Hochschule für Technik, Wirtschaft und Kultur Leipzig (FH)

Fachbereich Informatik, Mathematik und Naturwissenschaften
Studiengang Informatik

Diplomarbeit

Thema:

GeldKartenzahlung im Internet

-

Entwicklung einer GeldKartenschnittstelle

Verfasser: Matthias Kost 96 I

Betreuer: Prof. Dr. - Ing. K. Bastian

Hinweise zur Diplomarbeit

Typografische Besonderheiten: Wörtliche Zitate, sowie Dokumentnamen im Literaturverzeichnis werden *„kursiv und in Anführungszeichen“* dargestellt.
Programmteile erscheinen im `Schreibmaschinenstyle` .
Wichtige Teile sind **fett** hervorgehoben.

Fachbegriffe und Abkürzungen: werden im Glossar erklärt.

CD: Die Diplomarbeit enthält eine CD mit weiterführenden Informationen und Spezifikationen.

Das Literaturverzeichnis: befindet sich zusätzlich als HTML-Datei auf der beiligenden CD. Auf dieser Webseite sind die Verzeichniseinträge verlinkt, wenn es sich um eine Referenz auf Internetdokumente handelt. Die referenzierten Webseiten wurden zusätzlich auf CD gespeichert und verlinkt, da sich Internetadressen ändern können.

Der Quellkode der GeldKartenschnittstelle: ist auch auf der beiligenden CD zu finden.

Ehrenwörtliche Erklärung

Ich erkläre hiermit, dass ich die vorliegende Diplomarbeit selbständig angefertigt, noch nicht anderweitig für Prüfungszwecke vorgelegt, sowie keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Schkeuditz, den 21. August 2001

Matthias Kost

Inhaltsverzeichnis

Kapitel 1

Einleitung und Motivation

Die Hochschule für Technik, Wirtschaft und Kultur (HTWK) Leipzig leistet Mitarbeit an dem Projekt „eVerlage“ (http://www.everlage.de). In diesem Projekt wurde eine elektronische Bibliothek aufgebaut, auf deren Dokumente über das Internet zugegriffen werden kann.
Zur Zeit (Stand April 2001) existieren etwa 100 wissenschaftliche Publikationen in dieser Bibliothek. Diese stehen meist als HTML- und/oder PDF-Version zur Verfügung. Die Angebotsform liegt dabei in Händen des Verlages, welcher das Dokument für eVerlage zur Verfügung gestellt hat.
Die Dokumente sind primär zur Ansicht auf dem Rechner gedacht. Nur so kommt auch der Mehrwert von elektronischen Dokumenten, wie Volltextsuche, Verlinkung und Verwendung von multimedialen Elementen, zur Geltung.
Ziel der dokumentanbietenden Verlage ist es natürlich, mit den elektronischen Büchern Gewinn zu erwirtschaften. Die HTWK Leipzig untersucht deshalb für das eVerlage-Projekt, mit welchen Mechanismen und Bezahlmethoden dies möglich ist und integriert lohnenswerte Verfahren in das Projekt.
Zur Zeit wird dem Nutzer ein pay-per-view angeboten. Dies bedeutet, dass man als Kunde die Ansicht eines Dokumentes für einen bestimmten Zeitraum bezahlen muss. Dabei stellt ein Dokument allerdings meist kein komplettes Printmedium dar, sondern durch die elektronische Verfügbarkeit ist es möglich, auch nur Teile eines Buches (z.B. einzelne Kapitel) oder einer Zeitschrift (z.B. einen Artikel) zur Ansicht zu erwerben.
Dementsprechend niedrig können auch die Preise sein. Die elektronischen Bücher oder Zeitschriften sind an sich schon billiger als die Printversionen, da keine

Druckkosten entstehen und keine Waren transportiert werden müssen. Wenn ein Kunde ausserdem nur die ihn interessierenden Kapitel oder Artikel kauft, dann liegt der Kaufpreis meist unter 5 DM.
Es ist also notwendig, dass sogenannte Micropayment-Bezahlverfahren zur Verfügung stehen, welche Aufgrund geringer Transaktionskosten auch bei Kleinstbeträgen für den Händler gewinnbringend sind.
Ich finde das Thema Micropayment im Internet sehr interessant, da ich der Meinung bin, dass mit Hilfe solcher Zahlungssysteme vielfältige neue Anwendungen möglich sind.
Meine Diplomarbeit wird sich deshalb in Kapitel 2 mit der Problematik Zahlungsmethoden im Internet auseinandersetzen und dabei in Kapitel 3 das Bezahlverfahren „GeldKarte“ näher beleuchten. Kapitel 4 wird sich mit dem GeldKartechip auseinandersetzen. Im fünften Kapitel wird eine Softwareschnittstelle für die GeldKartenzahlung vorgestellt, bevor im sechsten und letzten Kapitel ein Fazit gezogen werden soll.

Kapitel 2

Zahlungssysteme im Internet und deren mögliche Verwendung im eVerlage Projekt

In diesem Kapitel werden zunächst die wichtigsten, für das Bezahlen im Internet geeigneten, Zahlungsverfahren vorgestellt. Dadurch soll der Leser einen Überblick über die Problematik des internetbasierten Zahlungsverkehrs bekommen, sowie die zur Zeit verfügbaren Lösungsansätze kennenlernen. Die Zahlungsverfahren werden nach bestimmten Kriterien bewertet und auf Einsatzmöglichkeiten im eVerlage-System überprüft.

2.1 Anforderungen an Zahlungssysteme

Zur Zeit existieren relativ viele Zahlungsverfahren mit unterschiedlichen Eigenschaften auf dem Markt. Es kommen ständig neue Systeme dazu, während andere wieder verschwinden, weil sie sich nicht durchsetzen konnten.
Aufgabe der HTWK Leipzig ist es, diese Systeme zu untersuchen und für eVerlage nutzbare Zahlungsverfahren zu implementieren. Da eVerlage als Händler auf Grund der Art der Waren und deren Vertrieb bestimmte Anforderungen an die Verfahren stellt, sind diese nicht alle geeignet.

Auch die Kunden, welche die elektronischen Dokumente zur Ansicht kaufen, stellen an das zu verwendente Zahlungsverfahren gewisse Anforderungen, auf die hier mit eingegangen werden soll. Schliesslich setzt sich ein Zahlungsverfahren nur durch, wenn es von allen beteiligten Parteien akzeptiert wird.

Als erstes muss man beachten, dass es zwei verschiedene Methoden gibt, um auf das eVerlage System zuzugreifen. Als Kunde hat man die Möglichkeit sich registrieren zu lassen oder anonym als Gastnutzer aufzutreten.

Als **registrierter Kunde** besitzt man bei eVerlage ein Konto. Dieses muss man mit einem der angebotenen Bezahlverfahren um einen gewünschten Betrag aufladen. Danach kann man auf die kostenpflichtigen Dokumente zugreifen, wobei der für die Ansicht zu zahlende Betrag von dem aufgeladenen Konto abgebucht wird. Falls der Restbetrag für die Ansicht eines Dokumentes nicht mehr ausreicht, muss das Konto erst erneut aufgeladen werden.
Ein registrierter Kunde zahlt also per Vorkasse, d.h. er zahlt erst Geld auf sein eVerlage-Konto ein und verwendet dieses später für den Einkauf. Als Grundlage eines solchen Prepaidverfahrens muss daher ein grosses Vertrauensverhältnis des Kunden zum Händler bestehen, da das Risiko nur beim Kunden liegt. Für den Händler hingegen hat diese Methode den Vorteil, dass die Bezahlung der Ware bereits erfolgte bzw. gesichert ist und somit keine Probleme mit zahlungsunfähigen oder -unwilligen Kunden entstehen.
Der Kunde ist dafür sehr mobil, da er sich nur gegenüber dem Händler authentifizieren muss, um bequem von seinem Kundenkonto bezahlen zu können. Auch hier muss der Kunde dem Händler vertrauen, dass das Kundenkonto ordnungsgemäss geführt wird. Einen verbindlichen Nachweis über den Kontostand und die geführten Transaktionen hat der Kunde nicht, da der Händler kein Kreditinstitut ist.
Solch ein Konto besitzt allerdings auch die Vorteile, dass jeder gewünschte Betrag ohne weitere Transaktionskosten bezahlt werden kann und der eigentliche Bezahlvorgang nach Aufladung des Kontos sehr einfach ist.
Da man bei der Registrierung nur eine e-mail-Adresse zwingend angeben muss und sich Benutzernamen und Passwort aussuchen kann, ist es möglich, relativ anonym aufzutreten. eVerlage kann zwar nachverfolgen, auf welche Dokumente der Nutzer zugegriffen hat, allerdings besitzt eVerlage, besonders bei Verwendung von anonymen e-mail-Adressen, keinerlei private Information über den Kunden. Dies ist natürlich nur bei anonymen Zahlungsverfahren möglich, bei denen eVerlage keine weiteren Daten vom Kunden benötigt.

Die zweite Möglichkeit eVerlage zu verwenden, besteht darin, sich als **Gastnutzer** anzumelden. Bei solch einem anonymen Gastnutzer existieren auf eVerlage- und Kundenseite andere **Anforderungen an Zahlungssysteme**. Diese sind:

1. **Anonymität**:
   Viele Nutzer bevorzugen es anonym einzukaufen, um nicht zum gläsernen Kunder zu werden und um einer Werbeflut zu entgehen. Anonymität ist einer der Gründe, warum das Internet so schnell von den Menschen akzeptiert wurde. Es sollte daher die Möglichkeit bestehen, das kostenpflichtige Angebot der digitalen Bibliothek zu nutzen und trotzdem gegenüber dem Händler eVerlage anonym zu bleiben.
   Falls keine vollständige Anonymität vorliegt, so ist es für viele Nutzer wichtig, dass sie so wenig wie möglich von sich preisgeben müssen.
   Für Händler ist diese Anonymität zwar eher negativ, da sie einiges an Kundenarbeit (gezielte Werbung, Profile) unmöglich macht. Allerdings muss der Händler dieses Manko in Kauf nehmen, wenn er Kunden gewinnen und behalten will.

2. **Micropaymentfähigkeit**:
   Da die abzubuchenden Beträge sehr gering sind (z.B. 1 DM), muss das Zahlungssystem sich für die Transaktion von Kleinstbeträgen eignen.
   Eine Voraussetzung dafür sind geringe Transaktions- und Nebenkosten auf Händler- sowie Kundenseite. Möglichst kleine Kosten für Transaktionen werden natürlich bei allen Zahlungssystemen angestrebt. Doch besonders bei Micropaymentzahlungssystemen müssen diese im Pfennigbereich oder sogar darunter liegen, um nicht unproportional gross zum Zahlungsbetrag zu sein, oder diesen gar zu übersteigen.

3. **Zahlungsbestätigung**:
   Die Dokumente werden erst ausgeliefert, wenn der eVerlage-Server eine Zahlungsbestätigung bekommen hat (Warenlieferung nach Zahlung). Da ein Kunde allerdings sofort nach dem Bezahlvorgang auf das gewünschte Dokument zugreifen will, muss die Zeitspanne zwischen Kauf und Auslieferung der Ware (Dokument ist zur Ansicht freigegeben) möglichst kurz oder gar unmerklich sein, also ein Echtzeit-Clearing vorliegen.
   Deshalb muss das Zahlungssystem den Händlerserver sofort über das

Ergebnis der Zahlung informieren, damit der Kunde das erworbene Dokument oder eine Fehlermeldung bei Fehlzahlung erhält.

4. **Sicherheit**:
   Natürlich benötigt ein Zahlungssystem gewisse Sicherheitsmechanismen, um sichere Zahlungstransaktionen zu gewährleisten.
   Wenn bei der Zahlung öffentliche Netze wie das Internet genutzt werden, dann müssen die Daten so verschlüsselt sein, dass sie ein Dritter nicht ohne weiteres entschlüsseln bzw. nicht weiter verwenden. Ausserdem müssen die Daten manipulationssicher sein, ein Dritter darf sie also nicht verändern können.
   Desweiteren sollte das Transaktionsschema so aufgebaut sein, dass bei Fehlzahlungen, Betrug und ähnlichen Vorfällen weder für die Händler- noch für die Kundenseite ein Risiko besteht, unrechtmässig Geld zu verlieren.
   Leider ist eine sehr hohe Sicherheit oftmals nur mittels recht komplizierter Verfahren realisierbar.

5. **Mobilität**:
   Ziel des eVerlage Projektes ist es Informationen jederzeit global über das Internet anzubieten, also ein „information on demand“ zur Verfügung zu stellen. So mobil wie der Kunde sein kann, so mobil sollte auch das Zahlungssystem sein. Schliesslich will der Nutzer von jedem Internet-Terminal auf das kostenpflichtige Angebot zugreifen können. Eine Abhängigkeit von nicht transportabler Hardware, fest installierter Software oder Schlüssel ist daher eher unerwünscht.

