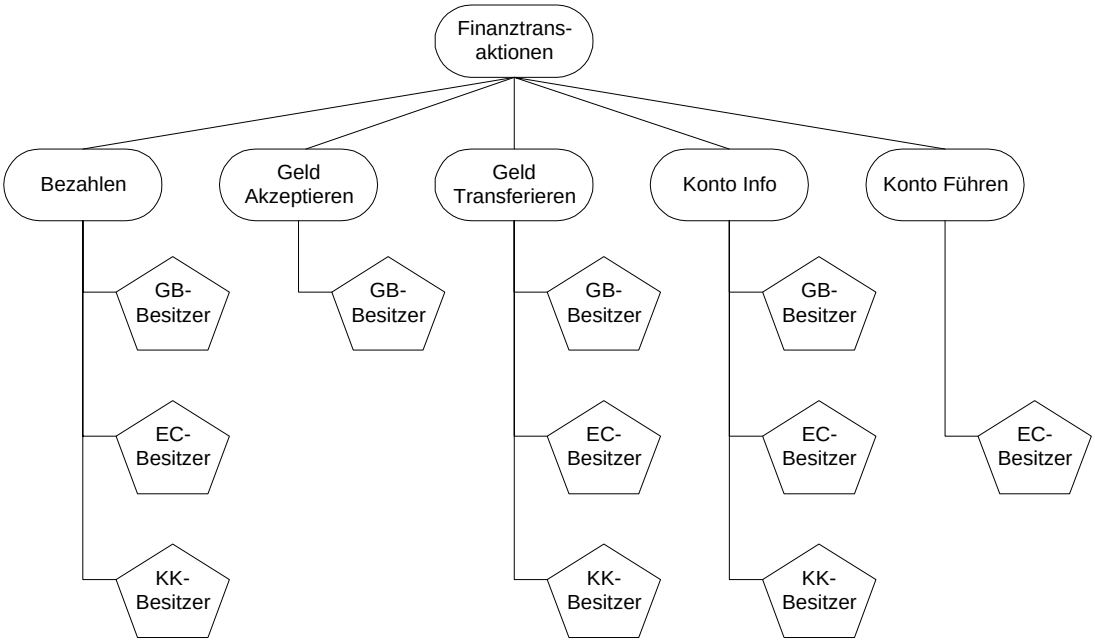


Abbildung 7-4: Aufgabenbereich „Finanztransaktionen“ mit Aufgaben und Rollen

Rollen und Aufgaben des Aufgabenbereichs Finanztransaktionen

a_1 *Bezahlen*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann mit der elektronischen Geldbörse bezahlen.

r_2 *EC-Besitzer*: Der *Karteninhaber* kann mit der EC-Karte einer Bank bezahlen.

r_3 *KK-Besitzer*: Der *Karteninhaber* kann mit der Kreditkarte bezahlen. Es werden die Kreditkartendaten gelesen und ein Beleg erstellt. Der *Karteninhaber* autorisiert diese Zahlung (bisher) mit seiner Unterschrift unter diesen Beleg.

a_2 *Geld Akzeptieren*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann elektronisches Geld von einer anderen elektronischen Geldbörse akzeptieren.

a_3 *Geld Transferieren*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann seine Geldbörse aufladen. Dabei wird Geld von einem ausgewählten Konto auf die elektronische Geldbörse geladen. Die Geldbörse kann auch mit Bargeld geladen werden, wenn die Ladestation Bargeld akzeptiert. Der *Karteninhaber* kann die Geldbörse teilweise oder ganz entladen. Dabei wird Geld von der elektronischen Geldbörse auf ein ausgewähltes Konto transferiert. An geeigneten Automaten kann auch Bargeld ausgezahlt werden. Dabei wird vorausgesetzt, daß Bargeld weiterhin als Zahlungsmittel existieren wird.

r_2 *EC-Besitzer*: Der *Karteninhaber* kann mit der EC-Karte einer Bank Überweisungen auf eigene oder fremde Konten durchführen. An geeigneten Automaten kann auch Bargeld ausgezahlt oder auf ein Konto eingezahlt werden.

r_3 *KK-Besitzer*: Der *Karteninhaber* kann mit der Kreditkarte Überweisungen auf eigene oder fremde Konten durchführen. An entsprechenden Automaten kann auch Bargeld ausgezahlt oder auf ein Konto eingezahlt werden.

a_4 *Konto Info*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann sich den Guthabenstand seiner elektronischen Geldbörse sowie die letzten Transaktionen und Aufbuchungen anschauen.

r_2 *EC-Besitzer*: Der *Karteninhaber* kann sich den Kontostand seines Girokontos anschauen und Informationen über seinen Dispositionskredit, die letzten Ein- und Ausgänge, Zinskonditionen und ähnliche Informationen bezüglich seines Kontos bekommen.

r_3 *KK-Besitzer*: Der *Karteninhaber* kann Informationen über den aktuellen Stand seines Kreditkartenkontos, sowie über Zinskonditionen und Ratenzahlungen erhalten.

a_5 *Konto Führen*

r_2 *EC-Besitzer*: Der Karteninhaber kann mit der EC-Karte einer Bank die Kontoführungsfunktionen nutzen, die die Bank anbietet. Diese Funktionen sind zum Beispiel Schecks bestellen, Daueraufträge einrichten, ändern oder löschen.

Aufgabenbereich „Administration“:

In diesem Aufgabenbereich kann die Chipkarte administriert werden. In verschiedenen Rollen können die Bereiche Geldbörse, EC-Karte und Kreditkarte getrennt voneinander administriert werden. Die Rolle des Bankadministrators ermöglicht das Einrichten von neuen Konten und ein Neukonfigurieren der Chipkarte durch Hinzufügen und Authentisieren von neuen Aufgaben oder durch Löschen von nicht mehr benötigten Aufgaben.

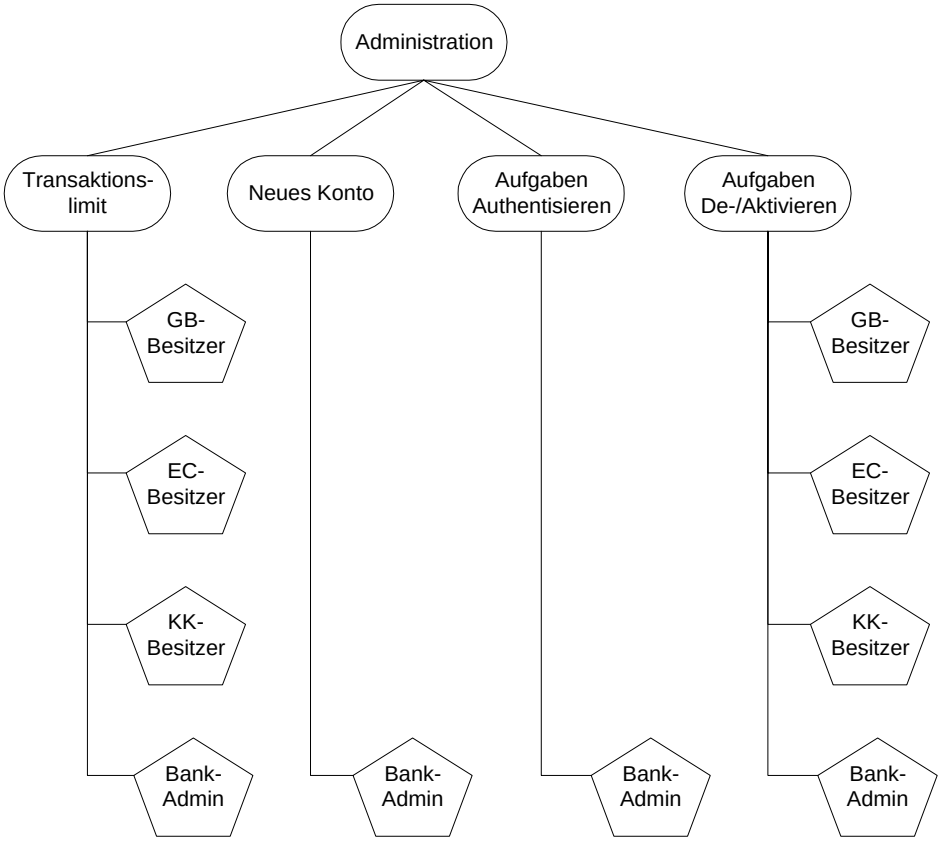


Abbildung 7-5: Aufgabenbereich „Administrieren“ mit Aufgaben und Rollen

Rollen und Aufgaben des Aufgabenbereichs Administration

a_6 *Transaktionslimit*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann für seine Geldbörsentransaktionen ein Limit einrichten, damit gewisse Transaktionsbeträge nicht überschritten werden. Im Rahmen der von der *Bank* vorgegebenen Begrenzungen kann sowohl der Transaktionsbetrag als auch die Anzahl der Transaktionen neu festgelegt werden.

r_2 *EC-Besitzer:* Alle Transaktionen, die über die EC-Karte abgewickelt werden und damit das Girokonto betreffen, können mit einem Limit versehen werden. Im Rahmen der von der *Bank* vorgegebenen Begrenzungen kann sowohl der Transaktionsbetrag als auch die Anzahl der Transaktionen neu festgelegt werden.

r_3 *KK-Besitzer*: Alle Transaktionen, die über die Kreditkarte abgewickelt werden, können mit einem Limit versehen werden. Im Rahmen der von der *Bank* vorgegebenen Begrenzungen kann sowohl der Transaktionsbetrag als auch die Anzahl der Transaktionen neu festgelegt werden.

r_4 *Bank-Admin*: Die *Bank* kann für die zu verwaltenden Konten den Dispositionskredit einrichten oder verändern. Sie legt den maximalen Verfügungsrahmen für die Konten fest. Sie kann das maximale Guthabenlimit für die elektronische Geldbörse festlegen.

a_7 *Neues Konto*

r_4 *Bank-Admin*: Ein neues Konto kann nur von der *Bank* eingerichtet werden. Der *Karteninhaber* kann diese Aufgabe in keiner Rolle erledigen.

a_8 *R&A Authentisieren*

r_4 *Bank-Admin*: Das Einrichten neuer Rollen und Aufgaben ist ebenfalls nur von der *Bank* möglich, da gewährleistet werden muß, daß keine maliziöse Aufgabe versehentlich oder absichtlich geladen wird. Diese Aministrationsaufgabe beinhaltet ebenfalls das Entfernen von Rollen und Aufgaben.

a_9 *R&A De-/Aktivieren*

r_1 *GB-Besitzer*: Der *Karteninhaber* kann Rollen und Aufgaben, die er in der Rolle als Geldbörsenbesitzer nicht mehr erledigen möchte, auf der Chipkarte deaktivieren. Sie stehen ihm dann nicht mehr zur Verfügung. Er kann sie auf Wunsch wieder aktivieren.

r_2 *EC-Besitzer*: Der *Karteninhaber* kann Rollen und Aufgaben, die er in der Rolle als EC-Kartenbesitzer nicht mehr erledigen möchte, auf der Chipkarte deaktivieren. Sie stehen ihm dann nicht mehr zur Verfügung. Er kann sie auf Wunsch wieder aktivieren.

r_3 *KK-Besitzer*: Der *Karteninhaber* kann Rollen und Aufgaben, die er in der Rolle als Kreditkartenbesitzer nicht mehr erledigen möchte, auf der Chipkarte deaktivieren. Sie stehen ihm dann nicht mehr zur Verfügung. Er kann sie auf Wunsch wieder aktivieren.

r_4 *Bank-Admin*: Die *Bank* kann alle Aufgaben deaktivieren, dessen Konten sie verwaltet. Sie kann sie auf Wunsch wieder aktivieren.

7.1.2 Funktionen

Um die einzelnen Variablen des Zustandsautomaten ausreichend beschreiben zu können, wurden in Kapitel 6 entsprechende Funktionen entworfen und ausführlich beschrieben. Die Funktionen des R&A-Modells werden nun anhand der R&A-Anwendung erläutert.

Wie bereits in Kapitel 7.1 beschrieben, besteht die Menge der Subjekte aus zwei Elementen:

$$\text{Subjekte} = \{s_1, s_2\}$$
$$= \{ \textit{Karteninhaber}, \textit{Bank} \}$$

Definierte Rollen der R&A-Chipkarte in diesem Anwendungsbeispiel sind:

$$\text{Rollen} = \{r_1, r_2, r_3, r_4\}$$
$$= \{\text{GB-Besitzer, EC-Besitzer, KK-Besitzer, Bank-Admin}\}$$

Die Aufgaben, die zur R&A-Chipkarte dieses Anwendungsbeispiels gehören, sind:

$$\text{Aufgaben} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9\}$$
$$= \{\textit{Bezahlen}, \textit{Geld Akzeptieren}, \textit{Geld Transferieren},$$
$$\textit{Konto Info}, \textit{Konto Führen}, \textit{Transaktionslimit}, \textit{Neues Konto},$$
$$\textit{R\&A authentisieren}, \textit{R\&A De-/Aktivieren}\}$$

Für das Subjekt *Karteninhaber* existiert folgende Matrix der erlaubten Rollen-Aufgaben-Kombinationen:

(s_1) Inhaber Rollen Aufgaben	**(r_1) GB-Besitzer**	**(r_2) EC-Besitzer**	**(r_3) KK-Besitzer**	**(r_4) Bank-Admin**
(a_1) Bezahlen	✓	✓	✓	
(a_2) Geld Akzeptieren	✓			
(a_3) Geld Transferieren	✓	✓	✓	
(a_4) Konto Info	✓	✓	✓	
(a_5) Konto Führen		✓	✓	
(a_6) Transaktionslimit	✓	✓	✓	
(a_7) Neues Konto				
(a_8) R&A Authent.				
(a_9) R&A De-/Aktiv.	✓	✓	✓	

Tabelle 7-2: Autorisierte Rollen-Aufgaben-Kombinationen für das Subjekt Karteninhaber

Die autorisierten Rollen-Aufgaben-Kombinationen jedes Subjekts werden jeweils durch eine Matrix beschrieben. Ein Eintrag in der Matrix (Tabelle 7-2) bedeutet, daß das

Subjekt *Karteninhaber* die Aufgabe der entsprechenden Zeile in der Rolle der entsprechenden Spalte erledigen darf.

Für das Subjekt *Bank* existiert ebenfalls eine Matrix, in der die erlaubten (autorisierten) Rollen-Aufgaben-Kombinationen definiert sind (Tabelle 7-3).

(s_2) Bank Rollen Aufgaben	**(r_1) GB-Besitzer**	**(r_2) EC-Besitzer**	**(r_3) KK-Besitzer**	**(r_4) Bank-Admin**
(a_1) Bezahlen				
(a_2) Geld Akzeptieren				
(a_3) Geld Transferieren				
(a_4) Konto Info				
(a_5) Konto Führen				
(a_6) Transaktionslimit				✓
(a_7) Neues Konto				✓
(a_8) R&A Authent.				✓
(a_9) R&A De-/Aktiv.				✓

Tabelle 7-3: Autorisierte Rollen-Aufgaben-Kombinationen für das Subjekt Bank

Autorisierte Rollen:

Das Subjekt *Karteninhaber* kann die Rollen *GB-Besitzer*, *EC-Besitzer* und *KK-Besitzer* nutzen, deshalb sind diese drei Rollen für den Karteninhaber autorisiert. Die Menge der autorisierten Rollen ist eine Teilmenge aller Rollen, die in Kapitel 7.1.1 beschrieben sind.

$$\text{AutRollen}\ (s_1) = \{r_1, r_2, r_3\}$$

Die Rolle *Bank-Admin* ist für das Subjekt *Bank* autorisiert. Dies ist eine Teilmenge aller Rollen, die in Kapitel 7.1.1 beschrieben sind.

$$\text{AutRollen}\ (s_2) = \{r_4\}$$

Autorisierte Aufgaben:

Das Subjekt *Karteninhaber* kann die Aufgaben *Bezahlen*, *Geld Akzeptieren*, *Geld Transferieren*, *Konto Info*, *Konto Führen*, *Transaktionslimit* und *R&A De- /Aktivieren* erledigen.

$$\text{AutAufgaben}\ (s_1) = \{a, a_2, a_3, a_4, a_5, a_6, a_9\}$$

Für das Subjekt *Bank* sind die Aufgaben *Transaktionslimit, Neues Konto, R&A Authentisieren* und *R&A De- /Aktivieren* autorisiert.

$$\text{AutAufgaben}\ (s_2) = \{a_6, a_7, a_8, a_9\}$$

Autorisierte Rollen-Aufgaben-Kombinationen:

Wie aus der Tabelle 7-2 ersichtlich, hat das Subjekt *Karteninhaber* folgende autorisierte Rollen-Aufgaben-Kombinationen:

$$\begin{aligned}\text{AutRollenUndAufgaben}\ (s_1) = \{&(r_1, a_1), (r_1, a_2),(r_1, a_3), (r_1, a_4), (r_1, a_6), (r_1, a_9),\\ &(r_2, a_1), (r_2, a_3), (r_2, a_4), (r_2, a_5), (r_2, a_6), (r_2, a_9),\\ &(r_3, a_1), (r_3, a_3), (r_3, a_4), (r_3, a_5), (r_3, a_6), (r_3, a_9)\}\end{aligned}$$

Wie aus der Tabelle 7-3 ersichtlich, hat das Subjekt *Bank* folgende autorisierte Rollen-Aufgaben-Kombinationen:

$$\text{AutRollenUndAufgaben}\ (s_2) = \{(r_4, a_6), (r_4, a_7), (r_4, a_8), (r_4, a_9)\}$$

Statische und Dynamische Trennung von Pflichten:

Die statische Trennung von Pflichten (Static Separation of Duty) bezieht sich sowohl auf Rollen als auch auf Aufgaben und auf Rollen-Aufgaben-Kombinationen. Zur Erläuterung wird für jede Funktion ein Beispiel herangezogen.