Daneben existieren noch **wünschenswerte Eigenschaften**, die ein Zahlungssystem mitbringen sollte:

1. **geringe Grundkosten**:
   Bei vielen Zahlungssystemen fallen neben den reinen Transaktionsgebühren auch Grundkosten an, z.B. der Jahresbeitrag eines Kunden bei einem Kreditinstitut oder eine monatliche Gebühr, die ein Händler an einen Zahlungsprovider bezahlen muss. Diese Kosten sollten möglichst gering sein, damit ein Zahlungssystem für Kunden und Händler ansprechend ist. Wenn z.B. die Einnahmen eines Händlers mittels eines bestimmten Zahlungssystemes geringer sind als die Grundkosten für dieses System, dann wird es der Händler aus wirtschaflichen Gründen nicht mehr anbieten.

2. **einfacher Einstieg**:
   Der Einstieg in ein Zahlungssystem sollte möglichst einfach erfolgen. Kauf von Zusatzhardware oder eine Anmeldung bei speziellen Zahlungssystemprovidern schreckt viele Kunden vorerst ab. Auch Händler werden kaum Zahlungssysteme einsetzen, deren Integration sehr kompliziert und teuer ist.

3. **einfache Bedienung**:
   Wenn der Einstieg geschafft ist, dann sollten die Zahlungsabwicklungen recht einfach, komfortabel und schnell erfolgen.
   Leider geht mit einer einfachen Bedienung oftmals eine geringere Sicherheit einher. Zahlungssystemanbieter müssen deshalb meist einen Mittelweg zwischen einfacher Bedienung und hoher Sicherheit wählen.
   Bei der Händlerseite wird im Hinblick auf einfache Bedienung eine Vollautomatisierung mit möglichst geringem Administrationsaufwand angestrebt.

4. **Abdeckung eines grossen Zahlungsbereiches**:
   Je grösser die Betragsspanne eines Zahlungssystems ist, um so vielfältiger kann es eingesetzt werden. Als Kunde will man mit einem Zahlungssystem möglichst viel bezahlen können, von 10 Pfennigen für eine Information bis zu grösseren Beträgen wie z.B. 2000 DM für einen Computer.

5. **grosse Akzeptanz**:
   Ein Zahlungssystem sollte von möglichst vielen Händlern und Dienstleistern unterstützt werden, da sich ein solches System nicht auf breiter Basis durchsetzen wird, wenn man es als Kunde nur innerhalb einiger eng begrenzter Anwendungen nutzen kann. Jedes neue Zahlungssystem hat mit dem „Henne-Ei-Problem“ zu kämpfen, da ein System ohne Kunden von kaum einem Händler angeboten wird, und andererseits Kunden keine Lösungen nutzen, mit denen sie nur sehr begrenzt einkaufen können.

6. **Transparenz**:
   Für Kunden und Händler muss der Zahlungsablauf transparent sein. Als Kunde will man wissen, wieviel man bezahlt, an wen man bezahlt und ob die Zahlung erfolgreich war. Fehlermeldungen sollten detailliert, aber nicht zu technisch sein. Meist ist eine private Speicherung der Transaktionsdaten erwünscht, um auch später die Zahlungen verifizieren zu können.

7. **Stornierungsmöglichkeit**:
   Wenn ein Kunde materielle Ware bestellt, dann wird oft gefordert, dass die Bestellung stornierbar ist. Im Normalfall werden Stornierungen allerdings nicht mit dem Bezahlsystem durchgeführt, sondern per Überweisung, Gutschrift etc. geregelt.
   Da eVerlage ein pay-per-view anbietet, ist eine Stornierung schwer oder unmöglich, da der Kunde die Ware ja bereits erhalten hat. Theoretisch könnte man aber zum Beispiel die Zugriffsdauer auf ein Dokument verkürzen. Wenn ein Kunde also eine Jahreslizenz erworben hat, und ihm das Dokument nach einem Tag nicht gefällt, so könnte man die Jahreslizenz in eine Tageslizenz umwandeln und dem Kunden den Differenzbetrag gutschreiben. Dies wäre allerdings nur bei registrierten Kunden mit einem festen eVerlage-Konto möglich. Bei einem Gastnutzer besteht dagegen keine Stornierungsmöglichkeit.

Diese Zusatzeigenschaften sind zwar nicht essentiell, aber oftmals dafür verantwortlich, ob ein Zahlungssystem akzeptiert und genutzt wird oder nicht.
„Bei der Diskussion um neue, elektronische Zahlungssysteme geht es also erstens nicht um die conditio sine qua non des Internet-Handels, sondern um Kriterien und Ansprüche, denen solche Zahlungssysteme genügen sollen: Sicherheitsstandards, Grad der Integration ins bestehende Bankensystem, Fairness der Verfahren, Bargeldnähe, Anonymität, Kontounabhängigkeit etc. Anders formuliert: Zahlungssysteme für den Internet-Handel werden nur bis zu einem gewissen Grad durch Sachzwänge determiniert. In einem nicht zu unterschätzenden Maße nehmen sie vermittelt über Interessen und Wertvorstellungen der Akteure [Kunden, Händler, beteiligte Banken/Paymentprovider] - *und das schließt Konflikte ein - Gestalt an.“* [1]

Es ist sehr schwer zu prognostizieren, wie sich der Zahlungssystemmarkt insgesamt entwickeln wird, da sich auch die Anforderungen von Händlern und Kunden sowie der Wissensstand der Nutzer ständig verändern. Es ist daher für Händler und Zahlungssystemanbieter notwendig zu wissen, was die Kunden wünschen und was sie ablehnen. Aus diesem Grund ist es auch für einen Händler wie eVerlage wichtig, aktuelle Studien zum Kundenverhalten zu konsultieren. Eine gute Quelle für Informationen ist die jährliche Studie „Internet-Zahlungssysteme aus Sicht der Verbraucher“ der Universität Karlsruhe. Die zur Zeit aktuelle Version kann man unter http://iww.uni-karlsruhe.de/IZV4 bestellen.

Desweiteren exisitiert das Projekt „Technikfolgenabschätzung zu Elektronischen Zahlungssystemen für digitale Produkte und Dienstleistungen im Internet“ (PEZ), das von ITAS (Institut für Technikfolgenabschätzung und Systemanalyse, http://www.itas.fzk.de), gestützt vom BMBF, durchgeführt wird.

2.2 Existierende Zahlungssysteme für das Internet

In diesem Unterkapitel werden die wichtigsten, zur Zeit (Stand Sommer 2001) verfügbaren Zahlungssysteme für das Internet vorgestellt, bewertet und auf Brauchbarkeit für eVerlage untersucht. Dabei werden die gerade dargestellten Anforderungen und Wünsche in tabellarischer Form zusammengefasst. Da sich einige Verfahren ähnlich sind und über andere keine weiteren Informationen vorlagen, werden allerdings nicht alle tabellarisch aufbereitet. Die wünschenswerte Eigenschaft der Stornierung wird nicht mit dargestellt, da sie für ein pay-per-view System wie eVerlage irrelevant ist, weil der Kunde die Ware bereits erhalten hat. Die Grundkosten werden für Händler und Kunde getrennt betrachtet.
Da Zahlungssysteme zum Teil länderspezifisch sind, werden hier nur die Lösungen betrachtet die bereits in Deutschland exisitieren oder bald verfügbar sein werden. Weil sich der Zahlungssystemmarkt im Internet erst entwickelt, entstehen ständig neue Verfahren, während bereits existierende Systeme aus diversen Gründen eingestellt werden. Trotzdem haben sich bereits verschiedene Lösungsansätze für Zahlungsverfahren im Internet herauskristallisiert. Ich werde die Verfahren nach der verwendeten technischen Grundlage (z.B. auf Mobiltelefonbasis) ordnen, da so die Gruppeneigenschaften am besten ersichtlich werden. Auch eine Einschätzung der Verwendungsmöglichkeit der Systeme für eVerlage gestaltet sich so übersichtlicher. Natürlich gibt es noch eine Reihe weiterer möglicher Anordnungskriterien wie z.B. die Unterteilung in Micro- und Macropaymentverfahren.
Die Grundidee für dieses Unterkapitel entstand beim Betrachten der Zahlungssystemübersicht [2] auf http://www.electronic-commerce.org. Damals wurden die Bezahlverfahren aber noch recht kurz und knapp abgehandelt und nicht weiter bewertet. Auch der Kostenaspekt wurde nicht behandelt. Deshalb hatte ich den Beschluss gefasst, die wichtigsten verfügbaren Zahlungssysteme näher zu betrachten und auf Eignung für eVerlage zu untersuchen. Inzwischen gibt es

jedoch auch recht gute Artikel im Internet, sowie Bücher zum Thema Zahlungsverfahren. Aus diesem Grund, und um den Rahmen dieser Diplomarbeit nicht zu sprengen, werden einige Verfahren in diesem Kapitel nur erwähnt. Eine genauere Beschreibung kann der interessierte Leser im Anhang zu finden.
Weil der Zahlungssystemmarkt sehr dynamisch und schnelllebig ist, wurden die Informationen hauptsächlich aus dem Internet sowie aktuellen Fachzeitschriften entnommen oder per e-mail angefordert.

Bevor die Verfahren vorgestellt werden, soll die folgende Tabelle 2.1 einen Überblick über die wichtigsten Zahlungssysteme des Internets verschaffen. (A) bedeutet dabei, dass die Beschreibung dieses Verfahrens im Anhang zu finden ist.

Typ	Zahlungsverfahren
klassisch	Nachnahme Rechnung
Kontobasis	Lastschrift/Bankeinzug Net900 Kontopass (A) Online Überweisung
Kreditkarte	Kreditkarte eops-Cards (A) Secure Electronic Transfer (SET) CashRegister (A) aposto (A)
Mobiltelefon	paybox Street Cash eops-Mobile (A) eops-PIN (A) moneybox (A) Payitmobile (A)
Chipkarte	GeldKarte mondex (A)
Inkasso	click&buy
Telefonrechnung	Net900 classic eops-Call (A) eops-Connector (A) PurePay (A) Weitere Anbieter (A)
Prepaid	Paysafecard
Elektronisches Geld	eCash CyberCoin millicent
Elektronische Schecks	NetCheque
Paymanagement	e-Tra Wirecard Isoft powercash21 weitere Anbieter

Tabelle 2.1: Zahlungssysteme im Internet

2.2.1 Klassische Zahlungsverfahren

Klassische Zahlungsverfahren sind schon lange in Deutschland etabliert. Sie haben ihre Wurzeln in der Bezahlung von Leistungen oder physischen Waren. Der Kunde zahlt meist von seinem Konto oder in bar. Durch neue elektronische Bezahlverfahren werden die herkömmlichen Methoden zwar etwas zurückgedrängt, aber auf absehbare Zeit trotzdem nicht verschwinden.

Nachnahme:
Diese Bezahlmethode wird oftmals bei der Bestellung von physischen Waren genutzt. Der Kunde bezahlt in bar (bis 3000 DM) oder mit garantiertem ec-Scheck (bis 400 DM pro Scheck), wenn er die Ware vom Bringdienst entgegennimmt. Dies hat den Vorteil, dass man wie in einem Laden Geldwert gegen Warenwert tauscht. Leider ist die Ware meist verpackt, so dass es schwer ist, sich vom korrekten und funktionsfähigen Inhalt zu überzeugen. Bei der Deutschen Post beträgt die Transaktionsgebühr 6.50 DM. (alle Preise aus Preisliste der Deutschen Post, Stand 2001)

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Nein	Name und Lieferadresse sind dem Händler bekannt
Micropayment möglich	Nein	Transaktionsgebühr zu hoch
Echtzeit-Clearing	Nein	Kunde kann Ware ablehen
Sicherheit hoch	Ja	Abhängig vom Bringdienst
Mobilität hoch	Ja	
Kosten Händler gering	Ja	Nur Transaktionsgebühr
Kosten Kunde gering	Ja	Keine
Einstieg einfach	Ja	
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	ca. 10 bis 3000 DM
Akzeptanz hoch	Ja	Jeder erwachsene Mensch kann Händler oder Kunde sein
Tranparenz hoch	Ja	

Tabelle 2.2: Bewertung Nachnahme

Eignung für eVerlage: Nachnahme ist nur eingeschränkt für eVerlage verwendbar, da keine Anonymität, keine Micropaymentfähigkeit und kein Echtzeit-Clearing vorliegen. Man könnte dieses Zahlverfahren maximal dazu einsetzen, das Kundenkonto eines registrierten Kunde aufzuladen, indem dieser einen Brief von eVerlage mittels Nachnahme bezahlt.