Die Rolle *Bank-Admin* schließt sich im Anwendungsbeispiel mit den Rollen *GB-Besitzer, EC-Besitzer, KK-Besitzer* vom Prinzip her aus.

$$\text{AusschlRollen}\ (r_4) = \{r_1, r_2, r_3\}$$

Die Aufgabe *Bezahlen* schließt sich prinzipiell mit den Aufgaben *Neues Konto* und *Aufgaben aktivieren* aus.

$$\text{AusschlAufgaben}\ (a_1) = \{a_8, a_9\}$$

Die statische Trennung von Pflichten für Rollen-Aufgaben-Kombinationen wird an einem Beispiel aus der R&A-Anwendung erläutert. Die Rollen-Aufgaben-Kombination (*GB-Besitzer, Bezahlen*) schließt sich mit den Rollen-Aufgaben-Kombinationen, die nicht autorisierte Kombinationen für das Subjekt *Karteninhaber* sind. Weiterhin schließt sich diese Kombination mit allen Rollen-Aufgaben-Kombinationen des Subjekts *Bank* aus.

AusschlRollenUndAufgaben (r_1, a_1) = {(r_1, a_5), (r_1, a_7), (r_1, a_8), (r_2, a_2),
(r_2, a_7), (r_2, a_8), (r_3, a_2), (r_3, a_7), (r_3, a_8), (r_4, a_1), (r_4, a_2),
(r_4, a_3), (r_4, a_4), (r_4, a_5), (r_4, a_6), (r_4, a_7), (r_4, a_8), (r_4, a_9)}

Die dynamische Trennung von Pflichten (Dynamic Separation of Duty), also der gegenseitige Ausschluß von aktuellen Rollen, aktuellen Aufgaben und aktuellen Rollen-Aufgaben-Kombinationen muß ebenfalls bei der Konfiguration des Systems festgelegt werden. Die Rollen-Aufgaben-Kombination (*GB-Besitzer, Geld Transferieren*) schließt sich mit allen nicht autorisierten Rollen-Aufgaben-Kombinationen aller Subjekte aus. Weiterhin schließt sie sich in diesem Bespiel mit allen autorisierten Rollen-Aufgaben-Kombinationen des Subjekts *Bank* aus. Zusätzlich schließt sie sich mit folgenden Rollen-Aufgaben-Kombinationen (*GB-Besitzer, Bezahlen*), (*GB-Besitzer, Geld akzeptieren*), (*GB-Besitzer, Geld Transferieren*), (*GB-Besitzer, Konto Info*), (*GB-Besitzer, Transaktionslimit*), (*GB-Besitzer, R&A De-/Aktivieren*) (*EC-Besitzer, Geld Transferieren*) und (*KK-Besitzer, Geld Transferieren*) aus.

Das bedeutet, daß die Rollen-Aufgaben-Kombination (*GB-Besitzer, Geld Transferieren*) nicht gleichzeitig mit den angegebenen Kombinationen ausgeführt werden darf.

AusschlAktRollenUndAufgaben (r_1, a_3) = {(r_1, a_5), (r_1, a_7), (r_1, a_8),
(r_2, a_2), (r_2, a_7), (r_2, a_8), (r_3, a_2), (r_3, a_7),
(r_3, a_8), (r_4, a_1), (r_4, a_2), (r_4, a_3), (r_4, a_4), (r_4, a_5),
(r_4, a_6), (r_4, a_7), (r_4, a_8), (r_4, a_9),
(r_1, a_1), (r_1, a_2), (r_1, a_3),(r_1, a_4),
(r_1, a_6), (r_1, a_9), (r_2, a_3), (r_3, a_3)}

Zugriffe:

Für jede einzelne Rollen-Aufgaben-Kombination wird festgelegt, welche Prozeduren in welcher Reihenfolge auf die Objekte zugreifen, also welches Handlungsmuster ausgeführt wird. Der Zugriff, also die Beschreibung eines Handlungsmusters mit seinen einzelnen Handlungsschritten, wird exemplarisch am autorisierten Zugriff des Subjekts *Karteninhaber* in der Rolle *GB-Besitzer* und der Aufgabe *Bezahlen* beschrieben.

AutZugriff(s_1, r_1, a_1) = [(p_1, o_2), (p_8, o_1), (p_8, o_2), (p_7, o_3), (p_3, o_6)]

Dabei greift das Subjekt *Karteninhaber* in der Rolle *GB-Besitzer* mit der Aufgabe *Bezahlen* auf folgende Objekte mit folgenden Prozeduren zu: Zuerst wird das Tageslimit der Geldbörse gelesen (p_1, o_2), anschließend wird der Guthabenstand der Geldbörse um den gewünschten Betrag reduziert (p_8, o_1). Das verfügbare Tageslimit wird reduziert (p_8, o_2) und die Tagesverfügungen erhöht (p_7, o_3). Zum Schluß wird ein Eintrag an das Transaktionslog angehängt (p_3, o_6). Die Übersicht über die einzelnen Datenobjekte und Prozeduren folgt im Anschluß.

7.1.3 Objekte

Folgende Datenobjekte werden in der R&A-Anwendung verwendet und sind auf der R&A-Chipkarte gespeichert. Um die Liste der Datenobjekte inhaltlich zu strukturieren, wird zwischen Objekten unterschieden, die in den einzelnen Rollen verwendet werden. In der ersten Tabelle werden die Objekte beschrieben, die in der Rolle des Geldbörsenbesitzers von Bedeutung sind (Tabelle 7-4).

Geldbörsenobjekte	**Beschreibung**
(o_1) Guthabenstand GB	Dieses Objekt enthält den aktuellen Guthabenstand der Geldbörse.
(o_2) Tageslimit GB	Das Tageslimit ist der Betrag, über den maximal pro Tag verfügt werden kann. Dieses Objekt ist optional, es wird vom Karteninhaber eingerichtet.
(o_3) Tagesverfügungen GB	Dieses Objekt enthält die Summe der Beträge, die pro Tag von der Geldbörse abgebucht wurden.
(o_4) FehlbedienZähler GB	Dieses Objekt enthält die Fehlversuche bei der Authentisierung. Bei jeder falschen Authentisierung wird der Fehlbedienungszähler herabgezählt. Ist er gleich Null, wird die Geldbörse gesperrt.
(o_5) Authentisierung GB	Dieses Objekt enthält die Authentisierungsdaten des Geldbörsenbenutzers gegenüber dem System.
(o_6) Transaktionslog	Dieses Objekt enthält die Daten der letzten Transaktionen, die mit der Geldbörse getätigt wurden.
(o_7) Aufbuchungslog	Dieses Objekt enthält die Daten der letzten Aufbuchungen.
(o_8) Guthabenlimit GB	Dieses Objekt enthält das Limit für den Guthaben-stand der Geldbörse. Er wird von der Bank festgelegt.
(o_9) Anzahl Transaktionen	Dieses Objekt enthält die Anzahl der Transaktionen, die gespeichert werden sollen. Die Anzahl wird vom Karteninhaber festgelegt.
(o_{10}) Anzahl Aufbuchung	Dieses Objekt enthält die Anzahl der Aufbuchungen, die gespeichert werden sollen. Die Anzahl wird vom Karteninhaber festgelegt.

Tabelle 7-4: Geldbörsenobjekte

Die zweite Tabelle beschreibt die Datenobjekte, die in der Rolle *EC-Besitzer* notwendig sind. Dabei handelt es sich um Kontoinformationen über das (oder die) Girokonten, wie zum Beispiel Kontonummer, Bankleitzahl und Dispositionskredit. Weiterhin wird ein Gültigkeitsdatum gespeichert, das unabhängig von den Gültigkeitsdaten der anderen Anwendungen ist (Tabelle 7-5).

Prinzipiell können mehrere Konten mit der R&A-Chipkarte verwaltet werden. Es muß dann für jedes Konto ein eigener Datensatz wie in Tabelle 7-5 festgelegt werden. In diesem Beispiel wird sich nur auf ein Konto beschränkt. Äquivalentes gilt für die Anzahl der EC-Karten und Kreditkarten.

EC-Kartenobjekte	**Beschreibung**
(o_{11}) Gültigkeit EC	Dieses Objekt enthält das Gültigkeitsdatum der EC-Karte. Es wird von der Bank festgelegt und kann von dieser verlängert werden.
(o_{12}) Verfügungslimit EC	Dieses Objekt enthält den Betrag, über den pro Tag und Konto maximal verfügt werden darf. Es wird von der Bank festgelegt und kann von dieser verändert werden.
(o_{13}) Kontonr. (Giro)	Dieses Objekt enthält die Nummer des Girokontos.
(o_{14}) Konto existiert (Giro)	Dieses Objekt enthält die Information, ob ein Girokonto existiert. Diese wird von der Bank beim Anlegen beziehungsweise Löschen der Girokonten verändert.
(o_{15}) BLZ	Dieses Objekt enthält die Bankleitzahl.
(o_{16}) Authentisierung EC	Dieses Objekt enthält die Authentisierungsdaten des Benutzers gegenüber dem System.
(o_{17}) Autorisierung EC	Dieses Objekt enthält die Autorisierungsdaten des Benutzers gegenüber dem System.
(o_{18}) FehlbedienZähler EC	Dieses Objekt enthält die Fehlversuche bei der Authentisierung. Bei jeder falschen Authentisierung wird der Fehlbedienungszähler herabgezählt. Ist er gleich Null, wird die EC-Karte gesperrt.
(o_{19}) Transaktionslimit EC	Dieses Objekt enthält das Limit für den Betrag, über den maximal bei einer Transaktion verfügt werden kann. Dieses Objekt ist optional, es wird vom Karteninhaber eingerichtet.
(o_{20}) Tageslimit EC	Dieses Objekt enthält das Limit für den Betrag, über den maximal pro Tag und Konto verfügt werden kann. Dieses Objekt ist optional, es wird vom Karteninhaber eingerichtet.
(o_{21}) Tagesverfügungen EC	Dieses Objekt enthält die Summe der Beträge, die pro Tag und Konto abgebucht wurden.

Tabelle 7-5: EC-Kartenobjekte

Die Datenobjekte der Rolle *KK-Besitzer* beschreiben zum Beispiel die Gültigkeitsdaten der einzelnen Kreditkarten mit jeweiligen Kreditkartennummern. Weiterhin ist zu den Kreditkarten der jeweilige Kreditrahmen gespeichert, über den der Benutzer verfügen darf (Tabelle 7-6).

Kreditkartenobjekte	**Beschreibung**
(o_{22}) Gültigkeit KK	Dieses Objekt enthält das Gültigkeitsdatum der Kreditkarte. Es wird von der Bank festgelegt und kann von dieser verlängert werden.
(o_{23}) KK Nr.	Dieses Objekt enthält die Nummer der Kreditkarte.
(o_{24}) KK existiert	Dieses Objekt enthält die Information, ob eine Kreditkarte existiert. Diese wird von der Bank festgelegt.
(o_{25}) Kreditrahmen KK	Dieses Objekt enthält den Betrag, über den pro Tag und Karte maximal verfügt werden darf. Es wird von der Bank festgelegt und kann von dieser verändert werden.
(o_{26}) Hersteller ID	Dieses Objekt enthält die Identifikationsnummer des Herstellers.
(o_{27}) Authentisierung KK	Dieses Objekt enthält die Authentisierungsdaten des Benutzers gegenüber dem System.
(o_{28}) FehlbedienZähler KK	Dieses Objekt enthält die Fehlversuche bei der Authentisierung. Bei jeder falschen Authentisierung wird der Fehlbedienungszähler herabgezählt. Ist er gleich Null, wird die Kreditkarte gesperrt.
(o_{29}) Transaktionslimit KK	Dieses Objekt enthält das Limit für den Betrag, über den maximal bei einer Transaktion verfügt werden kann. Dieses Objekt ist optional, es wird vom Karteninhaber eingerichtet.
(o_{30}) Tageslimit KK	Dieses Objekt enthält das Limit für den Betrag, über den maximal pro Tag und Karte verfügt werden kann. Dieses Objekt ist optional, es wird vom Karteninhaber eingerichtet.
(o_{31}) Tagesverfügungen KK	Dieses Objekt enthält die Summe der Beträge, die pro Tag und Karte abgebucht wurden.

Tabelle 7-6: Kreditkartenobjekte

Für die Administration der R&A-Chipkarte sind verschiedene Datenobjekte notwendig. Da es sich hier um relativ wenige Daten handelt, werden in der Tabelle 7-7 die Datenobjekte für die Rollen der Subjekte *Bank* und *Karteninhaber* zusammengefaßt. Der *Karteninhaber* kann zum Beispiel Transaktionslimits für die einzelnen Aufgaben

festlegen. Weiterhin kann er Rollen, Aufgaben und Rollen-Aufgaben-Kombinationen deaktivieren, wenn er sie für eine bestimmte Zeit nicht erledigen möchte. Er kann sie jederzeit wieder aktivieren.

Administrationsobjekte	**Beschreibung**
(o_{32}) Authentisierung (Inh.)	Dieses Objekt enthält die Authentisierungsdaten des Benutzers gegenüber dem System.
(o_{33}) FehlbedienZähler (Inh.)	Dieses Objekt enthält die Fehlversuche bei der Authentisierung. Bei jeder falschen Authentisierung wird der Fehlbedienungszähler herabgezählt. Ist er gleich Null, werden die Benutzeradministrations-funktionen gesperrt.
(o_{34}) Authentisierung (Bank)	Dieses Objekt enthält die Authentisierungsdaten der Bank gegenüber dem System.
(o_{35}) Rollen, Aufgaben, R&A-Kombinationen deaktiviert (Inh.)	Dieses Objekt enthält die Information, ob bestimmte Rollen, Aufgaben und Rollen-Aufgaben-Kombina-tionen vom Benutzer deaktiviert sind.
(o_{36}) Rollen, Aufgaben, R&A-Kombinationen deaktiviert (Bank.)	Dieses Objekt enthält die Information, ob bestimmte Rollen, Aufgaben und Rollen-Aufgaben-Kombina-tionen von der Bank deaktiviert sind.
(o_{37}) Protokolldatei (Bank)	Hier werden die Aktionen des Bankadministrators gespeichert.

Tabelle 7-7: Administrationsobjekte

7.1.4 Prozeduren und Zugriffe

In den folgenden Tabellen wird gezeigt, welche Rollen-Aufgaben-Kombination welchen Prozedur-Objekt-Listen (Handlungsmuster) zugeordnet ist. Der Zugriff auf die oben beschriebenen Objekte erfolgt über Prozeduren.

Da jedes Objekt, das gelesen oder in irgendeiner Form verändert wird, sowohl geöffnet als auch geschlossen werden muß, wird auf die explizite Beschreibung des Öffnens und Schließens verzichtet. Dies sei für jedes Objekt, auf das zugegriffen wird, implizit vorausgesetzt. Ferner wird auf die Protokollierung der Initialisierungsfunktionen verzichtet. Beispielsweise muß das Objekt Tagesverfügungen jeden Tag, sofern ein Zugriff gewünscht wird, initialisiert werden, also negativ verändert werden. Dies sei ebenfalls implizit vorausgesetzt.

In Tabelle 7-8 sind alle für die R&A-Anwendung definierten Prozeduren aufgelistet.

Abkürzung	Prozedur	
(p_1) r	lesen	(read)
(p_2) w	schreiben	(write)
(p_3) a	anhängen	(append)
(p_4) i	Inhalt löschen	(initialise)
(p_5) c	erzeugen	(create)
(p_6) d	löschen	(delete)
(p_7) p	positiv verändern	(positive change)
(p_8) n	negativ verändern	(negative change)

Tabelle 7-8: Zulässige Prozeduren

Der detaillierte Zugriff auf die bereits vorgestellten Objekte soll anhand nachfolgender Tabelle 7-9 verdeutlicht werden. Als Beispiel wird ein Teil des Aufgabenbereiches *Finanztransaktionen* mit den Aufgaben *Bezahlen* und *Geld Akzeptieren* in allen zulässigen Rollen aufgegriffen. Auf eine vollständige Darstellung der verwendeten Prozeduren in den übrigen Rollen-Aufgaben-Kombinationen wird im Rahmen dieser Arbeit verzichtet. In der folgenden Tabelle werden nur die Datenobjekte aufgelistet, auf die mittels der Prozeduren zugegriffen wird. Alle anderen Objekte sind in der Tabelle nicht beschrieben.

Rollen/Aufgaben **Objekte**	**(r_1 / a_1) GB-Besitzer / Bezahlen**	**(r_1 / a_2) GB-Besitzer / Geld akzep**	**(r_2 / a_1) EC-Besitzer / Bezahlen**	**(r_3 / a_1) KK-Besitzer / Bezahlen**
(o_1) Guthabenstand GB	n	p		
(o_2) Tageslimit GB	r			
(o_3) Tagesverfüg. GB	p			
(o_5) Authentisierung GB	r			
(o_6) Transaktionslog	a / w			
(o_7) Aufbuchungslog		a / w		
(o_8) Guthabenlimit GB		r		
(o_9) Anzahl Transakt.	r			
(o_{10}) Anzahl Aufbuch.		r		

Fortsetzung ...