Rechnung:
Bei einer Rechnung bekommt der Kunde ein Schreiben, in welchem neben dem zu zahlenden Geldwert die Bankverbindung des Händlers angegeben ist. Er muss innerhalb einer bestimmten Frist diesen Geldwert auf das Konto des Händlers überweisen oder bar schicken.
Diese Bezahlmethode kommt oft bei der Bezahlung von Waren sowie Dienstleistungen zum Einsatz. Transaktionsgebühren entstehen indirekt durch Überweisungskosten oder Porto für Kunden, und eventuell durch Portokosten für den Händler bei Rechnungsstellung. Lohnende Zahlung sind daher erst ab 5 DM realisierbar [3].
Es exisitieren auch Clearing-Firmen wie iclear (http://www.iclear.de), welche die Rechnungsstellung komplett übernehmen.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Nein	Name,Anschrift und evtl. Kontoverbindung sind bekannt
Micropayment möglich	Nein	Transaktionskosten zu hoch
Echtzeit-Clearing	Nein	Kunde hat Zahlungsfrist
Sicherheit hoch	Ja	Aber nicht zahlende Kunden können ein Problem sein
Mobilität hoch	Ja	
Kosten Händler gering	Ja	Evtl. Portokosten
Kosten Kunde gering	Ja	Porto- oder Überweisungskosten
Einstieg einfach	Ja	
Bedienung einfach	Ja	Aber etwas verwaltungsaufwendig
Zahlungsbereich gross	Ja	ab ca. 5 DM,nach oben offen
Akzeptanz hoch	Ja	Jeder erwachsene Mensch kann Händler oder Kunde sein
Tranparenz hoch	Ja	

Tabelle 2.3: Bewertung Rechnung

Eignung für eVerlage: Zahlung per Rechnung besitzt die selben gravierenden Nachteile wie Nachnahme und kann deshalb auch nur zur Aufladung des Kundenkontos eines registrierten Nutzers verwendet werden.

2.2.2 Zahlung auf Kontobasis

Für die folgenden drei Zahlungsverfahren benötigen Kunde und Händler jeweils ein Konto. Welche Bank das Konto führt, bleibt jeder Partei selbst überlassen. Das Geld wird bei einem Zahlvorgang direkt über die Banknetze von dem Kundenkonto auf das Konto des Händlers transferiert.

Bankeinzug/Lastschrift:
Ein Bankeinzug ermöglicht es einem Händler, den fälligen Betrag direkt vom Konto des Kunden abzubuchen. Dies wird oft bei sich wiederholenden Zahlungen wie Mieten, Zeitungsabonnements oder Mitgliedsbeiträgen genutzt. Der Händler benötigt für den Einzug die Kontoverbindung des Kunden.
An Kosten entsteht für Kunden gegebenenfalls eine Buchungsgebühr. Der Händler muss eine einmalige Zahlung von mindestens 1200 DM, sowie variable monatliche Grundgebühren einkalkulieren.
Die Buchungsgebühr beträgt ca. 0,05 DM zzgl. einer Überprüfungsgebühr zw. 0,13 und 1,35 DM (Kosten aus [3]). Bei nicht gedecktem Kundenkonto können zusätzliche Verwaltungskosten entstehen.
Im Anhang wird das elektronische Lastschriftverfahren „eops-Transactions“ kurz vorgestellt.

Eignung für eVerlage: Das Lastschriftverfahren ist sehr gut für registrierte Kunden geeignet. Diese können beliebig viele Dokumente lesen, und eVerlage zieht die dafür anfallenden Kosten ohne weiteren Aufwand für den Kunden selbstständig ein. Allerdings muss auch beachtet werden, dass viele Menschen sehr vorsichtig bei der Genehmigung zum Lastschriftverfahren sind.
Begrenzt ist die Bezahlmethode auch für nicht registrierte Nutzer verwendbar. Diese sind zwar nicht anonym, müssen aber nur Namen und Bankverbindung angeben. Auf Grund der Transaktionskosten lohnt allerdings erst der Kauf von etwas höherpreisigen Dokumenenten (ab 2 DM). Bei nicht registrierten Kunden sollten elektronische Lastschriftverfahren wie eops-Transactions verwendet werden, um das Echtzeit-Clearing zu garantieren.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Nein	Name und Kontoverbindung sind bekannt
Micropayment möglich	Kaum	Erst ab ca. 2 DM sinnvoll
Echtzeit-Clearing	Ja	Bei elektronischen Lastschriften
Sicherheit hoch	Ja	
Mobilität hoch	Ja	
Kosten Händler gering	Nein	
Kosten Kunde gering	Ja	Nur Kontoführungskosten
Einstieg einfach	Ja	
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	2 DM bis bankabhängige Grenze
Akzeptanz hoch	Ja	Jeder erwachsene Mensch kann Händler oder Kunde sein
Tranparenz hoch	Ja	Aber ständige Kontrolle durch Kunden notwendig

Tabelle 2.4: Bewertung Bankeinzug/Lastschrift

Online-Überweisung:
Die Firma fun communications bietet ein Zahlungsmodul namens „fun HomePay“ an. Damit können Online-Überweisungen vorgenommen werden. Somit schlägt dieses Verfahren eine Brücke zwischen Homebanking und Internetpayment. Der Kunde benötigt deshalb auch eine Homebanking-PIN und eine Transaktionsnummer (TAN) zum Bestätigen der Überweisung.
Das Clearing geschieht in Echtzeit. Leider ist das Verfahren laut fun communications nur für Macropayment geeignet und könnte somit wieder nur als weitere Aufladem̈oglichkeit eines eVerlage-Kontos angeboten werden. (Informationen aus Prospekt von fun communications)

2.2.3 Zahlung per Kreditkarte

Kreditkarte:
Bei der Zahlung mit der Kreditkarte werden über einen normalerweise abgesicherten Internetkanal (SSL, TSL) die Kreditkartendaten wie Name und Kreditkar-

tennummer des Kunden dem Händler übermittelt. Dieser lässt mit Hilfe dieser Daten das Geld vom Konto des Kunden auf sein Konto buchen. Diese Abwicklung erfolgt über die Banknetze.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Nein	Name und Kreditkarten-nummer bekannt
Micropayment möglich	Nein	Nur bedingt, da sinnvoller Mindestbetrag 3 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Nein	Kreditkartendaten relativ ungesichert
Mobilität hoch	Ja	Kreditkarte gut transportabel
Kosten Händler gering	Nein	Einmalig ab 400 DM, monatlich variabel (z.B. 60 DM), pro Transaktion: ca. 3% Disagio zzgl. Überprüfungsgebühr zw. 0,12 und 1,35 DM (Kosten z.T. aus [3])
Kosten Kunde gering	Ja	Jahresgebühr Kreditkarte
Einstieg einfach	Ja	Muss aber Kreditkarten-besitzer sein
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	3 DM bis kundenabhängiges Limit
Akzeptanz hoch	Ja	Sehr viele Händler, ca. 10 Millionen Endkunden
Tranparenz hoch	Ja	

Tabelle 2.5: Bewertung Kreditkarte

Eignung für eVerlage: Die Kreditkartenzahlung ist leider mit relativ hohen Transaktionskosten verbunden. Dadurch lohnen erst Zahlungen ab 3 DM [3], auch wenn diese in Deutschland nicht üblich sind. Somit ist die Kreditkarte hauptsächlich für die Aufladung des Kundenkontos verwendbar.

SET - Secure Electronic Transfer:
Anders als in den USA, ist die Kreditkartenzahlung in Europa nicht sehr weit verbreitet. Dies liegt vor allem in der mangelnden Datensicherheit begründet. Damit die Akzeptanz der Kreditkarte steigt, haben sich die grossen Kreditkartenentwickler und IT-Systemanbieter wie Visa, Mastercard, Microsoft, Netscape, IBM, GTE, SAIC, Terisa Systems, VeriSign u.a. zusammengeschlossen und den offenen Standard SET geschaffen.
Basis ist nach wie vor die Kreditkarte. Allerdings werden bei SET nicht die Kartendaten wie Name und Kartennummer übertragen, sondern es wird ein signiertes (Public-Key-Verfahren) Kundenzertifikat benutzt.
Bevor man als Kunde bezahlen kann, benötigt man eine Kreditkarte (VISA oder MasterCard/EuroCard) und ein Kunden-Passwort. Diese beantragt man bei seinem Kreditinstitut. Desweiteren muss auf dem Kundenrechner ein SET-Wallet installiert werden. Leider werden bis jetzt auch hier nur Windows-basierte Lösungen angeboten.
Von dem Kreditinstitut erhält man die Kreditkarte, falls man noch keine besessen hat, sowie das Einmal-Passwort. Mit diesem Passwort und dem Wallet muss man nun sein Kundenzertifikat erstellen, welches die Kreditkartennummer, den öffentlichen Schlüssel, eine Signatur der Zertifizierungsstelle und weitere Informationen enthält. Dabei wird auch eine Nutzerkennung und ein Passwort vereinbart, welche später für das Öffnen der Wallet notwendig sind. Diese Prozedur ist relativ umständlich und arbeitsaufwendig.
Jetzt kann der Kunde mit der Kreditkarte im Netz bezahlen. Er wählt SET als Bezahlmethode aus, worauf sich das Wallet öffnet. Nach Eingabe von Nutzer-ID und Passwort werden die verwendbaren Karten angezeigt. Der Kunde wählt eine Karte aus und prüft die Bestellinformationen, die im Wallet angezeigt werden. Nach der Bestätigung wird ihm die Händleridentität angezeigt. Auch diese ist mittels Public-Key-Verfahren signiert. Bestätigt der Kunde auch diese Information, dann wird das Kundenzertifikat an das Händlersystem übertragen. Dieses lässt das Zertifikat bei dem zuständigen Kreditinstitut in Echtzeit prüfen und informiert das Wallet des Kunden über den Ausgang der Zahlung. Das Wallet speichert die Transaktionsdaten für spätere Kontrollzwecke (Informationen zum Ablauf aus [10]).
Da die Zahl der Betrugsfälle im Kreditkartenbereich besonders im Internet stark gestiegen ist, dürften auch die Banken und Händler ein Interesse an dieser sichereren Art der Kreditkartenzahlung haben.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Händler erhält nur anonymes Kundenzertifikat
Micropayment möglich	Nein	Nur bedingt, da sinnvoller Mindestbetrag 3 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	Verwendung von Public-Key-Verfahren, keine Kreditkartennummer wird verschickt
Mobilität hoch	Nein	Export und Einbindung auf anderem Rechner möglich, aber umständlich
Kosten Händler gering	Nein	
Kosten Kunde gering	Ja	Jahresgebühr Karte
Einstieg einfach	Nein	Recht komplizierte Vorarbeiten notwendig
Bedienung einfach	Nein	Authentifizierung und Kartenauswahl
Zahlungsbereich gross	Ja	3 DM bis kundenabhängiges Limit
Akzeptanz hoch	Ja	Sehr viele Händler, ca. 10 Millionen Endkunden
Tranparenz hoch	Ja	

Tabelle 2.6: Bewertung SET

Eignung für eVerlage: SET eignet sich genau wie die Kreditkarte wegen der mangelnden Micropaymentfähigkeit nur für die Aufladung des Kundenkontos eines registrierten Nutzers. SET hat dabei gegenüber der Kreditkarte die Vorteile der Anonymität und höheren Sicherheit, ist dafür im Gegenzug aber komplizierter zu bedienen. Daher wird es vorerst nicht bei eVerlage angeboten werden.

2.2.4 Zahlung per Mobiltelefon

In Deutschlang besitzen immer mehr Menschen ein Mobiltelefon. Diese sind sehr gut zu transportieren, ständig erreichbar und personengebunden. All diese Eigenschaften haben dazu geführt, dass auch Zahlungssysteme entwickelt wurden, die Mobiltelefone benutzen. Durch die Personenbindung ist eine Identifikation des Kunden automatisch gegeben, so dass der Kunde mit seinem Telefon nur noch die

Zahlung bestätigen muss. Da solche Telefone im eingeschalteten Zustand mitunter auch anderen Personen zugänglich sind, wird meist noch eine weitere Authentifizierung mittels PIN durchgeführt.
Ein Nachteil der Verfahren auf Mobiltelefonbasis sind zur Zeit die möglichen Probleme wie Funklöcher und leere Batterien. Dadurch kann es passieren, dass das jeweilige Zahlungssystem kurzfristig nicht genutzt werden kann.
Dafür ist andererseits die Mobilität enorm hoch, da man das Mobiltelefon in der Regel bei sich führt.
Die Mobilfunknetze gelten als sehr sicher [18]. Dies ist eine Grundvoraussetzung für die folgenden Verfahren, da z.B. Authentifikations-PINs über diese Netze versandt werden.

paybox:
paybox war die erste Entwicklung auf dem Gebiet der Zahlung per Mobiltelefon. Der Kunde muss sich mit seiner Kontoverbindung bei diesem Unternehmen registrieren lassen. Daraufhin erhält man eine PIN und hat 5 DM Grundgebühr an die Paybox AG zu zahlen. Dem Kunden entstehen bis auf diese jährliche Grundgebühr keine weiteren Kosten (Anmeldung aus [15]).
Der Kunde wählt im Shop des Händlers die gewünschten Waren. Bei dem folgenden Bezahlvorgang per paybox muss er seine Mobilfunknummer oder eine mit der Paybox AG vereinbarte Aliasnummer angeben. Diese Informationen schickt der Händlerserver zusammen mit den Transaktionsdaten an das paybox-System, welches dann über das GSM-Netz eine Verbindung zu dem Mobiltelefon des Kunden herstellt. Dem Kunden werden nun per Sprachcomputer Verwendungszweck und Betrag mitgeteilt. Der Kunde kann dann die Zahlung mittels der von der Paybox AG ausgegebenen PIN bestätigen oder den Vorgang abbrechen. Bei getätigter Zahlung zieht paybox den Zahlungsbetrag vom Kundenkonto per Lastschrift ein und überweist die Einnahmen zweimal monatlich auf das Händlerkonto.
Interessant ist auch die Möglichkeit, Geld zwischen zwei paybox-Kunden zu transferieren, wobei paybox den Einzug und die Überweisung für die Kunden abwickelt (Zahlungsablauf aus [3]).
Die Deutsche Bank ist mit 50 % an der Paybox AG beteiligt und sorgt somit für einen reibungslosen Zahlungsverkehr.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Händler kennt nur Alias oder Mobiltelefonnummer
Micropayment möglich	Ja	Zur Zeit ab 1 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	Kunde muss aber PIN geheim halten
Mobilität hoch	Ja	
Kosten Händler gering	Nein	Einmalig 250-2500 Euro, 100-300 Euro monatlich, 3-5 % Transaktionsgebühr [3])
Kosten Kunde gering	Ja	5 DM/Jahr
Einstieg einfach	Ja	Aber Anmeldung notwendig
Bedienung einfach	Ja	
Zahlungsbereich gross	Nein	5-50 Euro
Akzeptanz hoch	Ja	Nutzung auch in Taxiunternehmen, 120.000 Kunden (3/2001 [3]))
Tranparenz hoch	Ja	Transaktionen über Webseite abrutbar

Tabelle 2.7: Bewertung paybox

Eignung für eVerlage: paybox ist bereits in eVerlage integriert. Das System lohnt sich aber nur zum Aufladen des Kundenkontos oder für den Kauf von hochpreisigen Dokumenten.