(o_{11}) Gültigkeit EC	r	r	r	
(o_{12}) Verfügungslimit EC			r	
(o_{13}) Kontonr. (Giro)			r	
(o_{14}) Konto existiert			r	
(o_{15}) BLZ			r	
(o_{16}) Authentisierung EC			r	
(o_{19}) Transakt.limit EC			r	
(o_{20}) Tageslimit EC			r	
(o_{21}) Tagesverfüg. EC			p	
(o_{22}) Gültigkeit KK				r
(o_{23}) KK Nr.				r
(o_{24}) KK existiert				r
(o_{25}) Kreditrahmen KK				r
(o_{26}) Hersteller ID KK				r
(o_{27}) Authentisier. KK				r
(o_{29}) Transakt.limit KK				r
(o_{30}) Tageslimit KK				r
(o_{31}) Tagesverfüg KK				p
(o_{35}) R, A, R&A deaktiv	r	r	r	
(o_{36}) R, A, R&A deaktiv				r

Tabelle 7-9: Beispielhafte Zugriffe auf Objekte

7.2 Dynamischer Aspekt des R&A-Modells

Nachdem der statische Aspekt des R&A-Modells in diesem Anwendungsbeispiel verdeutlicht wurde, wird nun der dynamische Aspekt anhand von zwei Benutzungsszenarien skizziert. Als Beispiel für die Darstellung des dynamischen Aspekts des R&A-Modells wird eine typische Benutzung der Chipkarte im elektronischen Zahlungsverkehr skizziert. Die Chipkarte hat zwei Subjekte als Benutzer, den *Karteninhaber* und die *Bank*. Es werden zwei Szenarien beschrieben. Im Ersten benutzt der *Karteninhaber* die R&A-Chipkarte zum Bezahlen mit der Geldbörse. Im Zweiten benutzt er sie zum Laden der Geldbörse und zum Setzen von Transaktionslimits.

Im ersten Szenario wählt der *Karteninhaber* genau eine Rolle und eine Aufgabe, die in der Rolle erledigt werden soll. Das bedeutet, daß im ersten Szenario keine Aktionen gleichzeitig ausgeführt werden. Das erste Szenario beschreibt die Situation, in der der *Karteninhaber* in der Rolle als Geldbörsenbesitzer bezahlen möchte.

Im zweiten Szenario wählt der *Karteninhaber* zwei Rollen-Aufgaben-Kombinationen, die gleichzeitig ausgeführt werden sollen. Der *Karteninhaber* wählt als erste Rollen-Aufgaben-Kombination, daß er seine Geldbörse laden möchte. Das zugehörige Handlungsmuster wird nicht sofort ausgeführt, sondern der *Karteninhaber* wählt als zweite Rollen-Aufgaben-Kombination das Setzen von einem Transaktionslimit in der Rolle als EC-Kartenbesitzer. Diese beiden Rollen-Aufgaben-Kombinationen schließen sich nicht gegenseitig aus (siehe Kapitel 7.1.2) und dürfen somit gleichzeitig ausgeführt werden.

Basis für die graphische Darstellung beider Szenarien ist das Zustandsdiagramm aus Kapitel 6.4.5, in dem alle möglichen Zustände und Überführungsfunktionen dargestellt sind. Für die jeweiligen Szenarien werden ausschließlich die Zustände und Überführungsfunktionen des Zustandsdiagramms beschrieben, die für das Szenario relevant sind.

7.2.1 Bezahlen mit der Geldbörse

Das erste Szenario beschreibt die Wahl einer Rolle und anschließend einer Aufgabe, die in der Rolle erledigt werden soll. Der *Karteninhaber* kauft in einem Geschäft eine Ware und entscheidet sich, mit seiner elektronischen Geldbörse zu bezahlen. Abbildung 7-6 beschreibt die Zustände und Überführungsfunktionen des Zustandsdiagramms und die entsprechenden Regeln, die erfüllt sein müssen.

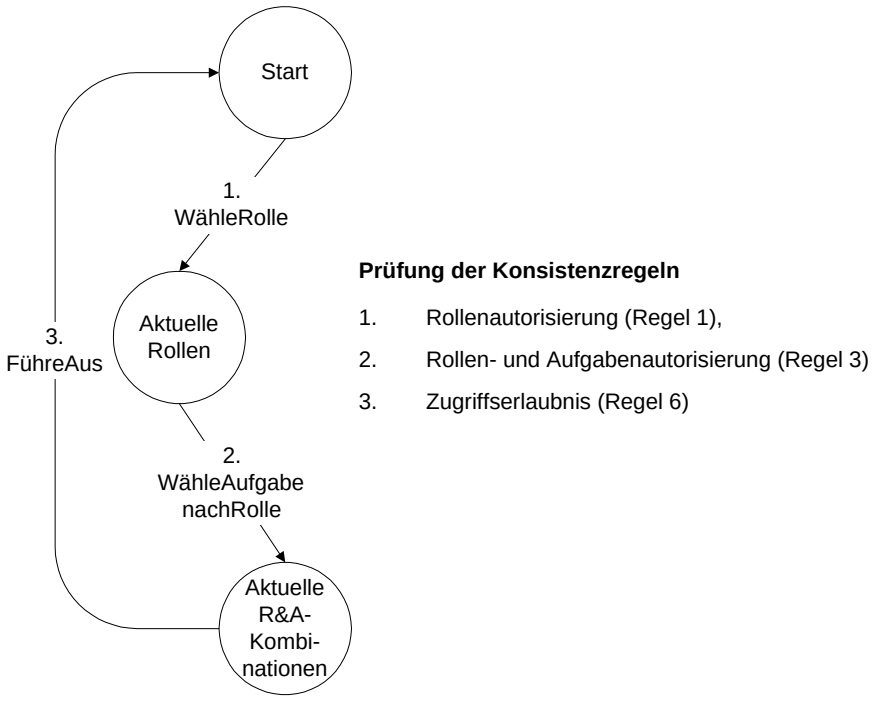


Abbildung 7-6: Bezahlen mit der Geldbörse

Der Ablauf des Szenarios 1 beginnt damit, daß das Subjekt *Karteninhaber* die Rolle *GB-Besitzer* durch die Zustandsüberführungsfunktion WähleRolle auswählt. Die Rolle *GB-Besitzer* befindet sich in der Menge der autorisierten Rollen für den *Karteninhaber*. Zu diesem Zeitpunkt hat der *Karteninhaber* noch keine aktuelle Rolle, so daß die Prüfung der dynamischen Trennung von Pflichten nicht stattfinden muß. Die, bisher leere, Menge der aktuellen Rollen wird um die Rolle *GB-Besitzer* erweitert.

WähleRolle (Karteninhaber, GB-Besitzer)

IF *GB-Besitzer* ∈ AutRollen (*Karteninhaber*)
AND
AktRollen (*Karteninhaber*) = ∅

THEN AktRollen* (*Karteninhaber*) = AktRollen (*Karteninhaber*) ∪
{*GB-Besitzer*}

ELSE „Fehlerbehandlung“

Danach wählt der *Karteninhaber* die Aufgabe *Bezahlen* mit der Überführungsfunktion WähleAufgabeNachRolle. Die Aufgabe *Bezahlen* gehört in der Rolle *GB-Besitzer* zu der Menge der autorisierten Rollen-Aufgaben-Kombinationen für das Subjekt *Karteninhaber*. Da der *Karteninhaber* bisher noch keine aktuelle Rollen-Aufgaben-Kombination ausgewählt hat, ist diese Menge leer. Aus diesem Grund muß die Regel der dynamischen Trennung von Pflichten nicht überprüft werden. Nach den Überprüfungen wird die, bisher noch leere, Menge der aktuellen Rollen-Aufgaben-Kombinationen um das Tupel (*GB-Besitzer, Bezahlen*) erweitert. Die Menge der aktuellen Aufgabe wird ebenfalls um die Aufgabe *Bezahlen* erweitert.

WähleAufgabeNachRolle (Karteninhaber, GB-Besitzer, Bezahlen)

IF (*GB-Besitzer, Bezahlen*) ∈ AutRollenUndAufgaben (*Karteninhaber*)
AND
AktRollenUndAufgaben (*Karteninhaber*) = ∅

THEN AktAufgaben* (*Karteninhaber*) =
AktAufgaben (*Karteninhaber*) ∪ {*Bezahlen*}
AktRollenUndAufgaben* (*Karteninhaber*) =
AktRollenUndAufgaben (*Karteninhaber*) ∪
{(*GB-Besitzer, Bezahlen*)}

ELSE „Fehlerbehandlung“

Nachdem nun eine Rolle und eine Aufgabe ausgewählt wurden, die die entsprechenden Regeln erfüllen, wird die Überführungsfunktion FührеAus ausgeführt, die überprüft, ob in der gegebenen Rollen-Aufgaben-Kombination ein Handlungsmuster existiert, dessen Handlungsschritte einzeln ausgeführt werden können. Diese Überprüfung erfolgt in der Funktion ErlaubterZugriff, die in der Regel Zugriffserlaubnis definiert ist. Nach Ausführen der Handlungsschritte wird die Menge der aktuellen Rollen-Aufgaben-Kombinationen um die Rollen-Aufgaben-Kombination (*GB-Besitzer, Bezahlen*) reduziert, die gerade abgearbeitet wurde. Ebenso wird die Menge der aktuellen Rollen um die Rolle *GB-Besitzer* und die Menge der aktuellen Aufgaben um die Aufgabe *Bezahlen* reduziert.

FührеAus(Karteninhaber, GB-Besitzer, Bezahlen)

IF ErlaubterZugriff

THEN „Ausführen“,

AktRollenUndAufgaben* (*Karteninhaber*) =

AktRollenUndAufgaben (*Karteninhaber*) \

{(*GB-Besitzer, Bezahlen*)}

AktRollen* (*Karteninhaber*) =

AktRollen (*Karteninhaber*) \ {*GB-Besitzer*}

AktAufgaben* (*Karteninhaber*) =

AktAufgaben (*Karteninhaber*) \ {*Bezahlen*}

ELSE „Fehlerbehandlung“

Äquivalent würde eine umgekehrte Reihenfolge bei der Auswahl einer Rolle und einer Aufgabe erfolgen. Wenn zuerst eine Aufgabe gewählt wird, prüft die Regel Aufgabenautorisierung (Regel 2), ob die Aufgabe für das Subjekt autorisiert ist. Ebenfalls wird bei einer anschließenden Wahl der Rolle geprüft, ob diese Rollen-Aufgaben-Kombination für das Subjekt autorisiert ist (Regel 3).

7.2.2 Geldbörse laden und Transaktionslimit für EC-Karte setzen

In diesem Szenario möchte der *Karteninhaber* zwei Aktionen gleichzeitig durchführen. Er befindet sich an einem Bankautomaten und möchte zum einen seine Geldbörse aufladen. Das Aufladen seiner Geldbörse soll von seinem Girokonto erfolgen. Dies kann er in der Rolle *GB-Besitzer* und der Aufgabe *Geld Transferieren* erledigen. Da der *Karteninhaber* nicht so häufig an einem Bankautomaten vorbeikommt, möchte er gleichzeitig auch eine zweite Rollen-Aufgaben-Kombination erledigen. Er möchte für Zahlungen mit seiner EC-Karte ein Transaktionslimit einrichten, so daß pro Zahlung ein bestimmter Betrag nicht überschritten werden darf. Dies kann er in der Rolle *EC-Besitzer* und der Aufgabe *Transaktionslimit* erledigen.

Beide Rollen-Aufgaben-Kombinationen sollen gleichzeitig ausgeführt werden. Dabei muß geprüft werden, ob diese Kombinationen sich nicht gegenseitig ausschließen. Wäre das der Fall, dürften sie nicht gleichzeitig ausgeführt werden.

Aus Anwendungssicht ist es für diesen speziellen Fall nicht unbedingt erforderlich, beide Rollen-Aufgaben-Kombinationen gleichzeitig zu erledigen. Dennoch soll anhand dieses einfachen Beispiels die Regel der dynamischen Trennung von Pflichten erläutert werden. Da das R&A-Modell keineswegs ausschließlich für eine Anwendung im elektronischen Zahlungsverkehr konzipiert wurde, sondern generell auf viele unterschiedliche Anwendung abbildbar ist, dient dieses Beispielszenario der Veranschaulichung der einzelnen Regeln.

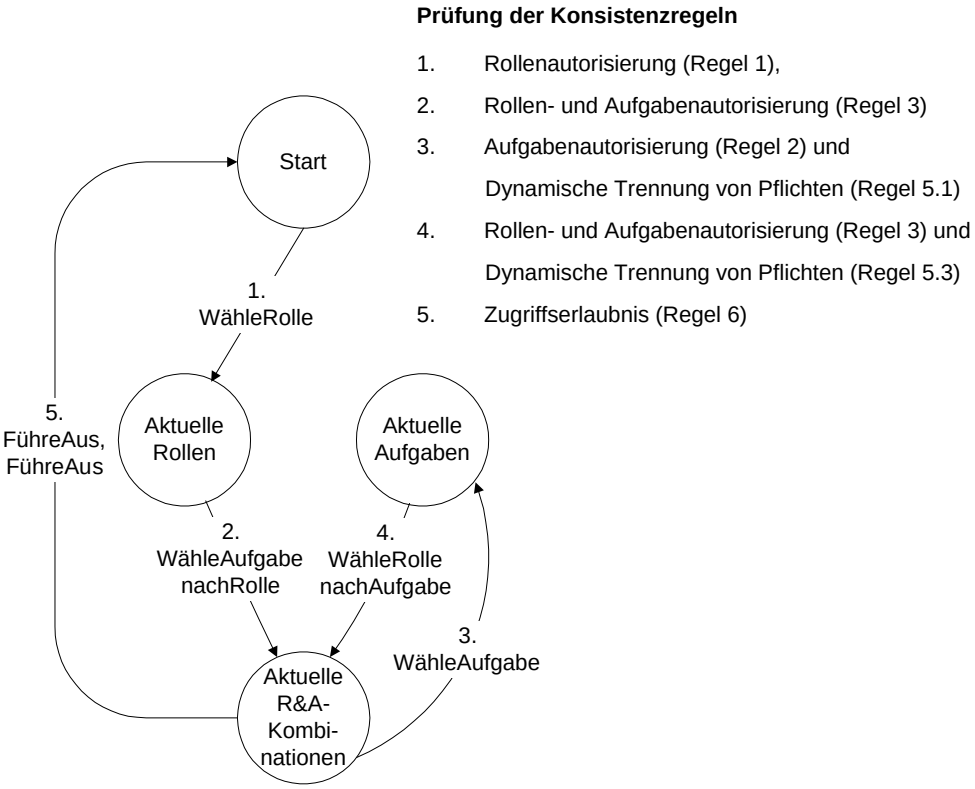


Abbildung 7-7: Geldbörse laden und Transaktionslimit für EC-Karte setzen

Der *Karteninhaber* wählt zunächst die Rolle *GB-Besitzer* durch die Überführungsfunktion WähleRolle. Die Rolle *GB-Besitzer* befindet sich in der Menge der autorisierten Rollen für den *Karteninhaber*. Zu diesem Zeitpunkt hat der *Karteninhaber* noch keine aktuelle Rolle, so daß die Prüfung der dynamischen Trennung von Pflichten nicht stattfinden muß. Die, bisher leere, Menge der aktuellen Rollen wird um die Rolle *GB-Besitzer* erweitert.

WähleRolle (Karteninhaber, GB-Besitzer)

IF *GB-Besitzer* ∈ AutRollen (*Karteninhaber*)

AND

AktRollen(*Karteninhaber*) = ∅

THEN AktRollen* (*Karteninhaber*) =

AktRollen (*Karteninhaber*) ∪ {*GB-Besitzer*}

ELSE „Fehlerbehandlung“

Danach wählt der *Karteninhaber* die Aufgabe *Geld Transferieren* mit der Überführungsfunktion WähleAufgabeNachRolle. Die Aufgabe *Geld Transferieren* und die vorher gewählte Rolle *GB-Besitzer* gehören zu den autorisierten Rollen-Aufgaben-Kombinationen für das Subjekt *Karteninhaber*. Da das Subjekt bisher noch keine aktuelle Rollen-Aufgaben-Kombination ausgewählt hat, ist diese Menge leer. Aus diesem Grund muß die Regel der dynamischen Trennung von Pflichten nicht überprüft werden. Nach den Überprüfungen wird die, bisher noch leere, Menge der aktuellen Rollen-Aufgaben-Kombinationen um die Kombination (*GB-Besitzer, Geld Transferieren*) erweitert. Die Menge der aktuellen Aufgaben wird um die Aufgabe *Geld Transferieren* erweitert.