Street Cash:
Ein paybox recht ähnliches System ist Street Cash der Firma Inatec. Bevor der Kunde das Zahlungsverfahren nutzen kann, muss er sich auch hier vorher bei dem Paymentanbieter registrieren lassen. Dabei müssen Mobiletelefonnummen und persönlichen Zahlungspräferenzen (z.B. MasterCard, VISA, Lastschrift) angegeben, sowie eine PIN festgelegt werden (Registriervorgang aus [16]).
Soll ein Bezahlvorgang gestartet werden, so muss der Kunde in einem Webformular sein Login (e-mail-Adresse+Passwort oder Telefonnummer) eingeben. Der Händlerserver übermittelt eine Nachricht mit dem Preis der Ware und dem Login des Kunden an den Street Cash-Server. Daraufhin sendet dieser eine SMS mit einer Zahlungsaufforderung an das Kundenmobiltelefon. Ist der Kunde einverstanden, dann bestätigt er die Zahlung, indem er die geheime Autorisierungs-PIN per SMS zurücksendet. Nach positiver Prüfung der PIN erhält der Händler eine

Bestätigung des Vorganges vom Street Cash-Server (Ablauf von [17]).
Der Hauptunterschied zu paybox besteht darin, dass bei Street Cash das SMS-Verfahren als Kommunikationsmittel benutzt wird. Dies bringt aber auch viele SMS-typische Nachteile mit sich. SMS-Nachrichten werden nicht zwingend in Echtzeit ausgeliefert. Dadurch kann es passieren, dass Zahlungen lange dauern, oder ein Bezahlvorgang nicht zustande kommt, weil die Nachrichten erst zu spät zugestellt werden. Der SMS-Versand erfolgt über die GSM-Mobilfunknetze. *„Die GSM-Mobilfunknetze gelten derzeit weltweit als die sichersten Wege der Datenübertragung.“* [18]
Street Cash kann mit der paysafecard gekoppelt werden, so dass theoretisch Zahlungen ab 0.01 Euro möglich wären. Die Transaktionsgebühren betragen aber 0.30 Euro ([19]), so dass Zahlungen unter 1 Euro kaum lohnenswert sind.
Will man Street Cash im Internet anbieten, so muss man als Händler an das Paymentsystem powercash21, welches ebenfalls von Inatec stammt, angeschlossen sein [16]. Durch diese Anbindung entstehen allerdings sehr hohe Grundkosten beim Händler (powercash21: einmalig 645 Euro zzgl. Mehrwertsteuer, 27 Euro zzgl. Mehrwertsteuer mtl. ([19]). Der Kunde muss die SMS-Kosten für die PIN-Übertragung bezahlen.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Kann in Verbindung mit anonymer paysafecard genutzt werden
Micropayment möglich	Mittel	Da 0.30 Euro Transaktionsgebühr
Echtzeit-Clearing	Ja	Aber Probleme mit SMS-Versand möglich
Sicherheit hoch	Ja	GSM-Verschlüsselung
Mobilität hoch	Ja	
Kosten Händler gering	Nein	Hohe Anfangskosten für Händler
Kosten Kunde gering	Mittel	SMS-Versandkosten
Einstieg einfach	Ja	Aber Anmeldung notwendig
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	
Akzeptanz hoch	Nein	Noch im Anfangsstadium
Tranparenz hoch	Nein	Ungewissheit beim SMS-Versand

Tabelle 2.8: Bewertung Street Cash

Eignung für eVerlage: Die möglichen Probleme mit dem SMS-Versand und die hohen Grundkosten auf der Händlerseite sind negative Punkte dieses Verfahrens. Auch die Kundenzahlen sind noch recht gering. Eventuell könnte man über eine reine Pilotimplementierung der Street Cash Lösung diskutieren, falls dies möglich wäre. Die powercash21 Plattform verdient eventuell eine separate Eignungsprüfung für eVerlage.

2.2.5 Zahlung mit Chipkarte

Chipkarten oder Smartcards sind wegen ihrer positiven Eigenschaften, wie Robustheit, Transportabilität und Handlichkeit, immer noch recht beliebt und kommen als Kredit- oder Sparkassenkarte, Krankenkassenkarte, Studentenausweis und in vielen weiteren Gebieten zum Einsatz. Inzwischen ist dabei die Technologie sehr weit vorangeschritten, so dass die Chips bereits kleine Microcomputer mit RAM und Dateisystem darstellen. Dies eröffnet vielen neuen Anwendungen die Möglichkeit, die Chipkarte für ihre Zwecke zu nutzen. Eine dieser Möglichkeiten ist die Verwendung als transportable Geldbörse.

GeldKarte:
Die GeldKarte ist ein Smartcard mit GeldKartechip, der bereits heute auf den meisten Kredit- und Sparkassenkarten vorhanden ist. Auf diesem Chip befindet sich die Applikation „electronic cash“, welche die Funktionalität einer elektronischen Geldbörse anbietet. Diese Karten werden deshalb auch „Börsenkarten“ genannt. Da zur Zeit bereits mehr als 45 Millionen solcher Karten im Umlauf sind, ist die Anzahl der potentiellen Nutzer sehr hoch. GeldKarten wurden nur in Deutschland emitiert, so dass dieses Verfahren auch nur dort nutzbar ist.
Der Kunde muss als erstes seine Karte an einem Ladeterminal aufladen. Dabei wird Geld von seinem Konto nach Eingabe der PIN auf die Karte transferiert, aber auch Baraufladung ist möglich. Die Karte kann dabei mit maximal 400 DM aufgeladen werden.
Der Kunde kann dann mit der Karte bezahlen. Befindet er sich in einem Geschäft, dann zahlt er an dem dortigen Kartenterminal. Wenn der Kunde aber über seinen Internetrechner einkaufen will, so muss er sich erst ein spezielles Kartenterminal zulegen und installieren.
Wurde der Bezahlvorgang gestartet, so hat der Kunde die Chipkarte in das Kartenterminal einzuführen. Daraufhin wird auf dem Display des Kartenterminals der

abzubuchende Betrag und bei Internetzahlungen noch die Händleridentität angezeigt. Der Kunde muss nun die Zahlung mit der Kartenterminaltastatur bestätigen oder abbrechen. Eine PIN ist dabei nicht notwendig.
Wurde die Zahlung bestätigt, so wird der Betrag sofort von der Karte des Kunden auf die Händlerkarte gebucht. Dabei entstehen nur 0.3 %, aber mindestens 0.02 DM Disagio (Disagio aus [37]). Durch diese geringen Transaktionskosten ist die GeldKarte sehr gut als Micropaymentzahlungsmittel verwendbar. Der Händler rechnet meist einmal pro Tag seine gebuchten Einnahmen ab.
Neben den normalen, kontobezogenen Karten (Börsenkarten) gibt es weisse GeldKarten (Wertkarten), die nicht kontogebunden und somit vollkommen anonym sind. Weisse GeldKarten können nur gegen Bargeld am Bankschalter mit speziellen Terminals (BSFT-BankenSonderFunktionsTerminal) aufgeladen werden.
Verliert man seine GeldKarte, so hat der Finder die Möglichkeit, den aktuellen Ladebetrag über das Internet auszugeben, da keine PIN-Eingabe erforderlich ist und ein Paymentserver normalerweise keine Information über gesperrte Karten besitzt, womit ein Abweien der Zahlung nicht möglich ist.

Eignung für eVerlage: Das GeldKarteverfahren ist gut für eVerlage geeignet, da es alle wichtigen Anforderungen erfüllt. Es erfolgt eine anonyme, sichere Transaktion in Echtzeit. Micropayment kann wegen geringer Transaktionskosten sehr gut realisiert werden. Der Zahlungsbereich ist recht gross und die Preise können vom Händler pfenniggenau festgelegt werden. Die Zahl der potentiellen Nutzer ist durch die breitgefächerte Ausgabe des GeldKartechips sehr hoch, aber leider durch einen recht aufwendigen und kostenintensiven Einstieg begrenzt.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Anonym gegenüber Händler und evtl. gegenüber Bank
Micropayment möglich	Ja	
Echtzeit-Clearing	Ja	Transaktion erfolgt sofort
Sicherheit hoch	Ja	verschieden eSicherheitsmassnahmen Aber Problem bei Kartenverlust
Mobilität hoch	Ja	Karte mobil, aber Kartenterminal muss vorhanden sein
Kosten Händler gering	-	Keine Festpreise bekannt
Kosten Kunde gering	Nein	Benötigt ein Kartenterminal
Einstieg einfach	Nein	Einrichtung diverser Komponenten
Bedienung einfach	Nein	Manchmal etwas unklar
Zahlungsbereich gross	Ja	0.02-400 DM
Akzeptanz hoch	Ja	Aber im Internet noch wenige Händler Viele potentielle Kunden
Tranparenz hoch	Ja	Auflistung der letzten zehn Zahlungen und letzten drei Aufladungen möglich

Tabelle 2.9: Bewertung GeldKarte

2.2.6 Inkassoverfahren

click&buy:
Ein recht einfaches, aber effizientes Verfahren wird von der Firma Firstgate mit „click&buy“ angeboten. Es handelt sich dabei um ein Inkassosystem, in welchem Firstgate die Guthaben von Händlern sowie Kunden verwaltet, sowie die Transaktionen vornimmt.
Will man mit diesem Bezahlsystem einkaufen, so muss man sich zuerst bei Firstgate anmelden. Dabei füllt man eine Einzugsermächtigung mit seinen Kontodaten aus und gibt einen Benutzernamen sowie ein Passwort an, mit welchem man sich später authentifiziert. Um Betrug vorzubeugen werden 2 Pfennige auf dieses Konto überwiesen und im Verwendungszweck eine PIN angegeben, mit welcher der Kunde seine Zugang freischalten kann.
Gelangt man beim Surfen zu einem kostenpflichtigen, von Firstgate verwalteten

Inhalt, dessen Preis zwischen 10 Pfennigen und 9.90 DM liegen kann, so muss man sich zunächst mit Benutzernamen und Passwort authentifizieren. Daraufhin öffnet sich eine Webseite, auf welcher die Ware kurz dargestellt und die Kosten nochmals aufgelistet werden. Bestätigt der Kunde diese Information, so wird das Kundenkonto bei Firstgate mit dem Betrag belastet und die meist elektronische Ware ausgeliefert. Der Firstgate-Server fungiert dabei als Proxy, indem er die Datei vom Händlerserver holt und an den Kunden weiterleitet (Ablauf aus [3]).
Üblicherweise gewährt der Anbieter den Zugriff auf die Datei für ein Zeitfenster. Damit ist in den meisten Fällen gewährleistet, dass ein Kunde die bezahlte Ware auch bei Verbindungsproblemen erhält. Bei Reklamationen oder Problemen kann dieser Zugriff kurzfristig gesperrt werden.
Firstgate zieht je nach Betrag meist monatlich das vom Kunden ausgegebene Geld per Lastschrift von dessen Konto ein, und überweist andererseits die Monatseinnahmen eines Händlers auf das Händlerkonto. Transaktionen werden dabei in Zusammenarbeit mit der Deutschen Bank durchgeführt
Dabei entstehen dem Händler fixe Kosten von nur 5 Euro pro Monat und eine einmalige Anmeldegebühr von 25 Euro [26]. Leider ist dafür die nach Umsatz gestaffelte Provision sehr hoch und liegt zwischen 40 % bei 50 Euro/Monat und 30 % bei 5000 Euro/Monat. Für den Kunden fallen keine Kosten an. Dies ist ein grosser Bonus bei der Kundengewinnung und somit bei der Steigerung der Akzeptanz (Kosten aus [27]).