WähleAufgabeNachRolle (Karteninhaber, GB-Besitzer, Geld Transferieren)

IF (*GB-Besitzer, Geld Transferieren*) ∈

AutRollenUndAufgabe (*Karteninhaber*)

AND

AktRollenUndAufgaben (*Karteninhaber*) = ∅

THEN AktAufgaben* (*Karteninhaber*) =

AktAufgaben (*Karteninhaber*) ∪ { *Geld Transferieren* }

AktRollenUndAufgaben* (*Karteninhaber*) =

AktRollenUndAufgaben* (*Karteninhaber*) ∪

{(*GB-Besitzer, Geld Transferieren*)}

ELSE „Fehlerbehandlung“

Die gewählte Rollen-Aufgaben-Kombination wird nun jedoch noch nicht sofort ausgeführt, sondern der *Karteninhaber* wählt eine weitere Kombination aus, die dann gleichzeitig mit der ersten Kombination ausgeführt werden soll. Für die zweite Rollen-Aufgaben-Kombination wählt der *Karteninhaber* nun zunächst erst eine Aufgabe und anschließend eine Rolle, in der die Aufgabe erledigt werden soll. Er wählt die Aufgabe

Transaktionslimit aus, die zu der Menge der autorisierten Aufgaben für den *Karteninhaber* gehört. Da der *Karteninhaber* nun schon die aktuelle Aufgabe *Geld Transferieren* hat, muß überprüft werden, ob sich diese beide Aufgaben gegenseitig ausschließen. Da dies nicht der Fall ist, kann die Menge der aktuellen Aufgaben für den *Karteninhaber* um die aktuelle Aufgabe *Transaktionslimit* erweitert werden. Die Menge der aktuellen Aufgaben enthält nun die beiden Elemente *Geld Transferieren* und *Transaktionslimit*.

WähleAufgabe (Karteninhaber, Transaktionslimit)

IF *Transaktionslimit* ∈ AutAufgaben (*Karteninhaber*)

AND

Geld Transferieren ∉ AusschlAktAufgaben (*Transaktionslimit*)

THEN AktAufgaben* (*Karteninhaber*) =

AktAufgaben (*Karteninhaber*) ∪ {*Transaktionslimit*}

ELSE „Fehlerbehandlung“

Danach wählt der *Karteninhaber* die Rolle *EC-Besitzer* mit der Überführungsfunktion WähleRolleNachAufgabe, da er die Aufgabe *Transaktionslimit* für seine EC-Karte erledigen möchte. Die Rolle *EC-Besitzer* und die Aufgabe *Transaktionslimit* gehört zu der Menge der autorisierten Rollen-Aufgaben-Kombinationen für das Subjekt *Karteninhaber*. Da der *Karteninhaber* bereits eine aktuelle Rollen-Aufgaben-Kombination hat, erfolgt an dieser Stelle die Prüfung der dynamischen Trennung von Pflichten. Es wird für die bereits aktuelle Rollen-Aufgaben-Kombination des *Karteninhabers* (*GB-Besitzer, Geld Transferieren*) geprüft, ob sie sich mit der angefragten Kombination (*EC-Besitzer, Transaktionslimit,*) gegenseitig ausschließt. Da dies nicht der Fall ist, kann die Menge der aktuellen Rollen-Aufgaben-Kombinationen um das Tupel (*EC-Besitzer, Transaktionslimit*) erweitert werden.

WähleRolleNachAufgabe (Karteninhaber, EC-Besitzer, Transaktionslimit)

IF (*EC-Besitzer, Transaktionslimit*) ∈

AutRollenUndAufgaben (*Karteninhaber*)

AND

(*GB-Besitzer, Geld Transferieren*) ∉

AusschlAktRollenUndAufgaben (*EC-Besitzer, Transaktionslimit*)

THEN AktRollen* (*Karteninhaber*) =

AktRollen (*Karteninhaber*) \ {*EC-Besitzer*}

AktRollenUndAufgaben* (*Karteninhaber*) =

AktRollenUndAufgaben (*Karteninhaber*) ∪

{(*EC-Besitzer, Transaktionslimit*)}

ELSE „Fehlerbehandlung“

Nachdem nun zwei Rollen-Aufgaben-Kombinationen ausgewählt wurden, die sich nicht gegenseitig ausschließen, können deren Handlungsmuster ausgeführt werden. Die Überführungsfunktion FührеAus wird in diesem Beispiel zweimal (parallel) ausgeführt. Nach Ausführen eines jeden Handlungsmusters wird die Menge der aktuellen Rollen um die jeweils gerade verwendete aktuelle Rolle (*GB-Besitzer* oder *EC-Besitzer*), die Menge der aktuellen Aufgaben um die entsprechende Aufgabe (*Geld Transferieren* oder *Transaktionslimit*) und die Menge der aktuellen Rollen-Aufgaben-Kombination um die entsprechende Kombination ((*GB-Besitzer, Geld Transferieren*) oder (*EC-Besitzer, Transaktionslimit*)) reduziert.

FührеAus (Karteninhaber, GB-Besitzer, Geld Transferieren)

IF ErlaubterZugriff (*Karteninhaber, GB-Besitzer, Geld Transferieren*)

THEN „Ausführen“,

AktRollenUndAufgaben* (*Karteninhaber*) =

AktRollenUndAufgaben (*Karteninhaber*) \

{(*GB-Besitzer, Geld Transferieren*)}

AktRollen* (*Karteninhaber*) =

AktRollen (*Karteninhaber*) \ {*GB-Besitzer*}

AktAufgaben* (*Karteninhaber*) =

AktAufgaben (*Karteninhaber*) \ { *Geld Transferieren* }

ELSE „Fehlerbehandlung“

FührеAus (Karteninhaber, EC-Besitzer, Transaktionslimit)

IF ErlaubterZugriffe (*Karteninhaber, EC-Besitzer, Transaktionslimit*)

THEN „Ausführen“,

AktRollenUndAufgaben* (*Karteninhaber*) =

AktRollenUndAufgaben (*Karteninhaber*) \

{(*EC-Besitzer, Transaktionslimit*)}

AktRollen* (*Karteninhaber*) =

AktRollen (*Karteninhaber*) \ {*EC-Besitzer*}

AktAufgaben* (*Karteninhaber*) =

AktAufgaben (*Karteninhaber*) \ {*Transaktionslimit*}

ELSE „Fehlerbehandlung“

In diesem Kapitel wurde die Anwendbarkeit des R&A-Modells gezeigt. Es wurde eine R&A-Anwendung modelliert, die eine R&A-Chipkarte im elektronischen Zahlungsverkehr beschreibt. Die R&A-Anwendung hat einen statischen und einen dynamischen Aspekt. Der statische Aspekt beschreibt die Variablen des R&A-Modells, also die Subjekte, Rollen, Aufgaben, Prozeduren und Objekte der R&A-Anwendung. Ebenso wurden die Funktionen in der R&A-Anwendung erläutert, die die einzelnen Variablen beschreiben.

Der dynamische Aspekt des R&A-Modells wurde in zwei Szenarien beschrieben. Diese beiden Szenarien haben exemplarisch, an ausgewählten Beispielen die Benutzung einer R&A-Chipkarte verdeutlicht. Dafür wurden einzelne Überführungsfunktionen ausgeführt und bei den entsprechenden Zustandsübergängen wurde geprüft, ob die jeweiligen Konsistenzregeln erfüllt sind. Das erste Szenario beschrieb die Auswahl und Ausführung einer einzigen Rollen-Aufgaben-Kombination, während das zweite Szenario die Auswahl von zwei Rollen-Aufgaben-Kombinationen beschrieb, die gleichzeitig ausgeführt wurden. Dafür kam die Regel der dynamischen Trennung von Pflichten zur Anwendung.

Nachdem in diesem Kapitel die Anwendbarkeit des R&A-Modells gezeigt wurde, muß in einem nächsten Schritt das R&A-Modell praktisch umgesetzt werden. Dafür muß ein Projekt in der Praxis initiiert werden, das eine konkrete R&A-Anwendung im elektronischen Zahlungsverkehr modelliert und auf einer R&A-Chipkarte implementiert. Die dafür notwendigen Schritte werden im folgenden Kapitel 8 nach einer Zusammenfassung dieser Arbeit beschrieben.

8 Ausblick

8.1 Zusammenfassung

Diese Arbeit hatte zum Ziel, die Sicherheit im elektronischen Zahlungsverkehr zu erhöhen. Um dieses Ziel zu erreichen, wurde ein neues Rollen- und Aufgabenbasiertes Sicherheitsmodell (R&A-Modell) entwickelt, das mit multifunktionalen Chipkarten im elektronischen Zahlungsverkehr realisiert werden kann. Dieses R&A-Modell stellt einen innovativen Schritt über die bereits existierenden Sicherheitsmodelle hinaus dar.

Bei der Entwicklung des Sicherheitsmodells wurde darauf geachtet, daß für den Benutzer kein Administrationsaufwand entsteht, da Benutzerrollen von Administrationsrollen getrennt sind. Die Kombination eines Sicherheitsmodells mit Chipkarten und dessen praktische Anwendung wurde in der Literatur bisher nicht erörtert. Im Hinblick auf die technische Entwicklung von Chipkarten und deren Anwendungsgebieten wird der Einsatz von Sicherheitsmodellen, die in Chipkarten integriert werden, jedoch unbedingt notwendig werden. Mit dieser Arbeit wurde ein wesentlicher Schritt in diese Richtung geleistet.

Zu Beginn der Arbeit wurde die Situation des elektronischen Zahlungsverkehrs, besonders die Entwicklung des karten- und netzgestützten Zahlungsverkehrs untersucht (siehe Kapitel 2.3). Es wurde herausgearbeitet, daß unterschiedliche Zahlungsverfahren sehr unterschiedliche Sicherheitsanforderungen an Vertraulichkeit, Anonymität, Integrität, Verläßlichkeit, Zurechenbarkeit, informationeller Selbstbestimmung und einfacher Handhabung stellen. Dabei wurde speziell auf die Bedürfnisse und Sicherheitsanforderungen von Benutzern im elektronischen Zahlungsverkehr eingegangen.

Die Entwicklung unterschiedlicher Zahlungsverfahren bis heute wurde beleuchtet, und es wurde auf zwei mögliche Trends hingewiesen (siehe Kapitel 2.5). Der eine Trend beschreibt die Entwicklung von vielen inhomogenen Zahlungssystemen, die meist auf je einer Chipkarte basieren. Der zweite Trend beschreibt die Kombination verschiedener Zahlungsverfahren auf einer einzigen multifunktionalen Chipkarte und wurde als Basis für weitere Überlegungen verwendet.

Da das Sicherheitsmodell in erster Linie für die Anwendung auf multifunktionalen Chipkarten entwickelt wurde, wurde zunächst die Technologie der Chipkarten beleuchtet. Es wurde die Funktionsweise und der Aufbau von Chipkarten beschrieben (siehe Kapitel 3), ebenso die Funktionsweise von bekannten Chipkarten-Betriebssystemen (siehe Kapitel 3.5). Weiterhin wurden die Grundlagen von kryptographischen Algorithmen erläutert (siehe Kapitel 4), die in Chipkarten Verwendung finden können. Das R&A-Modell kann auf der Basis von Chipkarten realisiert werden. Es ist jedoch auch möglich, das Sicherheitsmodell auf einer größeren Rechenanlage zu realisieren.

Bevor die Entscheidung über die Entwicklung eines spezifischen Sicherheitsmodells getroffen wurde, wurden bereits bekannte Sicherheitsmodelle vorgestellt und daraufhin

überprüft, ob sie für die gestellten Anforderungen in Frage kommen (siehe Kapitel 5). Dazu wurden, neben den klassischen Zugriffskontrollkonzepten wie DAC und MAC, das Vertraulichkeitsmodell von Bell und LaPadula und das Integritätsmodell von Clark und Wilson vorgestellt. Weiterhin wurden ein Telekooperationsmodell, ein formales Datenschutzmodell und ein Rollenmodell erläutert und anschließend bewertet. Das Ergebnis der Bewertung zeigt, daß keines der vorgestellten Sicherheitsmodelle, wegen ihrer Eindimensionalität (entweder Rollen oder Aufgaben), die gestellten Anforderungen zufriedenstellend erfüllt. Deshalb wurde ein neues, spezifisches Sicherheitsmodell mit zwei Dimensionen entwickelt (Rollen und Aufgaben).

Das spezifische Sicherheitsmodell stellt Rollen und Aufgaben für Benutzer zur Verfügung (siehe Kapitel 6). Die Aufgaben beschreiben, ***was*** ein Benutzer tun kann, während Rollen beschreiben, ***wie*** ein Benutzer eine Aufgabe erledigen kann. Aufgaben können in verschiedenen Rollen erledigt werden. Es existieren eine Menge von Aufgaben und eine Menge von Rollen, die für unterschiedliche Benutzer auf vielfältige Art und Weise miteinander kombiniert werden können. Jede Kombination von Rollen und Aufgaben entspricht einer konkreten Zugriffsmöglichkeit auf vorhandene Datenobjekte. Die Rollen-Aufgaben-Kombination beschreibt weiterhin, welche Information über den Benutzer freigegeben werden darf und auf welche Ressourcen zugegriffen werden darf. Jede Rollen-Aufgaben-Kombination definiert dadurch ein spezifisches Sicherheitsniveau, das durch die Wahl der Kombination von Rollen und Aufgaben realisiert wird. Durch die Kombinationsmöglichkeit der Elemente dieser zwei unabhängigen Mengen ergibt sich eine feine Granularität in der Realisierung des spezifischen Sicherheitsniveau. Die Sicherheit auf Benutzerseite kann dadurch wesentlich erhöht werden.

Das R&A-Modell unterstützt das Prinzip der Pflichtentrennung (separation of duty). Dieses Prinzip hat einen statischen und einen dynamischen Anteil. Die statische Trennung von Pflichten definiert, welche Rollen und Aufgaben für welche Benutzer autorisiert sind. Die dynamische Trennung von Pflichten definiert, welche Rollen-Aufgaben-Kombinationen zur selben Zeit ausgeführt werden dürfen, ohne Konflikte zu erzeugen. Dieses Konzept erlaubt eine gleichzeitige konfliktfreie Ausführung von Anwendungen. Die konfliktfreie gleichzeitige Ausführung existiert für elektronische Zahlungssysteme bisher noch nicht. Ist dies jedoch möglich, ergeben sich vielfältige neue Möglichkeiten in der Gestaltung des elektronischen Zahlungsverkehrs, wobei die Sicherheit für den Benutzer an erster Stelle stehen muß.

Das Sicherheitsmodell wurde in Form eines Zustandsautomaten formal definiert (siehe Kapitel 6.4). Es wurden die Zustandsvariablen beschrieben. Weiterhin wurden Regeln beschrieben, die in den einzelnen Zuständen gelten und Überführungsfunktionen definiert, die von einem Zustand in den nächsten überführen. Die formale Definition wurde in einer formalen Beweisskizze fortgeführt. Es wurde ausführlich die Beweisidee des Zustandsautomaten diskutiert, die im Anschluß skizziert wurde. Ein vollständiger Beweis ist nur dann notwendig, wenn die sichere Implementierung des R&A-Modells gewährleistet werden soll. Eine sichere Implementierung setzt sichere, also bewiesene, Betriebssysteme voraus. Im Chipkartenbereich gibt es zur Zeit jedoch keine formal bewiesenen Betriebssysteme. Aus diesem Grund kann eine sichere Implementierung

nicht garantiert werden. Es gibt bisher nur Chipkarten-Betriebssysteme, die nach ITSEC in der Stufe E4 zertifiziert wurden. Stufe E4 fordert jedoch keinen formalen Beweis der Software, sondern nur deren Spezifikation. Aus diesem Grund ist der hier skizzierte Beweis völlig ausreichend für die vorliegende und für weiterführende Arbeiten.

Bisher bekannte Sicherheitsmodelle sind in der Literatur vorwiegend informell beschrieben oder formal in Form von Zustandsautomaten. Zusätzlich zur Darstellung als Zustandsautomat und der Beweisskizze wurde das R&A-Modell auch als gefärbtes Petrinetz beschrieben (siehe Kapitel 6.5). Petrinetze erlauben in ihrer graphischen Darstellung eine komplexe Darstellungsweise des zugrundeliegenden R&A-Modells. Konflikte und Widersprüche in Systemen können mit der speziellen graphischen Repräsentation von Petrinetzen erkannt werden. Weiterhin konnten Nebenläufigkeiten im R&A-Modell dargestellt werden.

Eine Bewertung des R&A-Modells führt zu folgenden Sicherheitsaussagen (siehe Kapitel 6.6):

- Freie Wahl von Rollen und Aufgaben durch zweidimensionale Struktur
- Feine Granularität der Zugriffe
- Zugriff über wohldefinierte Prozeduren auf Objekte
- Gleichzeitige Ausführung von konfliktfreien Anwendungen
- Statische und dynamische Trennung von Pflichten
- Minimierung der notwendigen Rechte
- Vertraulichkeit und Integrität
- Zugriffsrechte nach MAC und DAC
- Leichte Administration
- Erhöhung der Akzeptanz

Das Sicherheitsmodell erfordert eine aktive Beteiligung des Benutzers. Durch die Wahl einer Rolle und einer Aufgabe kann der Benutzer selbst entscheiden, was er erledigen möchte und welche Informationen er dafür freigeben muß. Um die Benutzung und die Autorisierung der einzelnen Rollen und Aufgaben zu verdeutlichen, wurde nach der Definition des R&A-Modells ein Anwendungsbeispiel des R&A-Modells beschrieben (siehe Kapitel 7). Dabei wurde eine Anwendung einer multifunktionalen Chipkarte modelliert, die unterschiedliche Zahlungs- und Bankfunktionen auf einer Chipkarte erlaubt.

Das Anwendungsbeispiel unterscheidet einen statischen und einen dynamischen Aspekt. Der statische Aspekt beschreibt, daß die einzelnen Rollen, Aufgaben und Rollen-Aufgaben-Kombinationen für die jeweiligen Benutzer autorisiert sein müssen. Er beschreibt weiterhin, welche Rollen und Aufgaben sich gegenseitig ausschließen, um die Anforderungen an das Prinzip der statischen Trennung von Pflichten zu erfüllen.

Der dynamische Aspekt beschreibt beispielhaft eine Benutzung einer multifunktionalen Chipkarte im elektronischen Zahlungsverkehr. Sollen zwei oder mehr Rollen-Aufgaben-Kombinationen gleichzeitig ausgeführt werden, wird überprüft, ob das Prinzip der dynamischen Trennung von Pflichten verletzt wird und sich diese Kombinationen somit gegenseitig ausschließen.

In dem vorgestellten Anwendungsbeispiel waren alle Anwendungen mit ihren Rollen und Aufgaben von vorn herein bekannt und waren zu Beginn der Benutzung definiert. Ein Problem kann dann auftauchen, wenn zu einem späteren Zeitpunkt weitere Anwendungen zu bereits aktiven Anwendungen hinzu geladen werden sollen (siehe Kapitel 3.4). Es muß sichergestellt werden, daß die Anwendungen authentisch sind und die Funktionalität haben, die beschrieben wurde. Ein Ansatz für die Lösung des Problems können Zertifikate für Anwendungen sein, die nicht nur die Echtheit der Anwendung bestätigen, sondern auch eine nachprüfbare Funktionsbeschreibung enthalten.

Eine praktische Implementierung des R&A-Modells auf Basis einer multifunktionalen Chipkarte hätte den Rahmen dieser Arbeit gesprengt. Im folgenden werden interessante und sinnvolle Aufgaben beschrieben, die als Fortführung dieser Arbeit gesehen werden.

8.2 Weiterführende Aufgaben

Um das R&A-Modell auf einer multifunktionalen Chipkarte in einer konkreten Anwendung zu realisieren, müssen zunächst alle Rollen und Aufgaben in dieser Anwendung genau spezifiziert werden. Dies wurde in Kapitel 7 bereits getan. Weiterhin müssen alle Zugriffe, die sich aus den einzelnen Rollen-Aufgaben-Kombinationen ergeben, definiert werden. Aus der Anzahl der Rollen, Aufgaben und Zugriffe ergeben sich Anforderungen an die zugrundeliegende Chipkarte bezüglich Speicherplatz und Prozessorleistung. Wegen Speicherplatzmangel oder zu geringer Prozessorleistung kann in der Konzeptionierung der Anwendung die flexible Kombination von Rollen und Aufgaben eingeschränkt werden. So kann es sinnvoll sein, die gleichzeitige Ausführung von mehreren Rollen-Aufgaben-Kombinationen zu unterbinden. Das R&A-Modell stellt jedoch prinzipiell die volle, maximal flexible Funktionalität zur Verfügung.

Nach Wahl einer geeigneten Chipkarte, kann das R&A-Modell in das Chipkarten-Betriebssystem integriert werden. Dabei ist prinzipiell zu unterscheiden, ob Chipkarten mit „klassischen“ Betriebssystemen wie STARCOS (siehe Kapitel 3.5.6) oder „neue“ Chipkarten mit Java verwendet werden (siehe Kapitel 3.5.7).

Die starre Datenstruktur von STARCOS stellt bei der Integration des R&A-Modells in dieses Betriebssystem eine wesentliche Einschränkung dar. Die Datenstruktur erlaubt keine unterschiedlichen Zugriffe auf dieselben Daten eines Benutzers in verschiedenen Rollen. Der nötige Aufwand, um das R&A-Modell in STARCOS zu integrieren, steht in keiner Beziehung zu dem erwarteten Ergebnis. Java bietet mit dem Java-Card Environment und speziell dem Gateway Modell eine recht gute Möglichkeit, das R&A-Modell zu integrieren. Die Rollen, Aufgaben und restlichen Zustandsvariablen müssen entsprechend auf Java-Klassen abgebildet werden.

Bei der Gestaltung einer konkreten Anwendung können Aspekte des R&A-Modells bereits in den Designprozeß einfließen. Zur Zeit gibt es große Bestrebungen, Teile der Anwendungssoftware bereits im Chipdesign mit aufzunehmen. Da das R&A-Modell nicht nur in Chipkarten-Betriebssysteme integriert werden, sondern auch in die jeweilige Anwendung mit einfließen soll, ist es durchaus denkbar, es bereits in der Designphase der Chipentwicklung formal zu berücksichtigen.

Die Darstellung des R&A-Modells als Petrinetz kann in weiterführenden Arbeiten vertieft werden. So können Simulationswerkzeuge eingesetzt werden, die eine Petrinetzdarstellung verifizieren. Für die Implementierung können Simulationswerkzeuge weiterhin helfen, Abläufe zu demonstrieren. Petrinetze können als Proptotypen in der Softwareentwicklung verwendet werden.

Die Implementierung des R&A-Modells in einer konkreten Anwendung stellt hohe Anforderungen an die Gestaltung der Benutzungsoberfläche. Die Benutzung des R&A-Modells erfordert eine aktive Beteiligung des Benutzers. Um eine einfache Handhabung zu gewährleisten, muß die Benutzungsoberfläche einfach und intuitiv gestaltet sein. Dafür wird es notwendig sein, daß der Benutzer zusätzlich zur Chipkarte ein Lese-/ Schreibgerät mit sich führt, über das er mit der Chipkarte sicher kommunizieren kann. Mit Hilfe dieses Gerätes kann er die R&A-Anwendung bedienen, zum Beispiel kann er Transaktionslimits einrichten.

Alternativ können Chipkarten mit integrierter Ein- und Ausgabe verwendet werden, so daß sich die Notwendigkeit eines zusätzlichen Gerätes erübrigt. Aktuelle technische Entwicklungen haben es möglich gemacht, ein Display direkt auf der Chipkarte zu integrieren, ohne daß sich die Dicke der Chipkarte wesentlich erhöht hat. Für den vollständigen Ersatz der zusätzlichen Lese-/ Schreibgeräte ist lediglich eine platzsparende Tastatur notwendig. Bei weiterhin gleichbleibend hoher Innovationsgeschwindigkeit wird diese Entwicklung in naher Zukunft verfügbar sein. Im Hinblick auf einfache Handhabung können die Erweiterungsvorschläge aus Kapitel 6.7 umgesetzt werden.

In diesem Zusammenhang sind Akzeptanz-Studien notwendig, um herauszufinden, ob multifunktionale Chipkarten vom Benutzer akzeptiert werden. Dazu muß in Studien ermittelt werden, inwieweit die Benutzung des R&A-Modells auf der Basis von Chipkarten die Benutzerakzeptanz erhöht. Es stellt sich die prinzipielle Frage, ob Akzeptanz auf Vertrauen in die zu nutzenden Systeme oder auf fehlenden alternativen Nutzungsmöglichkeiten beruht. Beruht Akzeptanz auf Vertrauen, muß geklärt werden, wie Vertrauen als Basis zur Akzeptanz geschaffen werden kann. Der Ansatz einer Idee zur Beantwortung dieser Frage ist die Schaffung eines gesellschaftlichen Konsens. Ein gesellschaftlicher Konsens beschreibt auf der einen Seite die Partizipation der Betroffenen und auf der anderen Seite das Offenlegen der Pläne und Informationen von Seiten der Anbieter neuer Systeme. Ein gesellschaftlicher Konsens kann erreicht werden, wenn die Betroffenen, also die Benutzer von multifunktionalen Zahlungssystemen, frühzeitig in die Gestaltung miteinbezogen werden und am Entwicklungsprozeß teilhaben können.

Eine bisher noch ungelöste Frage kommt aus dem rechtlichen Bereich. Wer haftet bei unvorhersehbaren Fehlfunktionen auf multifunktionalen Chipkarten? Multifunktionale Karten haben in der Regel nicht nur einen Anwendungsanbieter, sondern viele unterschiedliche Anbieter mit unterschiedlichen Interessen. Sollte eine Anwendung eine andere ungewollt beeinflussen, ist bisher noch nicht geklärt, wer in einem solchen Fall die Haftung für einen möglichen Schaden trägt. Die fehlende Regelung einer Verantwortungsverteilung der Anbieter darf nicht dazu führen, daß die Haftung auf den Benutzer abgewälzt wird. Hier bedarf es einer schnellen Erarbeitung der notwendigen rechtlichen Regelungen, um eine reibungslose Benutzung von multifunktionalen Chipkarten mit unterschiedlichen Anwendungsanbietern zu ermöglichen.

Weiterhin müssen Verfahren zum Wiederherstellen (Recovery) von verlorenen Daten auf der Chipkarte und datenschutzkonforme Logging-Mechanismen entwickelt werden. Eine mögliche Lösung des Recovery-Problems könnte darin liegen, daß der Benutzer selbst sich um eine Art Backup für seine lokalen Daten kümmert und der jeweilige Anwendungsanbieter sich um seine lokalen Daten. Wer allerdings für die Informationen zuständig ist, die auf dem Übertragungsweg verlorengehen oder manipuliert werden, ist noch zu klären. Zusätzlich besteht bei einer solchen Lösung die Gefahr, daß jegliche Haftung auf den Benutzer abgewälzt wird, wenn dieser seine Backup-Verpflichtung nicht erfüllt hat.

Wird das R&A-Modell auf einer Chipkarte implementiert und im Bereich des elektronischen Zahlungsverkehrs eingesetzt, stellt sich die Frage, wie eine solche „R&A-Chipkarte“ in einem offenen System evaluiert werden kann. Die Evaluierung von Sicherheitskriterien gewinnt weltweit eine immer größere Bedeutung. Der Trend der Harmonisierung nationaler Kriterien zeugt von einem internationalen Bedarf, Sicherheitseigenschaften nachweisbar zu zertifizieren. Viele Sicherheitsprodukte tragen heutzutage ein solches Zertifikat, das Aussagen über den Herstellungsprozeß und die Sicherheitseigenschaften macht (siehe Kapitel 2.2.1).

Bisher können jedoch nur Sicherheitsaussagen über lokale geschlossene Systeme getroffen werden, zum Beispiel über Chipkarten und Lesegeräte oder über Betriebssysteme, solange sie nicht in einem Netzverbund arbeiten. Eine R&A-Chipkarte kann jedoch nicht lokal und unabhängig vom restlichen System zertifiziert werden. Da sie mit vielen Partnern in einem offenen System kommunizieren muß, kann diese Chipkarte nicht losgelöst vom restlichen Anwendungssystem betrachtet werden. Es muß überprüft werden, welche minimalen Evaluationsanforderungen erfüllt werden können und ob neue Evaluationsanforderungen gestellt werden müssen.

In dieser Arbeit wurde ein Rollen- und Aufgabenbasiertes Sicherheitsmodell entwickelt, das mit Hilfe einer Chipkarte für den Anwendungsbereich des elektronischen Zahlungsverkehrs realisiert werden kann. Der Einsatz des R&A-Modells beschränkt sich jedoch nicht auf den elektronischen Zahlungsverkehr, sondern kann in allen Anwendungsbereichen erfolgen, in denen individuelle Sicherheitsbedürfnisse unterstützt werden sollen. Wie bei jedem Produkt, das in einer neuen Anwendungsumgebung eingesetzt werden soll, gilt auch hier, den Aufwand gegen den Nutzen abzuwägen. Geeignete Anwendungsgebiete für das R&A-Modell sind zum Beispiel Anwendungen im

Gesundheitsbereich. Zum Beispiel könnte das komplexe Versorgungssystem eines Krankenhauses durch die Einteilung in Rollen und Aufgaben sinnvoll strukturiert werden. Das R&A-Modell könnte zur Regelung des Zugangs zu Räumen, Geräten oder Diensten durch Besucher, Personal, Wartungstechniker oder Sicherheitsbeauftragte in der Verwaltung, in Hochschulen, Unternehmen oder Hotels verwendet werden.

Weitere Anwendungen des R&A-Modells finden sich im Bereich der mobilen Kommunikation. Dort kommen schon seit langem Chipkarten zum Einsatz. Würde dort das R&A-Modell integriert, könnte der Benutzer eines tragbaren Telefons oder eines anderen portablen Gerätes in verschiedenen Rollen die unterschiedlichsten Aufgaben erledigen, wie zum Beispiel Telefonieren, Mail bearbeiten oder im Internet recherchieren. Die Integration des R&A-Modells muß so gestaltet werden, daß eine intuitive Benutzung des portablen Gerätes durch den zusätzlichen Sicherheitsgewinn nicht beeinträchtigt wird.

Eine Realisierung des R&A-Modells in einer praktischen R&A-Anwendung stellt eine sinnvolle Weiterentwicklung der vorliegenden Arbeit dar und ist ganz im Interesse der Autorin.

Anhang

Literatur

[Abrams 1993] Abrams, M.D.: Renewed understanding of access control policies, Proceedings of the 16. NIST National Computer Security Conference NCSC 1993, Baltimore, 1993, S. 87 - 96

[Abrams, Joyce 1995a] Abrams, M.D., Joyce, M.V.: Trusted system concepts, Computer and Security, Vol. 14, Nr. 1, Elsevier Science Publishers, Januar 1995, S. 45 - 56

[Abrams, Joyce 1995b] Abrams, M.D., Joyce, M.V.: Trusted computing update, Computer and Security, Vol. 14, Nr. 1, Elsevier Science Publishers, Januar 1995, S. 57 - 68

[Abrams, Joyce 1995c] Abrams, M.D., Joyce, M.V.: New thinking about information technology security, Computer and Security, Vol. 14, Nr. 1, Elsevier Science Publishers, Januar 1995, S. 69 - 81

[Abrams, Eggers, LaPadula, Olson 1990] Abrams, M.D., Eggers, K.W., LaPadula, L.J., Olson, I.M.: A Generalised Framework for Access Control: An informal description, Proceedings of the 13. NIST National Computer Security Conference NCSC 1990, Oktober 1990

[Abrams, LaPadula, Olson 1992] Abrams, M.D., LaPadula, L.J., Olson, I.M.: A Generalised Framework for Access Control: A Formal Rule Set for the ORGON Policy, M92B0000037, Mitre Corperation, 1992

[AgV 1998] AgV: Kaufen mit der GeldKarte ist nicht anonym, Arbeitsgemeischaft der Verbraucherverbände, März 1998
`http://www.agv.de/politik/geld/geldkarte.htm (23.11.1998)`

[Anderson, Kuhn 1996] Anderson, R., Kuhn, M.: Tamper Resistance - A Cautionary Note, Proceedings of the 2. Workshop on Electronic Commerce, Oakland, CA, November 1996

[Bachmeier 1995] Bachmeier, R.: Chipkarten und Datenschutz, Der GMD-Spiegel 2/95, S. 55 - 59

[Bangemann 1995] Bangemann, M.: Europas Weg in die Informationsgesellschaft, 13. IFIP World Congress, Hamburg 1994, in: Informatik Spektrum 18/1995, Springer Verlag, Heidelberg 1995

[Bank Austria 1998] Bank Austria: Beschreibung von Ecash in Betrieb der Bank Austria, Mai 1998
`http://www.bankaustria.com (18.01.1999)`

[Barkley 1997] Barkley, J.F.: Comparing Simple Role Based Access Control Models and Access Control Lists, in: Proceedings of the 2. ACM Workshop on Role-Based Access Control, ACM Press, New York, November 1997

[Barkley, Cincotta, Ferraiolo, Gavrilla, Kuhn 1997] Barkley, J.F., Cincotta, A., Ferraiolo, D.F., Gavrilla, S.I., Kuhn, D.R.: Role Based Access Control for the World Wide Web, In: Proceedings of the 20. National Information System Security Conference, NIST- NSA , 1997

[Bartmann, Fotschki 1997] Bartmann, D., Fotschki, C.: Elektronische Geldbörse, Gutachten im Auftrag der Friedrich-Ebert-Stiftung, Instituts für Bankinformatik, Universität Regensburg, satz + druck, Düsseldorf, 1997

[BdB 1997] Bundesverband deutscher Banken: Informationen zur Zweigstellendichte und zum Beschäftigtenverhältnis
`http://www.bdb.de/daten/zahlen/Markt.htm (20.11.1998)`

[BDSB 1993] Bundesbeauftragter für den Datenschutz: 14. Tätigkeitsbericht, Bonn, April 1993

[BDSB 1994] Konferenz der Datenschutzbeauftragten des Bundes und der Länder: Chipkarten im Gesundheitswesen, Beschluß der 47. Konferenz, Potsdam, März 1994

[BDSB 1995] Konferenz der Datenschutzbeauftragten des Bundes und der Länder: Datenschutzrechtliche Anforderungen an den Einsatz von Chipkarten im Gesundheitswesen, Beschluß der 50. Konferenz, Potsdam, November 1995

[Beier, Brannigan 1995] Beier, B.R., Brannigan, V.M.: Patient Privacy in the Era of Medical Computer Networks: A new paradigm for a new technology, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, Dezember 1995, S. 709 - 712