Eignung für eVerlage: click&buy eignet sich wegen seiner Micropaymentfähigkeit gut zum Bezahlen von Dokumenten zur Ansicht. Eine Aufladung des Kundenkontos bei registrierten Kunden ist wegen der Zahlungsobergrenze von click&buy allerdings nur begrenzt möglich. Leider ist die Firstgate-Proxy-Technologie nicht mit der eVerlage-Lösung kompatibel, da keine dynamischen Links unterstützt werden. Damit ist eine Integration vorerst nicht möglich.
Ein weiteres Manko des Bezahlsystemes sind die hohen Transaktionsgebühren.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Nur Firstgate besitzt Kunden-informationen
Micropayment möglich	Ja	0.10-9.90 DM
Echtzeit-Clearing	Ja	Buchung auf Firstgate-Händlerkonto
Sicherheit hoch	Mittel	Basiert nur auf Passwort
Mobilität hoch	Ja	Keine spezielle Hard- oder Software
Kosten Händler gering	Nein	Hohe Umsatzprovision
Kosten Kunde gering	Ja	
Einstieg einfach	Ja	Nur Anmeldung und Freischaltung
Bedienung einfach	Ja	Nur Authentifizierung
Zahlungsbereich gross	Nein	Nur Micropayment bis 9.90 DM
Akzeptanz hoch	Ja	Schnell steigende Händler- und Kundenzahlen
Tranparenz hoch	Ja	Einfaches Modell und Kontokontrolle über Webseite

Tabelle 2.10: Bewertung click&buy

2.2.7 Zahlung über Telefonrechnung

Gegenwärtig gibt es in Deutschland kaum noch einen Haushalt ohne festen Telefonanschluss, auch wenn dieser Trend durch die weitere Verbreitung der Mobiltelefone eventuell rückläufig wird. Jeder feste Telefonanschluss wird auch abgerechnet. Somit erhalten fast alle Haushalte eine Telefonrechnung. Dies hat dazu geführt, dass eine Reihe von Zahlungssystemen darauf basiert, Zahlungen des Kunden einfach über die Telefonrechnung abzurechnen. Damit wird sehr effizient ein bereits bestehendes Zahlungsnetz weiter genutzt.
Bezahlt wird jeweils durch den Anruf zu einer kostenpflichtigen Servicenummer. Dies geschieht manuell durch den Nutzer oder automatisch per Modem/Netzanschluss. Als Kunde sollte man Vorkehrungen treffen die Telefonkosten ständig unter Kontrolle zu haben (spezielle Software, Gebührenzähler), da laut der Zeitschrift iX [3] auch schon Betrugsfälle mit der zu installierenden Software auftraten.
Während bei den mobiltelefonbasierenden Zahlungssystemen eine Anmeldung

erforderlich ist und das Telefon nur zu Authentifikation genutzt wird, so ist bei der Zahlung per Telefonrechnung eine Anmeldung nicht immer notwendig.

Net900 classic:
Die Firma in medias res bietet in Partnerschaft mit der deutschen Telekom Net900 classic an. Dabei wird eine 0192-Nummer in Verbindung mit dem Fernmelderechnungsdienst zur Abrechnung per Telefonrechnung genutzt [3].
Bei der klassischen Variante von Net900 muss der Nutzer als erstes ein Plug-In (Netscape, Internet Explorer, Opera) auf seinem Rechner installieren. Dieses arbeitet als Proxyserver und filtert so die kostenpflichtigen Seiten, welche anhand der URL identifiziert werden [28]. Existiert bereits ein Proxy, so werden die normalen Anfragen von dem Net900 Proxy weitergeleitet.
Bei einer kostenpflichtigen Seite jedoch weisst das Plug-In auf den zahlungspflichtigen Inhalt hin und gibt die genauen Kosten aus. Nach Bestätigung durch den Kunden unterbricht das Plug-In die bestehende Internetverbindung und initiiert eine Verbindung zu der kostenpflichtigen 0192-Nummer. Der Server hinter dieser Nummer liefert dann die gewünschten Daten aus. Nach dem Bezahlvorgang wird die normale Internetverbindung wieder hergestellt [3].
Als Tarife werden 0.29-4.99 DM/Minute oder 0.29-25.00 DM/Klick berechnet (Kosten aus [7]).
Für den Händler fallen 75 DM Einrichtungs- sowie 7.50 DM Monatsgebühren an (Gebühren aus [3]). Desweiteren muss eine Umsatzpauschale abgeführt werden. Diese hängt von dem Lizenznehmer ab und liegt zwischen 15 und 30 %.

Eignung für eVerlage: Bis auf die eingeschränkte Mobilität erfüllt Net900 classic die wichtigsten Anforderungen. Leider sind auch die Transaktionsgebühren recht hoch. Dafür sind die Grundgebühren niedrig, wodurch anfänglich geringere Nutzerzahlen nicht sehr problematisch sind. Eine Einbindung wäre sinnvoll, wenn die Zuwachsrate an Nutzern hoch genug ist, und die Integration relativ leicht vorgenommen werden kann.
Möglicherweise können wie bei click&buy Probleme mit der eingesetzten Proxytechnologie auftreten. Eine Nachfrage per e-Mail blieb leider unbeantwortet.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Nur Servicenummernanruf
Micropayment möglich	Ja	Ab 0.29 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	
Mobilität hoch	Nein	An Telefonanschluss gebunden
Kosten Händler gering	Mittel	Relativ hohe Umsatzprovision
Kosten Kunde gering	Ja	Keine
Einstieg einfach	Ja	Nur Softwareinstallation
Bedienung einfach	Ja	
Zahlungsbereich gross	Nein	Bis 25 DM sinnvoll
Akzeptanz hoch	Mittel	Mehr als 120 Händler, Kundenzahlen unbekannt
Tranparenz hoch	Mittel	Ohne Gebührenzähler ist ist Kostenkontrolle schwierig

Tabelle 2.11: Bewertung Net900 classic

2.2.8 Prepaidverfahren

paysafecard:
Die paysafecard.com Wertkarten AG aus Österreich bietet ein interessantes Prepaid-Zahlungsverfahren an. Als Kunde muss man nur eine paysafecard kaufen, um an dem Zahlungssystem teilzunehmen. Diese gibt es in 2 Grundversionen. Die metallblaue Karte ist für Kunden über 18 Jahre und mit einem Wert von ATS 300 , ATS 500 oder ATS 1000 (ATS=österreichische Schillinge) geladen. Die rote Karte ist für Kunden unter 18 bestimmt und mit ATS 300 oder ATS 500 geladen. Die rote Karte ist für bestimmte Webseiten (Erotik, Wetten usw.) gesperrt. Auf der Webseite der Firma exisitert eine Auflistung der Händler, welche die Karte verkaufen.
Bevor man mit der Karte einkaufen kann, muss man eine 16-stellige PIN freirubbeln. Der Server von paysafecard hat zu dieser PIN den Ladebetrag der Karte gespeichert. Das Guthaben wird also bei paysafecard direkt verwaltet, während Karte und PIN nur der Authentifikation dienen.
Will man ein paysafecard-Angebot kaufen, so wird man automatisch mit dem paysafecard-Server verbunden. Der Kunde muss die PIN seiner Karte eingegeben.

Der paysafecard-Server reserviert den zu zahlenden Betrag auf dem paysafecard-Konto und informiert den Händlerserver über den Erfolg der Transaktion. Der Händler kann die Ware ausliefern. Erst nach Auslieferung wird der reservierte Betrag wirklich gebucht.
Zahlungen sollen auch im Micropaymentbereicht möglich sein. Desweiteren ist es möglich bis zu 10 Karten für einen Einkauf zu verwenden, so dass die Obergrenze für die metallblaue Karte bei ATS 10000 (rund 720 Euro) liegt. Zusätzlich kann man die Karte auf der Firmenwebseite mit einem Passwort schützen. Dieses Passwort muss dann bei Zahlungen neben der PIN mit eingegeben werden. Somit ist ein Kartenverlust nicht mehr so problematisch, da der Finder nicht mit dieser Karte bezahlen kann, weil er das Passwort nicht kennt (Informationen aus [32]).
Als Kunde muss man sich nicht anmelden und keine Konto- oder sonstigen Daten preisgeben. Durch das Kartenkonzept ist paysafecard eines der anonymsten Bezahlverfahren.
Auch für Händler entstehen keine Grundgebühren. Es wird allerdings ein mit der paysafecard.com Wertkarten AG zu vereinbarendes Disagio fällig.
Da also für beide Seiten keine Grundkosten anfallen und der Einstieg recht einfach ist, darf man sicherlich mit schnell steigenden Händler- und Kundenzahlen rechnen. Seit März 2001 läuft die paysafecard in Deutschland an, aber auch andere europäische Länder folgen bereits.

Eignung für eVerlage: Die paysafecard ist ein interessantes, anonymes Micro- und Macropaymentverfahren. Da Prepaidkarten bei Mobiltelefonen sehr beliebt sind, werden sich die Kunden wahrscheinlich leicht mit diesem Konzept anfreunden. Insgesamt bietet das System recht günstige Voraussetzungen für eine Integration in eVerlage. Ein Hemmniss könnte allerdings, vor allem in Deutschland, zur Zeit noch der Erwerb der Prepaidkarten sein.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Komplette Anonymität gegenüber Händler und Zahlungssystemanbieter
Micropayment möglich	Ja	Ab 0.01 Euro
Echtzeit-Clearing	Ja	Echtzeitüberprüfung der PIN
Sicherheit hoch	Ja	Bei Passwortschutz
Mobilität hoch	Ja	Karte transportabel
Kosten Händler gering	Ja	Keine Fixkosten, aber Transaktionsgebühren
Kosten Kunde gering	Ja	Keine
Einstieg einfach	Ja	Kann spontan erfolgen
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	0.01-ca. 720 Euro
Akzeptanz hoch	Mittel	Aber steigend
Tranparenz hoch	Ja	Kontokontrolle möglich

Tabelle 2.12: Bewertung paysafecard

2.2.9 Elektronisches Geld

Bei den Überlegungen, wie man Zahlungen über das Internet realisieren kann, sind natürlich auch Ideen entstanden, virtuelles Geld zu erschaffen. Dabei wird das Geld durch verschlüsselte Dateien repräsentiert, welche auf Echtheit geprüft werden können. Diese Dateien können wie normales Geld weitergereicht werden. Diese Zahlungsmethode hat den Vorteil, dass nur das Geld auf Echtheit und nicht der Kunde auf Zahlungsfähigkeit getestet werden muss. Damit ist auch die Möglichkeit gegeben, anonym aufzutreten, da es für den Händler irrelevant ist, wer einkauft, solange die Echtheit des virtuellen Geldes garantiert ist.
Da bei einer Transaktion keine bankseitigen Überprüfungen notwendig sind, kommen die Verfahren ohne Transaktionsgebühren beim Einkauf aus. Somit ist auch die Möglichkeit gegeben, Zahlungen im Micropaymentbereich anzubieten.
Der Zahlungsablauf gestaltet sich meist folgendermassen:

1. Der Kunde lädt seine elektronische Geldbörse (Wallet) mit elektronischem Geld auf. Dabei wird sein Konto belastet.

2. Er bezahlt im Shop des Händlers, wobei die elektronischen Geldwerte transferiert werden.

3. Der Händler tauscht beim Anbieter das erworbene elektronische Geld wieder in normales Geld um.

Leider waren die bisherigen Lösungen nicht sehr erfolgreich, was sicher auch in der etwas umständlichen Handhabung des elektronischen Geldes begründet liegt. So wurde z.B. eCash von DigiCash für bankrott erklärt, obwohl es ein sehr ausgereiftes System sein soll, welches seiner Zeit voraus war. Auch CyberCoin von Cybercash (http://www.cybercash.de) wurde eingestellt, weil die Akzeptanz auf Kundenseite gering war, obwohl unter anderem die Dresdner Bank, die Commerzbank und die Hypovereinsbank hinter dem Unternehmen stehen [4].
Das Verfahren Millicent von der Digital Equipment Corporation ist noch recht neu und wird bald auch für den europäischen Markt verfügbar sein. Damit können elektronische Waren im Wert von 0.10 US$ bis 10 US$ bezahlt werden. Es werden händlerspezifische Geldgutscheine verwendet und das System ist nicht sehr sicher (Informationen von [33]).

Eignung für eVerlage: Eine Integration von Verfahren in eVerlage, die auf elektronischen Münzen basieren, scheint zur Zeit nicht sinnvoll, da die Systeme entweder nicht ausgereift sind oder nicht akzeptiert werden.

2.2.10 Elektronische Schecks

NetCheque:
Die University of South California bietet mit NetCheque ein Scheckverfahren für das Internet an. Dabei können Zahlungen zwischen zwei Personen getätigt werden, sei es von Kunde zu Händler oder zwischen Privatpersonen. Beide Parteien müssen dazu bei NetCheque angemeldet sein und eine Software inklusive elektronischem Scheckbuch installieren. Der Sender kann mit „write-cheque“ einen Scheck erstellen und ihn an den Empfänger schicken. Dieser kann sich dann mit „deposit-cheque“ den Betrag nach Überprüfung des Schecks gutschreiben (Informationen von [34]).

Eignung für eVerlage: Gering, denn bisher findet dieses etwas umständliche Verfahren noch keine grosse Anwendung.

2.2.11 Paymanagement

Da die Akzeptanz von Bezahlsystemen seitens der Händler auch stark vom Einrichtungs- und Wartungskosten abhängt, haben sich neben Hausbanken auch unabhängige Anbieter auf die Einrichtung solcher Systeme und die Abwicklung von Zahlungen spezialisiert.
Da ein Händler natürlich dem Kunden eine möglichst breite Palette an Bezahlmöglichkeiten bieten will, stellen auch die Paymentserviceprovider mehrere Verfahren zur Verfügung.
Auch hier gibt es sehr viele Lösungen mit unterschiedlichen Konditionen. Darum sollen hier nur einige Anbieter aufgelistet werden:

- e-Tra von Fun Communications (http://www.fun.de) unterstützt Online-Überweisung (fun HomePay), GeldKarte (fun SmartPay), Lastschrift (fun eddPay) und Verträge mit digitaler Signatur (fun eContractor)
- Wirecard (http://www.wirecard.de) bietet Lastschriften und sichere Kreditkartenzahlung
- Isoft (http://www.isoft.de) stellt Net900 und WebBill (Kreditkarte, Nachnahme, Lastschrift) und bald auch paybox zur Verfügung
- powercash21 (http://www.powercash21.de) von Inatec (http://www.inatec.de) bietet Kreditkarte, SET, Lastschrift-online, Street Cash, paysafecard, Leasing & Kredit, Mahnwesen, Inkasso und vieles mehr
- Gesellschaft für Zahlungssysteme (GSZ) (http://www.gsz.de)
- Telecash (http://www.telecash.de)
- Eurodebit (http://www.eurodebit.de)
- SelfServe (http://www.selfserve.de)

2.3 Integration des Bezahlverfahrens GeldKarte für eVerlage

Nachdem die wichtigsten Verfahren vorgestellt wurden, wird im folgenden Unterkapitel die Verwendbarkeit dieser Bezahlverfahren für das eVerlage-Projekt zusammengefasst.

Da das Projekt schon seit einiger Zeit im Internet läuft, werden bereits einige Zahlungsverfahren dem Kunden angeboten [35].

Integriert sind für registrierte Kunden:

- Lastschrift
- Paybox
- Gutscheinsystem
- Kreditkarte (in Vorbereitung, noch kein SSL-Zertzifikat vorhanden)

Integriert sind für Gastnutzer:

- Paybox
- Gutscheinsystem

Aufgrund der im vorangegangenen Unterkapitel gezogenen Eignungsbetrachtungen scheint eine Reihe weiterer Bezahlverfahren für eVerlage geeignet. Damit ein grösserer Kundenkreis erreicht werden kann, sollten Überlegungen angestellt werden, bei welchen Bezahlsystemen eine Einbindung lohnenswert sein könnte. Dabei müssen betriebswirtschaftliche Gesichtspunkte und mögliche Hard- und Softwareprobleme, wie inkompatible Schnittstellen zwischen Händler und Bezahlsystem, bei der Anbindung betrachtet werden. Diese Überprüfungen erachte ich für die folgenden Zahlungsverfahren als sinnvoll:

- SET oder aposto: für eine sicherere Kreditkartenzahlung

- Net900 Kontopass: sollte angeboten werden, um das Lastschriftverfahren auf den Micropaymentbereich auszuweiten
- Street Cash und/oder ein anderer mobilfunkbasierter Dienst: zum Bezahlen per Mobiltelefon
- GeldKarte: für eine sichere Zahlung mittels Chipkarte
- Net900 classic und/oder ein sonstiger Telefondienstanbieter: für das einfache Bezahlen per Telefonrechnung
- paysafecard: als anonymes Prepaidverfahren
- evtl. Nutzung eines Paymentproviders zur einfacheren und automatischen Zahlungsabwicklung

Bei all diesen Verfahren müssen Vorteile wie grössere Kundenzahlen und Nachteile, wie hohe Grundkosten abgewogen werden, bevor die Entscheidung über die Integration oder Nichtintegration des Zahlungssystems gefällt werden kann.

Da das GeldKarteverfahren als Micropaymentverfahren sehr gut in das eVerlage-Konzept zu passen scheint und auf der Händlerseite theoretisch relativ leicht und problemlos zu integrieren ist, wurde beschlossen es als Zahlungsverfahren für registrierte Kunden und Gastnutzer anzubieten.

Hier eine Auflistung der Vor- und Nachteile des GeldKartenzahlungsverfahrens:

Vorteile:

- Hohe Akzeptanz: Der GeldKartechip ist mit mehr als 45 Millionen Kredit-, Sparkassen- und sonstigen Chipkarten ausgegeben [36] worden. Somit existiert eine hohe Anzahl potentieller Nutzer. Allerdings wird dieses Potential noch viel zu wenig ausgeschöpft, was unter anderem auf zu geringe Werbung zurückzuführen ist.
- Grosser Zahlungsbereich: Mit der GeldKarte können Beträge von wenigen Pfennigen bis hin zu grösseren Summen von 400 DM transferiert werden. Die Preise unterliegen keiner Einschränkung und können pfenniggenau festgelegt werden.

- Geringe Transaktionskosten: Ein Disagio von nur 0.3 % macht das Zahlungssystem für den Händler sehr attraktiv und Micropaymentzahlungen wirtschaftlich.
- Multifunktionalität: Mit der GeldKarte kann man nicht nur im Internet, sondern auch an vielen anderen Orten (Strassenbahn, Parkautomat, Kaufhaus) bezahlen. Desweiteren kann die GeldKarte durch ein gutes Betriebssystem nicht nur für Zahlungsanwendungen sondern parallel z.B. auch als Studentenausweis, Rabattkarte oder ähnliche Sonderanwendungen benutzt werden. Dadurch ist die GeldKarte vielseitig und attraktiv, wodurch die Chancen auf breite Akzeptanz wachsen.
- Mobilität: Die GeldKarte ist ein mobiles Zahlungsmittel. Leider wird diese Mobilität durch die Notwendigkeit des Vorhandenseins eines Kartenterminals begrenzt.
- Sicherheit und Transparenz: GeldKartenzahlungen sind durch Verwendung diverser Sicherheitsmechanismen sehr sicher. Die Zahlung selbst läuft transparent ab und kann auch später durch temporäre Speicherung der Transaktionsdaten auf der GeldKarte vom Kunden überprüft werden.
- Echtzeit-Clearing: Bei Internetzahlungen ist ein sofortiges Clearing möglich. Damit hat der Händler eine Zahlungsgarantie, d.h er kann die Ware ohne Risikoausliefern. Umständliche Mahnverfahren werden so umgangen.
- Anonymität: Der Kunde bleibt mit diesem Zahlungsverfahren recht anonym gegenüber dem Händler. Auch wenn dieser aus den Zahlungsdaten die Kartennummer des Kunden ablesen kann, ist keine Verbindung zu den Personendaten möglich. Bei Verwendung einer „weissen GeldKarte“ ist komplette Anonymität sichergestellt.

Nachteile:

- Hardwarebedarf: Will man mit der GeldKarte im Internet einkaufen, so benötigt man ein spezielles Kartenterminal. Diese sind zur Zeit noch recht teuer (um 200 DM). Es ist allerdings anzunehmen, dass bei stärkerer Verbreitung des Zahlungssystems auch diese Geräte billiger werden. Leider würden laut IZV4-Studie [5] nur 16.2 % der Interneteinkäufer sich zusätzliche Hardware anschaffen, um mehr persönliche Sicherheit beim Bezahlen im

Internet zu erhalten. Somit wäre eine preisgünstige, aber sichere Integration in eine spezielle Computertastatur evtl. anstrebenswert.

- Kartenverlust: Über den Ladebetrag des Chips kann jeder verfügen, der im Besitz der Karte ist. Bei Verlust oder Diebstahl könnte der rechtmässige Besitzer somit den Ladebetrag einbüssen. Hier wäre es sinnvoll, wenn ein Händlersystem bei der Zahlungsabwicklung von den Banken erfahren könnte, ob eine Karte wegen Verlust gesperrt ist.

- Aufladung: Die GeldKarte muss stets mit genügend Bargeld aufgeladen sein. Zu Zeit kann man nur in Banken und Kreditinstituten die Aufladung vornehmen. Ein Ladevorgang am heimischen Computer ist zwar geplant, aber noch nicht spezifiziert.

- Deutschlandbeschränkung: Die GeldKarte wird nur in Deutschland akzeptiert. Prinzipiell könnten auch ausländische Firmen ein Bezahlen mit der GeldKarte anbieten, was bisher aber noch nicht der Fall ist.

- Akzeptanzproblem: Durch die Deutschlandbeschränkung ist die Zahl der Händler, welche die GeldKartenzahlung anbieten, vorerst relativ begrenzt. Es besteht also zur Zeit noch ein Akzeptanzproblem bei der GeldKartenzahlung im Internet. Sollte sich die Anzahl der Händler nicht erhöhen, wird kaum ein Kunde die nötigen 200 DM für ein Kartenterminal ausgeben, da das Kosten-Nutzen-Verhältnis zu schlecht ist. Es ist daher wichtig, dass der Einstieg für Händler billig und einfach erfolgt.

Trotz der Nachteile ist das GeldKarteverfahren sehr gut für den Kauf von elektronischen Waren im Micropaymentbereich, wie sie von eVerlage angeboten werden, geeignet.
Leider gibt es bei dieser recht neuen Technologie vor allem auf der Kundenseite noch Probleme in Bezug auf die Softwareunterstützung. Deshalb kann die GeldKartenzahlung bisher nur begrenzt eingesetzt werden. Aus diesem Grund habe ich eine Softwareschnittstelle geschrieben, welche eine Lücke in der Reihe der fehlenden Softwarekomponenten schliessen soll. Für die Einordnung der Schnittstelle wird zunächst in Kapitel 3 die GeldKartenzahlung im Internet detaillierter dargestellt. Kapitel 4 beleuchtet den zugrundeliegenden Chip näher, bevor in Kapitel 5 die eigentliche gk-Schnittstelle selbst beschrieben wird.

Kapitel 3

Bezahlen mit der GeldKarte im Internet

Im folgenden Kapitel wird das Bezahlverfahren „GeldKarte im Internet“ genauer betrachtet. Es werden dabei die möglichen Zahlungsabläufe und der Aufbau des Zahlungssystems dargestellt. Desweiteren werden die Hard- und Softwareanforderungen auf Händler- und Kundenseite näher beleuchtet und die verwendeten Sicherheitsmechanismen einer GeldKartenzahlung vorgestellt.

Vorbemerkung: In Deutschland existiert ein „Zentraler Kreditausschuss“ (ZKA, http://www.zka.de). Dieser ist eine bankübergreifende Institution, in der alle grossen Finanzinstitute und Banken zusammenarbeiten, und welche eine Normierungsfunktion hat. Der ZKA verfasst die „Schnittstellenspezifikationen der ZKA-Chipkarte “ (vormals „Schnittstellenspezifikationen für die ec-Karte mit Chip“). In dieser Spezifikation wird festgelegt, wie die GeldKarte aufgebaut ist, und wie sie sich verhalten soll. Desweiteren wurden und werden wichtige Punkte wie Zahlungsablauf, Kommunikation, Hard- sowie Softwareschnittstellen, Internet-Händlersysteme und vieles mehr spezifiziert. Der ZKA sorgt dafür, dass erst die theoretischen Grundlagen zur GeldKarte vorliegen, bevor auf deren Basis dann weitere Zahlungskomponenten und spezielle Projekte entwickelt werden können. Dies hat den Vorteil, dass es von Anfang an keine Kompatibilitätsprobleme gibt.
Aus diesem Grund oriertieren sich auch die in diesem Kapitel dargestellten Mechanismen an der Schnittstellenspezifikation des ZKA. Im Anhang wird beschrieben, wo man diese Spezifikation beziehen kann.

3.1 Darstellung der GeldKartenzahlung im Internet und ihrer drei Abbuchungsmodelle

Dieses Unterkapitel beschäftigt sich mit dem Zahlungsablauf und den daran beteiligten Parteien.

3.1.1 Ablauf einer GeldKartenzahlung im Internet

Bei einer GeldKartenzahlung im Internet sind mehrere Parteien beteiligt:

- Kunde
- Bank des Kunden
- Händler
- Bank des Händlers
- Händlerevidenzzentrale (HEZ)
- Kartenevidenzzentrale (KEZ)

Der **Kunde** bezahlt eine Ware beim **Händler** mit seiner GeldKarte. Die kartenausgebende **Bank des Kunden** speichert und verwaltet dabei den Ladewert der GeldKarte des Kunden. Die **Händlerevidenzzentrale** des Händlers rechnet die Tageseinnahmen ab und sorgt dafür, dass das Geld auf das Konto des Händlers bei der **Bank des Händlers** transferiert wird. Die **Kartenevidenzzentrale** ist eine Kontrollinstanz, die Schattensolden der GeldKarten führt, so dass Fehler oder Betrugsfälle registriert werden. Den Banken und Sparkassen stehen die folgenden 4 Einrichtungen Einrichtungen zur Verfügung, welche jeweils eine Händlerevidenz- und Kartenevidenzzentrale führen. Diese Einrichtungen [37] sind:

- *Informatik Kooperation für die Sparkassen-Finanzgruppe*

- *Betriebswirtschaftliches Institut der deutschen Kreditgenossenschaften (BIK) für den Bundesverband der deutschen Volksbanken und Raiffeisenbanken*
- *Bundesverband öffentlicher Banken Zahlungsverkehrsdienstleistung mbH (VöB Zvd) für den Verband öffentlicher Banken*
- *Bankverlag für den Bundesverband deutscher Banken*

Mit Hilfe von „Kopfstellen“ tauschen diese Institutionen Transaktionsdaten aus, um den verbandsübergreifenden Zahlungsverkehr zu ermöglichen.