[Bell, LaPadula 1973] Bell, D.E., LaPadula, L.J.: Secure Computer Systems:
Mathematical Foundations, NTIS AD-770 768 ,
A Mathematical Model, NTIS AD-771 543,
A Refinement of the Mathematical Model, NTIS AD-780 528,
MTR 2547, Vol. I-III, ESD-TR-73-278, Mitre Corporation, Bedford, MA, 1973

[Bell, LaPadula 1976] Bell, D.E., LaPadula, L.J.: Secure Computer Systems: Unified Expositions and Multics Interpretation, NTIS AD-A023 588, MTR 2997, ESD-TR-75-306, Mitre Corporation, Bedford, MA, März 1976

[Bibow, Wichmann 1998] Bibow, J., Wichmann, T.: Elektronisches Geld: Funktionsweise und wirtschaftspolitische Konsequenzen, RWI-Mitteilungen, Zeitschrift für Wirtschaftsforschung
`http://www.berlecon.de/tw (20.11.1998)`

[BIS 1996] Bank for International Settlements: Implications for central banks of the development of electronic money, BIS, Basel, 1996

[Bizer 1994] Bizer, J.: Smartcards als Instrument rechtsverbindlicher Telekooperation, in Datow, Kissinger, Lange (Hrsg.): Proceedings der MultiCard 1994, InTIME, Berlin, Januar 1994

[Bizer 1995] Bizer, J.: Der gesetzliche Regelungsbedarf digitaler Signaturverfahren, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, August 1995

[Blaze, Diffie, Rivest, Schneier, Shimomura, Thomson, Wiener 1996] Blaze, M., Diffie, W., Rivest, L.R., Schneier, B., Shimomura, T., Thomson, E., Wiener, M.: Minimal Key Lengths for Symmetric Ciphers to provide Adequate Commercial Security, Januar 1996

[BMBF 1997] Bundesministerium für Bildung und Forschung: Gesetz und Verordnung zur Digitalen Signatur, Bonn, 1997
`http://www.iid.de/iukdg/ (18.01.1999)`

[Böhle, Riehm 1998] Böhle, K., Riehm, U.: Blütenträume - Über Zahlungssystem-innovationen und Internet-Handel in Deutschland, Institut für Technik-folgenabschätzung und Systemanalyse, Wissenschaftliche Berichte FZKA 6161, Forschungszentrum Karlsruhe GmbH, Karlsruhe, Dezember 1998

[Bräutigam, Höller, Scholz 1990] Bräutigam, L., Höller, H., Scholz, R.: Datenschutz als Anforderung an die Systemgestaltung, Westdeutscher Verlag, 1990

[Brockhaus 1991] Brockhaus Enzyklopädie, 19. Auflage, Brockhaus-Verlag, Mannheim, 1991

[Bronstein, Semendjajew 1989] Bronstein, I.N., Semendjajew, K.A.: Taschenbuch der Mathematik, 24. Auflage, B.G. Teubner, Leipzig, 1979

[Brunnstein 1997] Brunnstein, K.: Towards a holistic View of Security and Safety of Enterprise Information and Communication Technologies: Adapting to a changing Paradigm, in: Yngström, L., Carlsen, J.: Information Security in Research and Business, Proceedings of the IFIP TC-11 SEC 1997 Konferenz, Chapman & Hall, Mai 1997

[Brunnstein, Fischer-Hübner 1990] Brunnstein, K., Fischer-Hübner, S.: Risk Analysis of „Trusted“ Computer Systems, Proceedings of the IFIP TC-11 SEC 1990 Konferenz, Helsinki, Mai 1990

[Brunnstein, Schier 1997a] Brunnstein, K., Schier, K., Global Digital Commerce: Impacts and Risks for Developments of Global Information Societies”, in: Berleur, J., Whitehouse, D. (Hrsg.): An ethical global information society: culture and democracy revisited, Proceedings of the IFIP WG 9.2 Corfu international conference, 8.-10. Mai 1997, Chapman & Hall,1997, S. 75 - 82

[Brunnstein, Schier 1997b] Brunnstein, K., Schier, K.: Sicherheitsrisiken beim Online-Banking, GI Geldinstitute, Heft 11-12, Dezember 1997, 28. Jahrgang, Hans Holzmann Verlag, Bad Wörishofen, 1997, S. 64 - 66

[BSI 1998] Bundesamt für Sicherheit in der Informationstechnik: IT-Grundschutzhandbuch, Bonn, 1998
http://www.bsi.de/gshb/deutsch/menue.htm (18.01.1999)

[Bürk, Pfitzmann 1990] Bürk, H., Pfitzmann, A.: Value Exchange Systems Enabling Security and Unobservability, Computers and Security, Vol. 9, Elsevier Science Publishers, 1990, S. 715 - 721

[Bundesregierung 1997] Bundesregierung: Entwurf eines Gesetzes zur Umsetzung von EG-Richtlinien zur Harmonisierung bank- und wertpapieraufsichtsrechtlicher Vorschriften, Bundestagsdrucksache 13/7142, Bundesanzeiger Verlagsgesellschaft, Bonn, März 1997
http://dip.bundestag.de (25.01.1999)

[BVerfGE 1986] Bundesverfassungsgericht: Bundesverfassungsgerichtsentscheid Nr. 45, Neue Juristische Wochenzeitschrift, 1986, S. 187ff

[CEPTF 1998] Consumer Electronic Payment Task Force: The Report of the Consumer Electronic Payment Task Force, Mai 1998
http://www.occ.treas.gov/emoney/ (20.11.1998)

[Chaum 1985] Chaum, D.: Security without identification: Card Computers to make Big Brother Obsolete, Communications of the ACM 28(19), ACM Press, 1985, S. 130 - 1044

[Chaum 1987] Chaum, D.: Sicherheit ohne Identifizierung, Informatik-Spektrum 10, 1987, S. 262 - 277

[Chaum 1992] Chaum, D.: Achieving electronic privacy, Scientific American, August 1992, S. 96 - 101

[Chaum 1996] Chaum, D.: An introduction to ecash, Proceedings of the IMAGINA 1996, Monaco, Februar 1996, S. 261 - 270

[Chen, Sandhu 1995] Chen, F., Sandhu, R.S.: Constraints for Role-Based Access Control, in: Proceedings of the 1. ACM Workshop on Role-Based Access Control, ACM Press, New York, November 1995

[Clark, Wilson 1987] Clark, D.D., Wilson, D.R.: A Comparism of Commercial and Military Computer Security Policies, Proceedings of the IEEE Symposium on Security and Privacy 1987, 1987, S. 184 - 194

[Clark, Wilson 1989] Clark, D.D., Wilson, D.R.: Evolution of a Model for Computer Integrity, Report of the Invitational Workshop on Data Integrity, NIST Publication 500-168, 1989, Section A2, S. 1 - 13

[CZ46 1998] O.V.: Pionier Digicash gesteht Pleite, Computerzeitung Nr. 46, Konradin Verlag, Leinfelden-Echterdingen, November 1998, S. 2

[Deininger, Lichter, Ludewig, Schneider 1996] Deininger, M., Lichter, H., Ludewig, J., Schneider, K.: Durchführung und Betreuung von Studien- Diplom- und Doktorarbeiten am Beispiel Informatik, 3. Auflage, vdf Hochschulverlag AG an der ETH Zürich, B.G. Teubner, Stuttgart, 1996

[DES-Challenge 1997] Distributed Net: DES Challenge I, Juni 1997
http://www.distributed.net/des (21.01.1999)

[DES-Challenge 1998] RSA Laboratories: DES Challenge II, Februar 1998
http://www.rsa.com/rsalabs/des2/ (21.01.1999)

[DES-Challenge 1999] Distributed Net: DES Challenge III, Januar 1999
http://www.distributed.net/pressroom/press-des-iii.html (21.01.1999)

[Dethloff 1968] Dethloff, J.: Deutsche Patenschrift OS DE 27 38 113, Deutsches Patentamt München, 1968

[Dethloff 1997] Dethloff, J.: Will „Bit Money“ Make The World Go Round?, Card Technology, Ausgabe 10, Oktober 1997

[Dethloff, Grötrupp 1968] Dethloff, J., Grötrupp, H.: Deutsche Patenschrift DE 1945777 C3, Deutsches Patentamt München, 1968

[Diffie, Hellman 1976] Diffie, W., Hellman, M.E.: New Directions in Cryptography, in: IEEE Transactions on Information Theory, Vol. 22, Nr. 6, IEEE Computer Society Press, Juni 1976

[Digicash 1996] Digicash: Produktinformation zu Digicash
http://www.digicash.com/ecash/ecash-home.html (20.11.1998)

[Digicash 1998] Digicash: Bank Austria goes Live with Digicash’s Ecash, Ankündigung, Mai 1998
http://www.digicash.com/index_e.html (21.01.1999)

[DoD 1985] United States of America, Department of Defense: Trusted Computer Systems Evaluation Criteria (TCSEC), DoD 5200.28-STD, Dezember 1985

[Dresdner Bank 1998] Dresdner Bank: Cybercash - Sicheres Bezahlen im Internet
http://www.dresdner-bank.de/
f_firmen/b_office/c_cash/home.htm (04.12.1998)

[DVD 1992] Deutsche Vereinigung für Datenschutz: Die Krankenversichertenkarte gefährdet Ihre Gesundheit, Bremen, 1992

[DVD 1995] Deutsche Vereinigung für Datenschutz: Die neue Bahncard gefährdet Ihr Persönlichkeitsrecht, Pressemitteilung, Bonn, Juli 1995

[Eberl 1998] Eberl, U.: Eintrittskarte in die Welt von morgen, in: Siemens (Hrsg.): Forschung und Innovation, Die Zeitschrift für Wissenschaft und Technik, 1/98, Siemens, 1998, S. 9 - 13

[Eisele 1995] Eisele, R.: Sicherheit und elektronische Unterschriften - SmartDisk, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, Juli 1995

[EMV 1996] Europay, Mastercard und Visa (EMV): Specification for Payment Systems, Version 3.0, Juni 1996

[Engel, Lessig 1997] Engel, A., Lessig, A.: Elektronische Zahlungsmittel im Internet, Übersicht und Bewertung aktueller Verfahren unter Berücksichtigung von Kriterien der Sicherheit und Funktionalität, Studienarbeit am Fachbereich Informatik, Universität Hamburg, Hamburg, Dezember 1997

[Engel, Lessig, Schier 1998] Engel, A., Lessig, A., Schier, K.: Chipkartenbasierte Zahlungssysteme - Der Große Bruder im Portemonnaie, Proceedings der OmniCard 1998, Die Chipkarte auf dem Weg zu Akzeptanz und Nutzung, Berlin, Januar 1998, S. 19 - 43

[Engelbrecht, Hildebrand, Jung 1994] Engelbrecht, R., Hildebrand, C., Jung, E.: Smart Cards as communication Tools in Health Information Systems, in: Brunnstein, K., Raubold, E.: 13. World Computer Congress 1994, Vol. 2, Elsevier Science Publishers, September 1994

[Europäische Kommission 1998] Europäische Kommission:Entwurfs eines Vorschlags einer Richtlinie des Europäischen Parlaments und des Rates über die Aufnahme, Ausübung, und Beaufsichtigung der Tätigkeit von E-Geldinstituten, Brüssel, 1998
`http://europa.eu.int/comm/dg15/de/finances/general/727.htm (25.01.1999)`

[Europäische Zentralbank 1998] Europäische Zentralbank: Bericht über elektronisches Geld, Frankfurt, 1998
`http://www.bundesbank.de/de/presse/wwu/ezb-veroeff.htm (25.01.1999)`

[Ferraiolo, Cugini, Kuhn 1995] Ferraiolo, D.F., Cugini, J.A., Kuhn, D.R.: Role Based Access Control (RBAC): Features and Motivation, 11. Annual Computer Security Applications Conference ACSAC 1995, New Orleans, IEEE Computer Society Press, Los Alamitos, Dezember 1995, S. 241 - 248

[Ferraiolo, Gilbert, Lynch 1993] Ferraiolo, D.F., Gilbert, D.M., Lynch, N.: An examination of federal and commercial access control policy needs, Proceedings of the 16. NIST National Computer Security Conference NCSC 1993, Baltimore, MD, 1993, S. 107 - 116

[Ferraiolo, Kuhn 1992] Ferraiolo, D.F., Kuhn, D.R.: Role Based Access Control, Proceedings of the 15. NIST National Computer Security Conference NCSC 1992, Baltimore, MD, October 1992, S. 554 - 563

[Finney 1993] Finney, H.: Detecting Double Spending
http://www.eff.org/pub/Privacy/Digital_money/ (20.11.1998)

[FirstSurf 1998] FirstSurf: Internetshopping Report 1998/99, Die große Nutzerumfrage, Käufer, Produkte, Zukunftsaussichten, Symposium Publishing, Düsseldorf, 1998
http://www.firstsurf.com/shoppingumfrage_text.html
(20.11.1998)

[Fischer-Hübner 1994a] Fischer-Hübner, S.: Ein formales Datenschutz-Modell, Proceedings der Fachtagung Sicherheit in Informationssystemen, vdf Verlag der Fachvereine, Zürich, März 1994

[Fischer-Hübner 1994b] Fischer-Hübner, S.: Towards a privacy friendly design and use of IT-security mechanisms, Proceedings of the 17. NIST National Computer Security Conference NCSC 1994, Baltimore, MD, Oktober 1994

[Fischer-Hübner 1995] Fischer-Hübner, S.: Considering Privacy as a Security-Aspect: A Formal Privacy-Model, DASY Paper No 5/95, Institute of Computer and Systems Sciences, Copenhagen Business School, Kopenhagen, Mai 1995

[Fischer-Hübner 1997] Fischer-Hübner, S.: A formal task-based privacy model and its implementation: An updated report, Proceedings of the 2. Nordic Workshop on secure computer systems NORDSEC ’97, Helsinki, November 1997

[Fischer-Hübner 1999] Fischer-Hübner, S.: Entwurf der Habilitationsschrift „Privacy-Enhancing Design and Use of Security Mechanisms“, 1999 (bisher unveröffentlicht)

[Fischer-Hübner, Schier 1996a] Fischer-Hübner, S., Schier, K.: Der Weg in die Informationsgesellschaft - Eine Gefahr für den Datenschutz?, in: Schinzel B. (Hrsg.): Schnittstellen, Vieweg, 1996

[Fischer-Hübner, Schier 1996b] Fischer-Hübner, S., Schier, K.: Risks on the way to the global information society, in: Katsikas, S.K., Gritzalis, D. (Hrsg.): Information Systems Security - Facing the information society of the 21. Century, Proceedings of the IFIP TC-11 SEC 1996, Chapman & Hall, Mai 1996, S. 487 - 488

[Ford, Baum 1997] Ford, W., Baum, M.S.: Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption, Prentice Hall, New Jersey, 1997

[Fox, Röhm 1996] Fox, D., Röhm, A.: Effiziente digitale Signatursysteme auf der Basis elliptischer Kurven, in: Proceedings der Arbeitskonferenz Digitale Signaturen, DuD Fachbeiträge, Vieweg, Wiesbaden, 1996

[Freier, Karlton, Kocher 1996] Freier, A.O., Karlton, P., Kocher, P.C.: Internet Draft: The SSL Protocol Version 3.0
`ftp://ds.internic.net/`
`internet-drafts/draft-freier-ssl-version3-01.txt`
`(20.11.1998)`

[Fumy 1995] Fumy, W.: Authentifizierung und Schlüsselmanagement, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, Oktober 1995, S. 607 - 613

[Garfinkel 1995] Garfinkel, S.: PGP: Pretty Good Privacy, O'Reilly & Associates, Sebastopol, 1995

[Gasser 1988] Gasser, M.: Building a secure Computer System, Van Nostrand Reinhold, New York, 1988

[Gavrila, Barkley] Gavrila, S.I., Barkley, J.F.: Formal Specification for Role Based Access Control User/Role and Role/Role Relationship Management, in: Proceedings of the 3. ACM Workshop on Role-Based Access Control, ACM Press, New York, Oktober 1998

[Gentz 1997] Gentz, W.: Die elektronische Geldbörse in Deutschland: Funktionsweise, Kosten und Nutzen für die Beteiligten, Diplomarbeit, Fachhochschule München, München, November 1997

[Grill, Gramlich, Eller 1998] Grill, W., Gramlich, L., Eller, R.: Gabler-Bank-Lexikon: Bank - Börse - Finanzierung, 11. Auflage, Gabler, Wiesbaden, 1996

[Grimm 1992] Grimm, R.: A Model of Security in Open Telecooperation, IFIP Transactions 07, Proceedings of the IFIP TC-6 WG-5 Working Conference on Upper Layer Protocols, Architectures and Applications (ULPAA 1992), Vancouver, CA, Elsevier Science Publishers, Mai 1992, S. 425 - 440

[Grimm 1993a] Grimm, R.: Ein Sicherheitsmodell für verbindliche Kooperation in offenen Systemen, in: Der GMD-Spiegel 2/1993, 1993, S. 45 - 51

[Grimm 1993b] Grimm, R.: Non-Repudiation in Open Telecommunication, Proceedings of the 16. NIST National Computer Security Conference NCSC 1993, Baltimore, MD, September 1993, S. 16 - 30

[Grimm 1994] Grimm, R.: Sicherheit für offene Kommunikation - Verbindliche Telekooperation, in: Horster P. (Hrsg.): Sicherheit in der Informations- und Kommunikationstechnik, Band 4, BI Wissenschaftsverlag, Mannheim, 1994