In dem folgenden Ablaufbild 3.1 wird der Geldkreislauf bei einer GeldKartenzahlung im Internet grafisch dargestellt. Danach wird der Ablauf genauer erklärt.

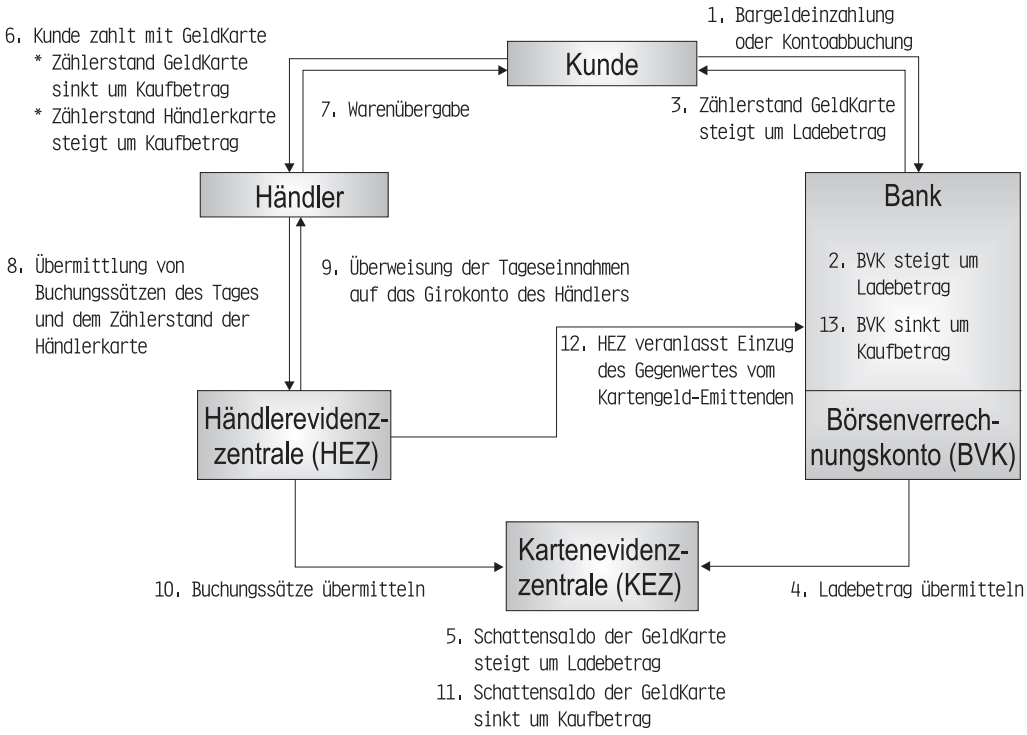

Abbildung 3.1: Geldkreislauf [38]

Ablauf:

1. **Bargeldeinzahlung oder Kontoabbuchung:** Der GeldKartenbesitzer will Geld auf seine Geldkarte laden. Dieser Vorgang erfolgt üblicherweise an einem Ladeterminal in einem Kreditinstitut. Dabei wir das Geld nach PIN-Eingabe vom Konto abgebucht und auf die Karte geladen. Bargeldbasierte Ladeterminals sind zwar auch möglich, aber nicht im breiten Einsatz und werden deshalb nicht weiter betrachtet. Kontoungebundene Karten können nur am Bankschalter an speziellen Terminals (BSFTs) aufgeladen werden. Es wird in nächster Zeit möglich sein, die kontogebundene GeldKarte am heimischen PC über ein angeschlossenes Kartenterminal aufzuladen (http://www.kuk.de), auch wenn dieser Vorgang noch nicht vom ZKA spezifiziert wurde. Diese Möglichkeit wird die Kundenakzeptanz des Zahlungsmittels GeldKarte erhöhen. Eine erste Version solch einer Ladetransaktion wurde auf der CeBit 2001 vorgestellt.

2. **Ladebetrag wird auf BVK (Börsenverrechnungskonto) gebucht:** Der eingezahlte Ladebetrag wird zusätzlich dem BVK der kartenausgebenden Bank (Bank des Kunden) gutgeschrieben. *„Börsenverrechnungskonten sind Sammelkonten, unter denen die kumulierten Guthaben aller von einem Institut oder einer Institutsgruppe herausgegebenen Börsenkarten typischerweise nach Kartentyp und Verfallsjahr getrennt verwaltet werden.“* [37]

3. **Erhöhen des Einheitenzählers auf der Karte:** Der Betrag der GeldKarte wird um den eingezahlten Ladebetrag erhöht. Der Maximalbetrag einer GeldKarte, welcher 400 DM bzw. 200 Euro beträgt, kann dabei nicht überschritten werden.

4. **Ladevorgang wird übermittelt:** Ladevorgang wird an die bankgruppeneigene KEZ übermittelt, welche die Schattensolden der ausgegebenen GeldKarten führt.

5. **Schattensaldo der Karte wird um Ladebetrag erhöht:** Der Schattensaldo wird um Ladebetrag erhöht. Das eingezahlte Geld ist also auf der GeldKarte, dem BVK und dem Schattensaldo der GeldKarte gespeichert.

6. **Kunde wählt im Internetshop seine Waren aus und zahlt diese mit der Zahlungsmethode „GeldKarte“:** Bei dieser Transaktion sinkt

der Zählerstand auf der GeldKarte, während der Zählerstand auf der Händlerkarte steigt. Die Händlerkarte befindet sich üblicherweise bei einem Zahlungsprovider.
Es existieren drei verschiedene Abbuchungsmodelle wie die Transaktion ablaufen kann (siehe 3.1.2).

7. **Warenübergabe:** Der Händler erfährt vom Zahlungssystem, dass der Bezahlvorgang erfolgreich war und liefert nun die Ware an den Kunden.

8. **Buchungssätze und Zählerstand des Tages werden übermittelt:** Zusammenfassung aller GeldKartenzahlungen (Buchungssätze) des Tages und dem Zählerstand der Händlerkarte zu einem Gesamtpaket, der sogenannten „Händlersammelgutschrift“, und anschliessende Übermittlung an die zuständige HEZ.

9. **Überweisung der Tageseinnahmen auf Girokonto des Händlers wird veranlasst:** Die HEZ überweist den Wert des Gesamtpaketes auf das Girokonto des Händlers. Dabei wird das Verrechnungskonto der HEZ belastet.
Dieser Schritt ist die einzige Stelle, an der die Bank des Händlers ins Spiel kommt.

10. **Buchungssätze werden weiter übermittelt:** *„Die HEZ trennt die Zahlungssätze anhand der Bankleitzahl der Börsenkarte* [GeldKarte des Kunden] *nach ihrer Verbandszugehörigkeit: Zahlungssätze mit Bankleitzahlen fremder Verbände bzw. Institute meldet die Händlerevidenzzentrale an die verantwortlichen Kopfstellen der Verbände und Zahlungssätze mit eigenen Bankleitzahlen an die eigene Kartenevidenzzentrale.* [...] *Die Kopfstellen trennen die gemeldeten Zahlungssätze nach den verantwortlichen Kartenevidenzzentralen auf und reichen sie an diese weiter.“* (aus [37])

11. **Schattensaldo wird um Kaufbetrag vermindert:** Der Schattensaldo der Kunden-GeldKarte bei der KEZ wird um den Kaufbetrag, den der Kunde beim Händler bezahlt hat (Schritt 6) und welcher mit dem Buchungssatz bei Schritt 10 von der HEZ übermittelt wurde, verringert.

12. **HEZ veranlasst Einzug des Gegenwertes vom Kartengeld-Emittenten:** Die HEZ zieht den Gegenwert der einzelnen Buchungssätze von den Kartengeld-Emittenten (GeldKarte ausgebende Banken) ein.

13. **GeldKarten-Aufladungsgegenwert auf BVK verringert sich um Kaufbetrag:** Dabei wird das BVK bei der Bank des Kunden um den Kaufbetrag verringert. Somit stimmt das Guthaben auf dem BVK wieder mit dem Ladewert aller ausgegebenen Kunden-GeldKarten überein.

Die bankinternen Verrechnungsmethoden können von interessierten Lesern in der Verrechnungsdarstellung [37] genauer nachgelesen werden.

3.1.2 Die drei Abbuchungsmodelle

Wie in Schritt 6 von 3.1.1 erwähnt, existieren drei Modelle, wie das Geld von der GeldKarte des Kunden auf die Händlerkarte gebucht werden kann (Informationen aus [40]). Diese sind:

- Abbuchen
- Inkrementelles Abbuchen
- Schnelles inkrementelles Abbuchen

Im folgenden werden nun diese Abbuchungsmodelle näher vorgestellt.

Abbuchen

Bei dem Vorgang des **Abbuchen** handelt es sich um einen einmaligen Abbuchungsvorgang. Es wird nur ein einzelner Betrag von der GeldKarte des Kunden abgebucht und der Händlerkarte gutgeschrieben.
Nachdem der Kunde seine Waren ausgewählt und GeldKarte als Zahlungmittel angegeben hat, wird (nach Start der GeldKarte-Applikation auf dem Kundenrechner) auf dem Display des Kartenterminals nach Überprüfung der Händleridentität diese angezeigt. So kann sich der Kunde sicher sein, dass der richtige Händler das Geld bekommt und kein Betrug (z.B. falscher Webshop unter zweiter URL) vorliegt.
Desweiteren wird auf dem Display des Kartenterminals der abzubuchende Betrag sowie dessen Währung angezeigt. Dadurch sind Zahlungsbetragsmanipulationen,

wie das Abbuchen eines höheren Betrages, ausgeschlossen.
Hat der Kunde diese Angaben durch Drücken der „Bestätigen“-Taste am Kartenterminal akzeptiert, wird der angezeigte Betrag von der GeldKarte des Kunden abgebucht und auf die Händlerkarte transferiert. Wenn keine Fehler aufgetreten sind, die ein Rückbuchen erforderlich machen könnten, ist die Zahlung hiermit abgeschlossen.
Abbuchen ist daher am besten für die einmalige Zahlung z.T. grösserer Beträge geeignet, wie z.B. CDs, Bücher, Konzertkarten, Softwarelizensen, Zeitungsabonnements. Natürlich können auch geringe einmalige Beträge wie z.B. ein Los oder eine kleine Spende mit der Zahlungsmethode „Abbuchen“ bezahlt werden.

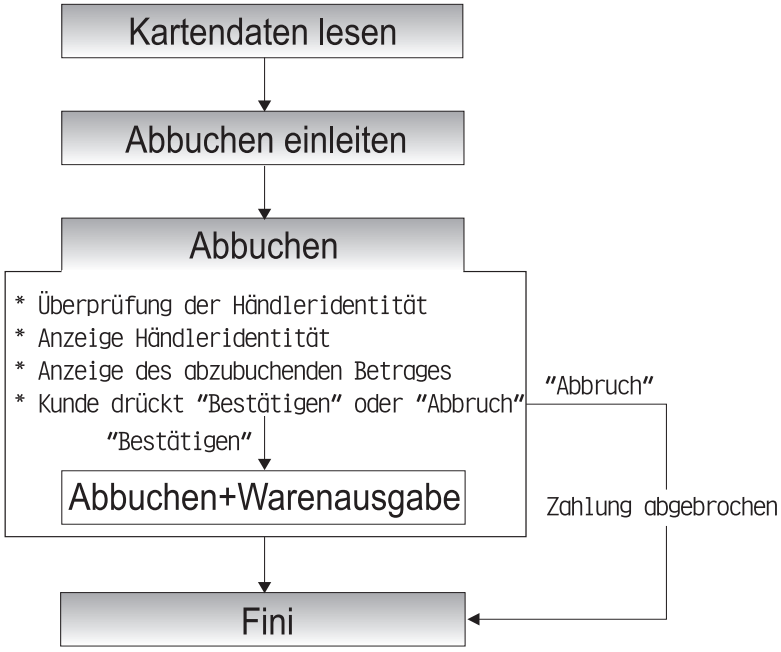

Abbildung 3.2: Abbuchen

Inkrementelles Abbuchen

Im Gegensatz zu dem gerade dargestellten einmaligen Abbuchen handelt es sich bei dem Verfahren des **Inkrementellen Abbuchen** um eine Transaktion in