[Grimm 1998] Grimm, R.: Die Rolle der Chipkarte im Electronic Commerce, in Fluhr, M. (Hsrg.): Die Chipkarte auf dem Weg zu Akzeptanz und Nutzung, Proceedings der OmniCard 1998, InTIME, Berlin, Januar 1998

[Grimm, Zangeneh 1996] Grimm, R., Zangeneh, K.: Cybermoney in the Internet: An Overview over new Payment Systems in the Internet, in: Communications and Multimedia Security, IFIP, Chapman & Hall, London, 1996, S. 183 - 195

[Guiri 1995a] Guiri, L.: A New Model for Role Based Access Control, 11. Annual Computer Security Applications Conference ACSAC 1995, New Orleans, IEEE Computer Society Press, Los Alamitos, Dezember 1995, S. 249 - 255

[Guiri 1995b] Guiri, L.: Role Based Access Control: A Natural Approach, in: Proceedings of the 1. ACM Workshop on Role-Based Access Control, ACM Press, New York, November 1995

[Hamann, Hirsch, Ondrusch 1995] Hamann, U., Hirsch, S., Ondrusch, S.: Krypto-Chipkarten - Sicherheit für jedermann, Siemens-Zeitschrift Special, FuE, 1995

[Hammer 1994] Hammer, V.: Beweiswert digital signierter Dokumente, in: Datow, Kissinger, Lange (Hrsg.): Proceedings der MultiCard 1994, InTIME, Berlin, Januar 1994

[Hammer 1995a] Hammer, V.: Vor- und Nachteile von Mehrfachzertifikaten für öffentliche Schlüssel, in: Bundesamt für Sicherheit in der Informationstechnik Fachvorträge: 4. Deutscher IT-Sicherheitskongreß, Bad Godesberg, Mai 1995

[Hammer 1995b] Hammer, V.: Digitale Signaturen mit integrierter Zertifikatskette, in: Brüggenmann, H., Gerhardt-Häckl, W. (Hrsg.): Verläßliche IT-Systeme, Proceedings der GI-Fachtagung VIS 1995, Vieweg, 1995

[Hammer 1995c] Hammer, V. (Hrsg.): Sicherungsinfrastrukturen: Gestaltungsvorschläge für Technik, Organisation und Recht, Springer Verlag, Berlin, 1995

[Heiring 1998] Heiring, W.: Sicherheit im GeldKarte-System, Banken Aktuell, Arbeitsmaterialien für den Unterricht
`http://www.stam.de/ibankak.htm (23.11.1998)`

[HmbDSB 1995] Hamburgischer Datenschutzbeauftragter: Kartengestützter Zahlungsverkehr, 14. Tätigkeitsbericht des Hamburgischen Datenschutzbeauftragten
`http://www.hamburg.de/Behoerden/HmbDSB/TB14/25_1.htm (23.11.1998)`

[Hühnlein 1996] Hühnlein, D.: Effiziente Exponation und optimale Punktdarstellung für Signatursysteme auf Basis elliptischer Kurven, in: Proceedings der Arbeitskonferenz Digitale Signaturen, DuD Fachbeiträge, Vieweg, Wiesbaden, 1996

[Hühnlein 1998] Hühnlein, D.: Implementierung Elliptischer Kurven auf Chipkarten, in: Proceedings der Arbeitskonferenz Chipkarten, DuD Fachbeiträge, Vieweg, Wiesbaden, 1998

[IBI 1997] Institut für Bankinformatik: Befragung der deutschen Bevölkerung über 14 Jahre nach heute genutzten und in Zukunft erwünschten komplexen Online-Diensten, Institut für Bankinformatik, Universität Regensburg, Regensburg, 1997
`http://www.rrwnt2.uni-regensburg.de/IBI/index.htm (07.04.1998)`

[ITSEC 1991] Amt für amtliche Veröffentlichungen der europäischen Gemeinschaft: Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik (ITSEC), Version 1.2, Luxemburg, 1991

[Jacob 1994] Jacob, J.W.: Forum Vertrauenswürdige Informationstechnik für Medizin und Gesundheitsverwaltung, Vortrag, Bonn, September 1994

[Jantzen, Valk 1998] Jantzen M., Valk, R.: Theoretische Grundlagen der Programmierung, Skript zur Vorlesung, Fachbereich Informatik, Universität Hamburg, Hamburg, Oktober 1998

[Java Card Forum 1997] Java-Card Forum: Java-Card Version 2.0, API Specification, Draft Proposal, September, 1997

[Kabel 1996] Kabel, N.: Die branchenübergreifende elektronische Geldbörse - technische Analyse und Einsatzmöglichkeiten im elektronischen Zahlungsverkehr, Diplomarbeit am Institut für Informatik und Gesellschaft, Technische Universität Berlin, Berlin, Dezember 1996

[Kessler, Mund 1993] Kessler, V., Mund, S.: Sicherheitsmodelle, Studie, Siemens, Juni 1993

[Kiranas 1995] Kiranas, A.: Point-Of-Sale (POS)-Systeme - Kryptologie, Magnetstreifen- und Chipkarten als Sicherheitswerkzeuge, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, Dezember 1995, S. 721 - 728

[Klein 1995] Klein, E.: IT-Sicherheit von Chipkarten, in: Bundesamt für Sicherheit in der Informationstechnik Fachvorträge: 4. Deutscher IT-Sicherheitskongreß, Bad Godesberg, Mai 1995

[Knorr, Schläger 1997] Knorr, M., Schläger, U.: Datenschutz bei elektronischem Geld, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, September 1995, S. 396 - 402

[Koblitz 1987] Koblitz, N.: Elliptic Curve Cryptosystems, in: Mathematics of Computations, Band 48, 1987, S. 203 - 209

[Kock 1997] Kock, C.: Abbildung eines Rollen- und Aufgabenbasierten Zugriffskonzeptes anhand eines Anwendungsbeispiels auf den elektronischen Zahlungsverkehr, Studienarbeit am Fachbereich Informatik, Universität Hamburg, Hamburg, Dezember 1997

[Kock 1999] Kock, C.: Konzeptionelle Integration des Rollen- und Aufgabenbasierten Sicherheitsmodell in ein Chipkarten-Betriebssystem am Beispiel STARCOS, Diplomarbeit am Fachbreich Informatik, Universität Hamburg, Hamburg, Februar 1999

[Kossakowski 1999] Kossakowski, K.P.: Information Technology Incident Response Capabilities, Dissertation am Fachbereich Informatik, Universität Hamburg, Hamburg, 1999 (in Arbeit)

[Kruse 1995] Kruse, D.: Sicherheitszertifikat für Chipkarten, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, September 1995, S. 537 - 542

[Kruse, Peuckert 1995] Kruse, D., Peuckert, H.: Chipkarte und Sicherheit, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, März 1995

[Kuhn 1997] Kuhn, D.R.: Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role Based Access Control Systems, in: Proceedings of the 2. ACM Workshop on Role Based Access Control, ACM Press, New York, November 1997

[Kunze 1997] Kunze, M.: Kartentausch, in C′T Report 3, Geld online, Heise Verlag, 1997, S. 126 - 129

[Kuopus 1995] Kuopus, J.: Smart Cards and Data Protection, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, März 1995

[Langenscheidt 1997] Langenscheidts Handwörterbuch, Englisch-Deutsch, Deutsch-Englisch, 11. Auflage, Langenscheidt, Berlin, 1997

[Martin 1997] Martin, A.: Praxiseinsatz der Chipkarte im GeldKarte-System der deutschen Kreditwirtschaft, in: Fluhr, M. (Hrsg): Die Chipkarte: eine Welt der Möglichkeiten, Proceedings der OmniCard 1997, InTIME, Berlin, Januar 1997

[Meister, Glade 1995] Meister, G., Glade, A.: Sicherheitskonzepte und Schlüsselmanagement in Chipkartensystemen am Beispiel STARCOS, in: Bundesamt für Sicherheit in der Informationstechnik Fachvorträge: 4. Deutscher IT-Sicherheitskongreß, Bad Godesberg, Mai 1995

[Meyers 1992] Meyers Taschenlexikon, 4. Auflage, 24 Bände, BI Taschenbuchverlag, Mannheim, März 1992

[Milicent 1997] Digital Equipment Corporation: Information und Beschreibung des Millicent Verfahrens
http://www.millicent.digital.com/ (19.11.1998)

[Mondex 1996] Mondex International: Produktinformation zu Mondex
http://www.mondex.com/ (19.11.1998)

[Müller 1998] Müller, S.: E-Commerce und Cybercash: Voraussetzungen und Marktchancen für Handel und Banken, in: Fluhr. M. (Hrsg.): Die Chipkarte auf dem Weg zu Akzeptanz und Nutzung, Proceedings der OmniCard 1998, InTIME, Berlin, Januar 1998, S. 212 - 225

[NIST 1997] National Institute of Standards and Technology: Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES), Gaithersburg, September 1997
http://csrc.nist.gov/encryption/aes/aes_9709.htm
(04.02.1999)

[NIST 1999] National Institute of Standards and Technology: Advanced Encryption Standard (AES) Development Effort, Gaithersburg, Januar 1999
http://csrc.nist.gov/encryption/aes/aes_home.htm
(04.02.1999)

[Oberquelle 1987] Oberquelle, H.: Sprachkonzepte für benutzergerechte Systeme, Informatikfachberichte 144, Springer Verlag, Heidelberg, 1987

[Ott 1997] Ott, A.: Regelsatzbasierte Zugriffskontrolle nach dem „Generalized Framework for AccessControl“-Ansatz am Beispiel Linux, Diplomarbeit am Fachbereich Informatik, Universität Hamburg, Hamburg, November 1997

[Peuckert 1997] Peuckert, H.: Electronic Commerce und Sicherheit, 1997
http://www.bsi.bund.de/literat/tagungsb/peuckert.html
(20.11.1998)

[Pfitzmann, Pfitzmann, Waidner 1995] Pfitzmann, A., Pfitzmann, B., Waidner, M.: Vertrauenswürdiger Entwurf portabler Benutzerendgeräte und Sicherheitsmodule, in: Brüggenmann, H., Gerhardt-Häckl, W. (Hrsg.): Verläßliche IT-Systeme, Proceedings der GI-Fachtagung VIS 1995, Vieweg, 1995

[Pfitzmann, Waidner, Pfitzmann 1990] Pfitzmann, B., Waidner, M., Pfitzmann, A.: Rechtssicherheit trotz Anonymität in offenen digitalen Systemen, Datenschutz und Datensicherheit 5/1990 und 6/1990, Vieweg, Wiesbaden, 1990, S. 243 - 253 und S. 305 - 315

[Pfleeger 1997] Pfleeger, C.: Security in Computing, 2. Auflage, Prentice Hall, 1997

[Posch 1999] Posch, R.: Defintion von relativer kommerzieller Sicherheit, Persönlicher Kontakt, Januar 1999

[Posch, Bock, Mayerwieser, Posch 1998] Posch, K., Bock, H., Mayerwieser, W., Posch, R.: Wesentliche Kriterien beim Kryptochipkartenentwurf, Proceedings der Arbeitskonferenz Chipkarten, DuD Fachbeiträge, Vieweg, März 1998

[Rankl, Effing 1996] Rankl, W., Effing, W.: Handbuch der Chipkarten, Aufbau-Funktionsweise-Einsatz, 2. Auflage, Carl Hanser Verlag, München, 1996

[Rankl, Effing 1999] Rankl, W., Effing, W.: Chipkarten-Betriebssysteme, in: Card Forum, Das aktuelle Kartenmagazin, 6. Jahrgang, Ausgabe 1/1999, Every Card Verlags GmbH, Lüneburg, Januar 1999

[Rivest, Shamir, Adlemann 1978] Rivest, R.L., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, in: Communications of the ACM, Vol. 21 Nr. 2, ACM Press, Februar 1978, S. 120 - 126

[Roßnagel 1994] Roßnagel, A.: Smartcards - Gefährdung oder Instrument des Persönlichkeitsschutzes, in Datow, Kissinger, Lange (Hrsg.): Proceedings der MultiCard 1994, InTIME, Berlin, Januar 1994

[Roßnagel 1995] Roßnagel, A.: Datenschutz in Sicherungsinfrastrukturen offener Telekooperation, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, Oktober 1995, S. 582 - 587

[Ruland 1993] Ruland, C.: Informationssicherheit in Datennetzen, Datacom Verlag, Bergheim, 1993

[Sandhu, Coyne, Feinstein, Youman 1994] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role Based Access Control: A Multi-Dimensional View, in: Proceedings of the 11. Annual Computer Security Application Conference, ACSAC 1994, Orlando, FL, IEEE Computer Society Press, Los Alamitos, Dezember 1994

[Sandhu, Coyne, Feinstein, Youman 1996] Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role Based Access Control Models, in: IEEE Computer, Vol. 29, Nr. 2, Februar 1996

[Sandhu 1998] Sandhu, R.S.: Role Activation Hierarchies, in: Proceedings of the 3. ACM Workshop on Role-Based Access Control, ACM Press, New York, Oktober 1998

[Schier 1993a] Dippel, K.: Design for Security Functions of Chipcard Software, in: Proceedings of the IFIP WG 9.6 Working Conference, Stockholm - St. Petersburg, Elsevier Science Publishers, August 1993

[Schier 1993b] Dippel, K.: Benutzungsoberflächen von Sicherungsfunktionen in Chipkarten - Anforderungen und Gestaltungsvorschläge, Diplomarbeit am Fachbereich Informatik, Technische Hochschule Darmstadt, Darmstadt, Oktober 1993

[Schier 1996] Schier, K.: Sicherheitsaspekte des Einsatzes von Chipkarten in sensitiven Anwendungen, in: Proceedings der ENCRESS-Tagung 1996, Hamburg, April 1996

[Schier 1997] Schier, K.: Vergleich und Bewertung aktueller Systeme im elektronischen Zahlungsverkehr, in: Proceedings of the German Unix User Group Conference (GUUG), Wiesbaden, September 1997

[Schier 1998a] Schier, K.: Sicherheitsaspekte aktueller Systeme im elektronischen Zahlungsverkehr, in: Stork, B. (Hrsg.): Offene Systeme, Zeitschrift deutschsprachiger Unix-Benutzer-Vereinigungen, Band 7, Nr. 1, Springer Verlag, Heidelberg, Februar 1998, S. 19 - 26

[Schier 1998b] Schier, K.: Zahlungssysteme im Internet - Eine sicherheitstechnische Bewertung, in: Heinen, I. (Hrsg): Internet - von der Idee zum kommerziellen Einsatz, Deutscher Internet Kongreß, dpunkt.verlag, Frankfurt, Mai 1998, S. 223 - 233

[Schier 1998c] Schier, K.: A role and task based security model for multifunctional smartcard applications in the area of electronic commerce, in: Papp, G., Posch, R.: Global IT-Security, Proceedings of the IFIP TC-11 SEC 1998 Conference, Wien/Budapest, Kluwer, September 1998

[Schier 1998d] Schier, K.: Multifunctional smartcard applications for electronic commerce - An application of the role and task based security model, in: Proceedings of the 15. Annual Computer Security Application Conference, ACSAC 1998, Phoenix, AR, IEEE Computer Society Press, Los Alamitos, Dezember 1998

[Schier 1999a] Schier, K.: Der autonome Kunde im Vordergrund - Freie Wahl der Zahlungsmodalität bei multifunktionalen Chipkarten, in: Fluhr, M. (Hrsg.): Anwendungsfelder und Sicherheitskomponenten der Chipkarte, Proceedings der OmniCard 1999, InTIME, Berlin, Januar 1999

[Schier 1999b] Schier, K.: Der autonome Kunde im Vordergrund - Freie Wahl der Zahlungsmodalität bei multifunktionalen Chipkarten, in: Card Forum, Das aktuelle Kartenmagazin, 6. Jahrgang, Ausgabe 1/1999, Every Card Verlags GmbH, Lüneburg, Januar 1999

[Schier 1999c] Schier, K.: Sicherheit elektronischer Zahlungssysteme im Internet, in: Krallmann, H. Scholz-Reiter, B. (Hrsg.): Industrie Management - Electronic Commerce, 15. Jahrgang, Ausgabe 1/1999, GITO-Verlag für Industrielle Informationstechnik und Organisation GmbH, Berlin, Februar 1999

[Schier, Fischer-Hübner 1995] Schier, K., Fischer-Hübner, S.: Datenschutzprobleme beim Einsatz von Chipkarten im Gesundheitswesen, GMDS-Jahrestagung Abstractband, 1995

[Schier, Fischer-Hübner 1998] Schier, K., Fischer-Hübner, S.: The Global Information Society and Electronic Commerce: Privacy Threats and Privacy Technologies, in: Computers and Networks in the Age of Globalization, Proceedings of the IFIP 5. World Conference Human Choice and Computers, August 1998, S. 503 - 515

[Schneier 1996] Schneier, B.: Angewandte Kryptographie - Protokolle, Algorithmen und Sourcecode C, Addison-Wesley, Bonn, 1996

[Schuster, Färber, Eberl 1997] Schuster, R., Färber, J., Eberl, M.: Digital Cash - Zahlungssysteme im Internet, Springer Verlag, Berlin, 1997

[SET 1997a] Mastercard, Visa: SET Secure Electronic Transaction Specification: Book 1: Business Description, Mai 1997
`http://www.setco.com/download.html (20.11.1998)`

[SET 1997b] Mastercard, Visa: SET Secure Electronic Transaction Specification: Book 2: Programmer's Guide, Mai 1997
`http://www.setco.com/download.html (20.11.1998)`

[SET 1997c] Mastercard, Visa: SET Secure Electronic Transaction Specification: Book 3: Formal Protocol Definition, Mai 1997
`http://www.setco.com/download.html (20.11.1998)`

[SGZ 1998] SGZ-Bank: Beschreibung der SET-Einführung der SGZ-Bank
`http://www.set.sgz-bank.de (20.11.1998)`

[Siemens 1994] Siemens, Bereich Halbleiter: Marketing-Kommunikation: ICs for Chipcards SLE 44C200, Short Product Information, September 1994

[Siemens 1995a] Siemens, Bereich Halbleiter, Marketing-Kommunikation: ICs for Chipcards SLE 44C42, Short Product Information, März 1995

[Siemens 1995b] Siemens, Bereich Halbleiter, Marketing-Kommunikation: ICs for Chipcards SLE 44C80, Short Product Information, März 1995

[Siemens 1995c] Siemens, Bereich Automatisierungstechnik, Kombinationstechnik: CardOS User's Manual, Version 2.1, 1995

[STARCOS 1996] Giesecke & Devrient: STARCOS S 2.1, Version 2.1, Reference Manual, München, Oktober 1996

[Stark, Schmiede 1996] Stark, C., Schmiede, R.: Ist Bürgerbeteiligung bei der Gestaltung einer Patientenchipkarte wünschenswert und machbar? in: Datow, Kissinger, Lange (Hrsg.): Die Chipkarte im Alltag, Proceedings der MultiCard 1996, Januar 1996, S. 235 - 248

[Starke 1990] Starke, P.H.: Analyse von Petrinetzen, B.G. Teubner, Stuttgart, 1990

[Stewart 1998] Stewart, D.C.: The Future of Digital Cash on the Internet, 1998
`http://www.araydev.com/commerce/JIBC/9703-02.html (20.11.1998)`

[Struif 1992] Struif, B.: Das elektronische Rezept mit digitaler Unterschrift, in: Reimer H., Struif, B. (Hrsg.): Kommunikation und Sicherheit, Bad Vilbel, Darmstadt, 1992, S. 71ff

[Struif 1994] Struif, B.: Sicherheit und Datenschutz bei elektronischen Rezepten, in Datow, Kissinger, Lange (Hrsg.): Proceedings der MultiCard 1994, InTIME, Berlin, Februar 1994

[Sun 1997] Sun Microsystems: Java-Card 2.0, Application Programming Interface, Revision 1.0 Final, Palo Alto, Oktober 1997

[Sun 1998a] Sun Microsystems: The Gateway Security Model in the Java Electronic Commerce Framework, White Paper, Februar 1998

[Sun 1998b] Sun Microsystems: Java-Card 2.0, Applet Developers Guide, Revision 1.12, Palo Alto, August 1998

[Sun 1999] Sun Microsystems: Java-Card Version 2.1, Aktuelle Dokumentation, 1999
`http://java.sun.com/products/javacard/publicreview.html (21.01.1999)`

[Swiss NetPay 1999] Informationen über den Ecash-Pilotversuch in der Schweiz
`http://www.swissnetpay.ch/de/index.html (21.01.1999)`

[Tanenbaum 1995] Tanenbaum, A.S.: Moderne Betriebssysteme, Prentice Hall, London, 1995

[Thompson 1998] Thompson, B.: UniKart: Three Payment Options, One Card, in: Card Forum International, Mai/Juni 1998, Mai 1998, S. 41 - 44

[Tietze, Schenk 1993] Tietze, U., Schenk, C.: Halbleiterschaltungstechnik, 10. Auflage, Springer Verlag, Berlin, 1993

[W3C 1997] World Wide Web Consortium, Beschreibung des Projekts „P3“
`http://www.w3.org/P3/Overview.html (20.11.1998)`

[Wächter 1995] Wächter, M.: Prinzipien des Datenschutzes und der Datensicherung, Datenschutz und Datensicherheit, Vieweg, Wiesbaden, August 1995

[Weikmann 1997] Weikmann, F.: Chipkarten-Betriebssysteme, in: it + ti, Informationstechnik und Technische Informatik 5/1997, Mai 1997

[Yngström 1996] Yngström, L.: A Systemic Holistic Approach to Academic Programmes in IT Security, Dissertation am Fachbereich Informatik der Universität Stockholm, Akademitryck, Edsbruk, 1996

[Zimmermann 1995] Zimmermann, P.R.: The Official PGP User’s Guide, The MIT Press, Cambridge, MA, 1995

Weitere Literatur bezüglich Standards und Normen findet sich im Anhang Normen im Anschluß an dieses Literarurverzeichnis.

Normen

[DIN 44300] Deutsches Institut für Normung: DIN 44300, Informationsverarbeitung, Begriffe, 1998

[EMV 2.0] Europay, Mastercard, Visa: Integrated Circuit Card Specifications for Payment Systems, Version 2.0
Part 1: Electromechanical Characteristics, Logical Interface and Transmission Protocols, 1995
Part 2: Data Elements and Commands, 1995
Part 3: Transaction Processing, 1995

[EN 726] European Norm: Identification Card Systems - Telecommunications integrated circuit(s) card and terminal,
Part 1: System Overview, 1994
Part 2: Security Framework, 1995
Part 3: Application independent card requirements, 1994
Part 4: Application independent card realted terminal requirements, 1994
Part 5: Payment methods, 1995
Part 6: Telecommunication features, 1995
Part 7: Security Module, 1996

[FIPS 46] Department of Commerce: Data Encryption Standard (DES), FIPS Pub 46, Federal Information Processing Standards Publications, 1977

[FIPS 140-1] Department of Commerce: Security Requirements for cryptographic modules, FIPS Pub 140-1, Federal Information Processing Standards Publications, Januar 1994

[FIPS 186] Department of Commerce: Digital Signature Standard (DSS), FIPS Pub 186, Federal Information Processing Standards Publications, Mai 1994

[ISO 7810] International Standardisation Organisation: ISO/IEC 7810, 2. Auflage, Information Technology - Identification Cards - Physical Characteristics, 1995

[ISO 7811] International Standardisation Organisation: ISO/IEC 7811, 2. Auflage, Information Technology - Identification Cards - Recording Technique,
Part 1: Embossing, 1995
Part 2: Magnetic Stripe, 1995
Part 3: Location of embossed characters on ID-1 cards, 1995
Part 4: Location of read-only magnetic tracks - Tracks 1 and 2, 1995
Part 5: Location of read-write magnetic track - Track 3, 1995
Part 6: High coercivity magnetic stripe, 1995

[ISO 7812] International Standardisation Organisation: ISO/IEC 7812, Information Technology - Identification Cards,
Part 1: Numbering system, 1993
Part 2: Application and registration procedures, 1993

[ISO 7813] International Standardisation Organisation: ISO/IEC 7813, 3. Auflage, Information Technology - Identification Cards - Financial Transaction Cards, 1990

[ISO 7816] International Standardisation Organisation: ISO/IEC 7816, Information Technology - Identification Cards - Integrated Circuit(s) Cards with contacts,
Part 1: Physical Characteristics, 1987
Part 2: Dimensions and locations of the contacts, 1988
Part 3: Electronic signals and transmission protocols, 1989, 1992, 1994
Part 4: Interindustry commands for interchange, 1995
Part 5: Numbering system and registration procedure for application identifier, 1994, 1995
Part 6: Interindustry data elements, 1995
Part 7: Interindustry commands for structured card query language, (Draft)
Part 8: Security related interindustry commands, (Draft)
Part 9: Enhanced interindustry commands, (Draft)
Part 11: Security architecture, (Draft)

[ISO 8372] International Standardisation Organisation: ISO 8372, Modes of Operation for a 64-Bit Block Cipher Algorithm, 1987

[ISO 8824] International Standardisation Organisation: ISO/IEC 8824, Information Technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), 1990

[ISO 8825] International Standardisation Organisation: ISO/IEC 8825, Information Technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), 1990

[ISO 9564] International Standardisation Organisation: ISO 9564, Banking - Personal Identification Number management and security
Part 1: PIN protection principles and techniques, 1991
Part 2: Approved algorithm(s) for PIN encipherment, 1991

[ISO 9594-8]. International Standardisation Organisation: ISO/IEC 9594-8: Information Technology - Open systems Interconnection - The Directory: Authentication Framework, Technical Corrigendum 1, Amendment 1, 1997

[ISO 9798] International Standardisation Organisation: ISO 9798, Information Technology - Security Techniques - Entity Authentication
Part 1: General model, 1991
Part 2: Mechanisms using symmetric encipherment algorithms, 1994
Part 3: Entity authentication using a public key algorithm, 1993
Part 4: Mechanisms using a cryptographic check function, 1994

[ISO 9992] International Standardisation Organisation: ISO 9992, Financial Transaction Cards - Messages between the Integrated Circuit Card and the Card Accepting Device,
Part 1: Concepts and Structures, 1990
Part 2: Functions, Messages (commands and responses), data elements and structures, 1994

[ISO 10202] International Standardisation Organisation: ISO DIS 10202, Financial Transaction Cards - Security Architecture of financial Transaction Systems using Integrated Circuit Cards,
Part 1: Card life cycle, 1991
Part 2: Transaction Process, 1991
Part 3: Cryptographic key relationship, 1995
Part 4: Secure application modules, 1995
Part 5: Use of algorithms, 1995
Part 6: Card holder verification, 1994
Part 7: Key management, 1994
Part 8: General principles and overview, 1994

[ISO 10536] International Standardisation Organisation: ISO/IEC 10536, Identification Cards - Contacless integrated circuit(s) cards,
Part 1: Physical charcteristics, 1992
Part 2: Dimension and location of coupling areas, 1994
Part 3: Electronic signals and reset procedures, 1995
Part 4: Answer to reset and transmission protocols, 1995

[ISO 11693] International Standardisation Organisation: ISO/IEC 11693, Optical memory cards, 1995

[ISO 11694] International Standardisation Organisation: ISO/IEC 11694, Optical memory - Linear recording method cards,
Part 1: Physical charcteristics, 1995
Part 2: Dimension and location of the accessible optical areas, 1995
Part 3: Optical properties and characteristics, 1995

[ISO 14443] International Standardisation Organisation: ISO/IEC WD 14443, Remote coupling communication cards,
Part 1: Physical charcteristics, 1996
Part 2: Radio frequency interface, 1996
Part 3: Transmission protocols, 1996
Part 3: Transmission security features, 1996

Abkürzungen

A

Abs	Absatz (eines Gesetzes)
ABS	Acrylnitril Butadin Styrol
AC	Access Conditions
ACID	Atomocity, Consistency, Isolation, Durability
ACK	Acknowledge
ACL	Access Control Lists
ACM	Association for Computing Machinery
ACSAC	Annual Computer Security Applications Conference
Admin	Administrator
AES	Advanced Encryption Standard
AgV	Arbeitsgemeischaft der Verbraucherverbände
AID	Aplication Identifier
APDU	Application Protocoll Data Unit
API	Application Programming Interface
ASN	Abstract Syntax Notation
ATR	Answer to Reset

B

BdB	Bundesverband deutscher Banken
BIS	Bank for International Settlements
BLP	Bell-LaPadula
BLZ	Bankleitzahl
BMBF	Bundesministerium für Bildung und Forschung
BSDB	Bundesbeauftragter für den Datenschutz
BSI	Bundesamt für Sicherheit in der Informationstechnik
BVerfGE	Bundesverfassungsgesetz

C

CBC	Cipher Block Chaining
CCD	Charge Coupled Device
CDI	Constrained Data Items
CEPTF	Consumer Electronic Payment Task Force
CFB	Cipher Feedback
CISC	Complete Instruction Set Computer
CLA	Class
CLK	Clock
CPU	Central Processing Unit
CRL	Certificate Revocation List
CZ	Computer Zeitung

D

DAC	Discretionary Access Control
DASY	Data System
DEA	Data Encryption Algorithm
DEC	Digital Equipment Corporation
DES	Data Encryption Standard
DF	Dedicated File
DFC	Decorrelated Fast Cipher
DIN	Deutsches Institut für Normung (Deutsche Industrie Norm)
DIS	Draft International Standard
DM	Deutsche Mark
DoD	Department of Defense
DOS	Disk Operation System
DRAM	Dynamic Random Access Memory
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
DuD	Datenschutz und Datensicherheit

DVD	Deutsche Vereinigung für Datenschutz

E

E2	Efficient Encryption Algorithm
EC	Eurocheque
ECB	Electronic Code Book
EDC	Error Detection Code
EDV	Elektronische Datenverarbeitung
EEPROM	Electrical Erasable Programmable Read Only Memory
EF	Elementary File
EFTPOS	Electronic Fund Transfer at Point of Sale
E-Geld	Elektronisches Geld
EMV	Europay, Mastercard, Visa
EN	European Norm
ETH Zürich	Eidgenössische Technische Hochschule Zürich
ETU	Elementary Time Unit

F

FCC	Federal Communications Commissions
FID	File Identifier
FIPS	Federal Information Processing Standards
FRAM	Ferroelectric Random Access Memory
FuE	Forschung und Entwicklung

G

GB	Geldbörse
GFAC	Generalised Framework for Access Control
GI	Gesellschaft für Informatik
GMD	Gesellschaft für Mathematik und Datenverarbeitung
GND	Ground

GPN	Gefärbtes Petrinetz
GSM	Global System for Mobile Communications
GUUG	German Unix User Group

H

HmbDSB	Hamburger Datenschutzbeauftragter
HPC	Hasty Pudding Cypher
Hrsg	Herausgeber

I

I/O	Input/Output
IBI	Institut für Bankinformatik
IC	Integrated Circuit
ICC	Integrated Circuit Card
ID	Identification
IDEA	International Data Encryption Algorithm
IEC	International Electrotechnical Commission
IEEE	Institute of Electronical and Electronics Engeneers
IFD	Interface Device
IFIP	International Federation of Information Processing
INS	Instruction
ISF	Internal Secret File
ISO	International Organisation for Standardisation
IT	Information Technology
ITSEC	Information Technology Security Evaluation Criteria
IVP	Integritätsverifikations Prozedur

J

JCF	Java-Card Forum
JCVM	Java-Card Virtual Machine

JECF	Java Electronic Commerce Framework
	K
KK	Kreditkarte
KTO-Nr	Kontonummer
KWG	Kreditwesengesetz
	L
LB	Landesbank
LED	Light Emiting Diode
	M
MAC	Mandatory Access Control
MD	Message Digest
MF	Master File
MIT	Massachusetts Institute of Technologie
MLI	Multiple Laser Image
MLS	Multilevel Security
mm	Millimeter
mod	Modulus
µs	Mikrosekunde (10^{-6})
ms	Millisekunde (10^{-3})
	N
NAD	Node Adress
NCSC	National Computer Security Conference
NIST	National Institute of Standards and Technology
ns	Nanosekunde (10^{-9})
NSA	National Security Agecy

O

OFB	Output Feedback
OM	Operation Mode
OSI	Open Systems Interconnection

P

PC	Personal Computer
PC	Polycarbonat
PCB	Protocol Control Byte
PET	Poly Ethyl Enterephtalat
PGP	Pretty Good Privacy
PIN	Personal Identification Number
PIX	Proprietary Application Identifier Extension
POS	Point of Sale
prEN	Pre European Norm
prETS	Pre European Telecommunication Standard
PTS	Protocol Type Select
PTT	Public Telephone Company
Pub	Publication
PVC	Poly Vinyl Chlorid

Q

q.e.d.	quod erat demonstrandum

R

R&A-Modell	Rollen- und Aufgabenbasiertes Modell
RAM	Random Access Memory
RBAC	Role Based Access Control
RC6	Rivest Cyper 6
RFU	Reserved for Future Use

RID	Registered Identifier
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RSA	Rivest, Shamir, Adleman (Verschlüsselungs-Algorithmus)
RST	Reset

S

SAFER+	Secure And Fast Encryption Routine
SB	Selbstbedienung
SCHUFA	Schutzgemeinschaft für allgemeine Kreditsicherung
SET	Secure Electronic Transactions
SGZ-Bank	Südwestdeutsche Genossenschafts-Zentralbank
SHA	Secure Hash Algorithmus
SIM - ME	Subscriber Identity Module - Mobile Equipment
SRAM	Static Random Access Memory
SS	Simple Security
STARCOS	Smart Card Chip Operating System
SWIFT	Society for Worldwide Interbank Financial Telecommuniactions

T

TAN	Transaktionsnummer
TC	Trusted Computer
TCSEC	Trusted Computer Systems Evaluation Criteria
TKT	Ticket
TLV	Tag, Length, Value
TP	Transformations Prozedur
TPDU	Transmission Protocol Data Unit
TV	Television

U

UDI	Unconstrained Data Items
ULPAA	Upper Layer Protocols Architectures and Applications
USA (US)	United States of Amerika

V

V	Volt
Vcc	Versorgungsspannung (Common Voltage)
VIS	Verläßliche IT-Systeme
Vol	Volume
Vpp	Programmierspannung (Programming Voltage)

W

W3C	World Wide Web Consortium
WEF	Working Elementary Files
WG	Working Group
WO	Write Once
WR	Write

X

XOR	Exklusiv Oder (logische Verknüpfung)