mehreren Schritten. In jedem dieser Schritte wird ein meist recht geringer Betrag transferiert.
Bei dieser Bezahlmethode wählt der Kunde anfangs „Inkrementelles Bezahlen mit der GeldKarte“ auf der Webseite des Händlers/Anbieters. Danach wird einmalig die Händleridentität auf dem Display des Kartenterminals angezeigt, damit der Kunde diese überprüfen kann. Anschliessend nutzt der Kunde den kostenpflichtigen Dienst des Anbieters.
Jedesmal, wenn ein Zahlungsbetrag abgebucht werden soll, z.B. beim Aufruf eines Dokumentes, wird der Betrag und die Währung auf dem Display des Kartenterminals angezeigt. Der Kunde muss diesen Betrag bestätigen, wenn er mit der Zahlung einverstanden ist. Dann wird der Geldwert dieses Teilbetrages von der Kunden-GeldKarte abgebucht und der Händlerkarte gutgeschrieben. Bei erfolgreicher Transaktion erhält der Kunde nun die Ware.
Bei der Bezahlmethode Inkrementelles Abbuchen ist es für den Kunden also bequem möglich mehrere kleine Beträge zu zahlen. Aufgrund dieser Tatsache wird Inkrementelles Abbuchen meist anders eingesetzt als das normale einmalige Abbuchen. Während mit Abbuchen sicher mehr (aber nicht ausschliesslich) materielle Waren bezahlt werden, so ist inkrementelles Abbuchen besonders gut für elektronische Dienstanbieter geeignet.
So können sehr gut elektronische Daten bzw. der Zugriff auf solche Daten über das Internet verkauft werden. Anwendungsbeispiele wären der Verkauf von Musik (z.B. 20 Pfennige pro Lied einer gestreamten CD), Suche in kostenpflichtigen Nachrichten- oder sonstigen Archiven/Datenbanken und natürlich auch der Zugriff auf Textdokumente wie elektronische Bücher, Zeitschriften und Normen, wie sie im eVerlage Projekt angeboten werden.
Der Kunde kann bei eVerlage z.B. in den Ausgaben der Zeitschrift ct recherchieren, oder wissenschaftliche Bücher lesen. Mit der GeldKarte und dem Inkrementellen Abbuchen ist es so möglich, dass der Kunde ein Kapitel eines Buches oder einer Zeitung zu einem geringen Betrag (1 DM) kauft. Findet der Kunde ein weiteres, interessantes Kapitel, muss er nur wieder den Betrag am Kartenterminal bestätigen und bekommt so Zugriff auf die gewünschten Daten. Dies hat auch den Vorteil, dass man wirklich nur für Informationen bezahlt, die man benötigt oder interessant findet.
eVerlage bietet auch die DIN an. Mit Hilfe des Inkrementellen Abbuchen könnte man also recht einfach einzelne Normen kaufen, ohne diese wie bisher erst bestellen zu müssen.

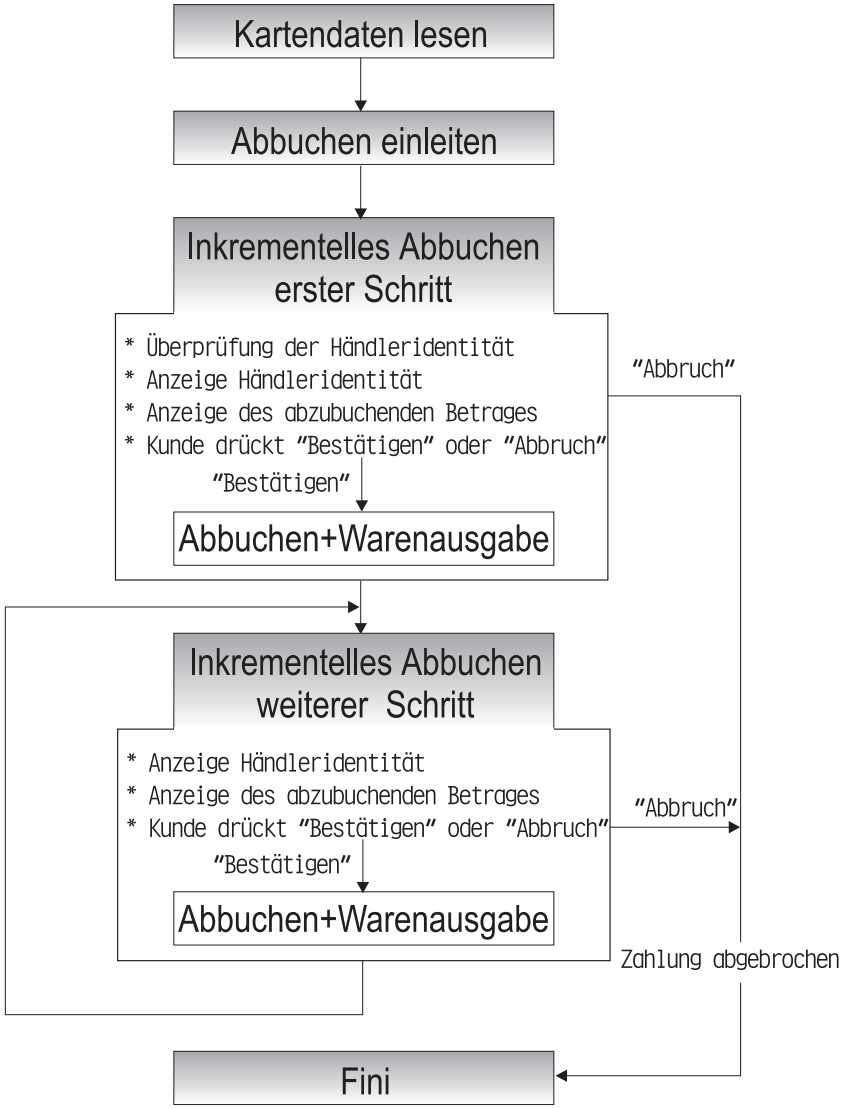

Abbildung 3.3: Inkrementelles Abbuchen

Schnelles inkrementelles Abbuchen

In manchen Anwendungsfällen würde das ständige Bestätigen des Zahlungbetrages den Kunden stören. Deswegen wurde die Bezahlmethode **Schnelles in-**

krementelles Abbuchen eingeführt, in welchem die Teilbeträge nicht extra bestätigt werden müssen, sondern automatisch abgebucht werden.
Dazu muss der Nutzer zwei Obergrenzen angeben. Zum einen die Obergrenze für den Gesamtbetrag, die in einer Session nicht überschritten werden soll (z.B. 20 DM). Zum zweiten die Obergrenze für die jeweiligen Teilbeträge, die bei einer Abbuchung nicht überschritten werden soll (z.B. 2 DM). Gibt der Kunde also 20 DM als Obergrenze Gesamtbetrag und 2 DM als Obergrenze Teilbetrag an, so kann er z.B. Dokumente bis zu 20 DM einkaufen, wobei ein Dokument nicht teuerer als 2 DM sein darf.
Diese Obergrenzen sind allerdings nicht frei vom Kunden wählbar. Der Händler legt beide Obergrenzen fest und bietet sie dem Kunden an. Der Kunde kann diese Grenzen nun annehmen, oder er verringert eine Grenze bzw. alle beide. Der Kunde kann die Grenze aber nicht höher ansetzen als vom Händler vorgeschlagen. Dieser Mechanismus bietet für den Händler gewisse Sicherheiten.
Nachdem die Obergrenzen festgelegt sind, wird einmalig die überprüfte Händleridentität angezeigt, bevor der Kunde im Rahmen der Obergrenzen nach Belieben einkaufen. Erst wenn die Obergrenze für den Gesamtbetrag erreicht wurde, muss sich der Kunde wieder um die Zahlung kümmern. Er wird dann gefragt, ob er die Obergrenze weiter erhöhen möchte. Setzt der Kunde die Grenze, unter Berücksichtigung der Händlergrenze, höher, so kann er weiter einkaufen bzw. kostenpflichtige Dienste nutzen. Wenn er die Grenze nicht weiter erhöht, ist der Zahlungsvorgang beendet. Eine Zahlung wird auch dann abgebrochen, wenn der abzubuchende Teilbetrag grösser als der festgelegte Maximalteilbetrag ist.

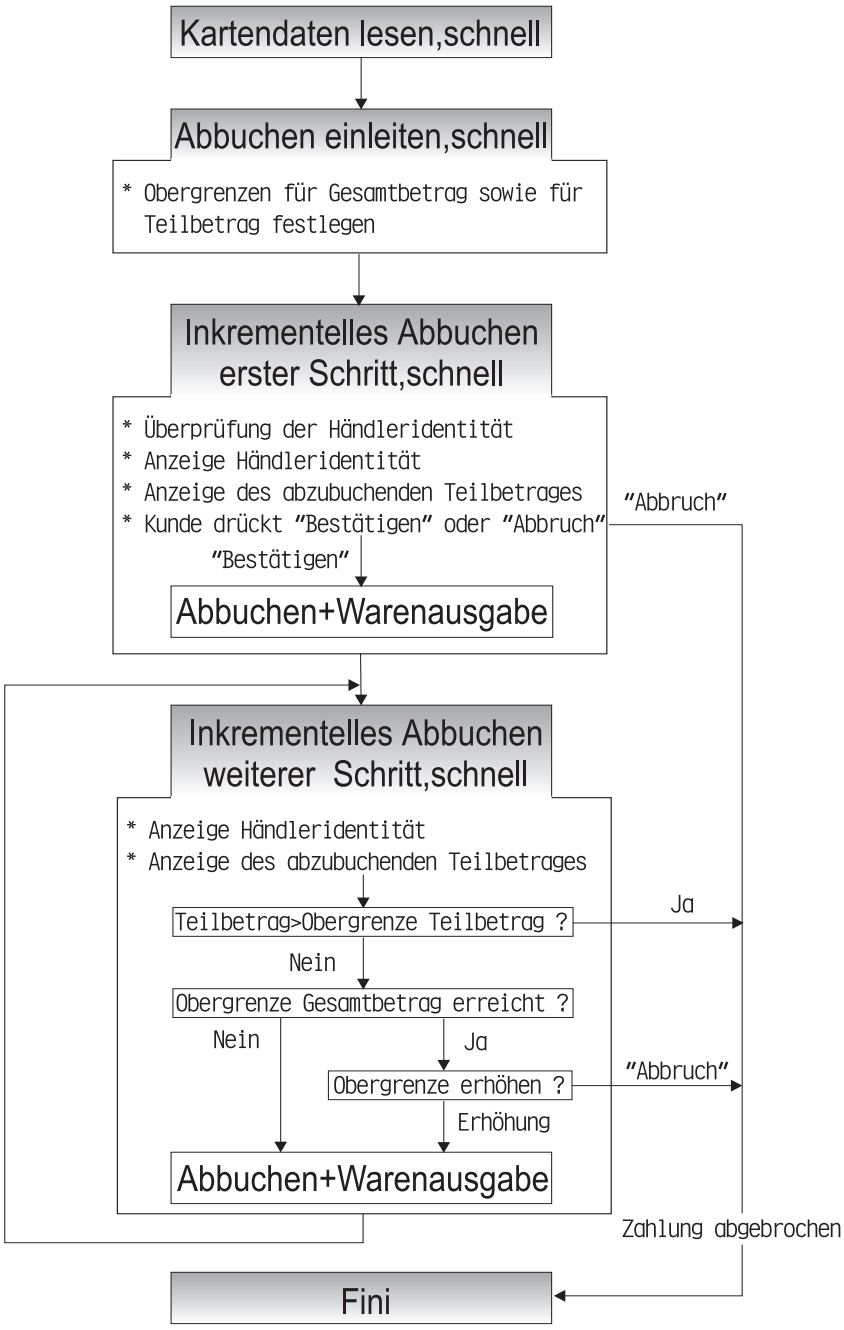

Abbildung 3.4: Schnelles inkrementelles Abbuchen

3.2 Internetzahlung auf Basis eines verteilten Händlersystems

3.2.1 Aufteilung des Kartenterminals bei einem verteilten Händlersystem

Um GeldKartenzahlungen zu realisieren, benötigt man ein vom ZKA zugelassenes Händlersystem. Bei einer normalen GeldKartenzahlung, wie in einem Geschäft oder an einem Parkscheinautomaten, existiert genau ein physisches Kartenterminal, ein sogenanntes **Akzeptanzterminal**. Darin ist eine Händlerkarte integriert. Ausserdem führt der zahlende Kunde seine GeldKarte in den Kartenterminalslot ein. Somit entsteht eine Einheit aus Kartenterminal, Händlerkarte und Kunden-GeldKarte, welche Geld von der Kunden-GeldKarte auf die Händlerkarte transferieren kann.
Bei einer Internetzahlung hingegen befnden sich Händlerkarte und GeldKarte des Kunden an räumlich getrennten Orten. Die Händlerkarte befindet sich beim Zahlungssystemprovider und die Kunden-GeldKarte beim Kunden. Somit ist es nicht mehr möglich, dass sich beide Karten mit Hilfe nur eines abgesicherten Gerätes austauschen. Das Akzeptanzterminal einer normalen GeldKartenzahlung muss daher für Internetzahlungen aufgespalten werden. Diese Aufteilung des Terminals wird in folgender Grafik 3.5 verdeutlicht:

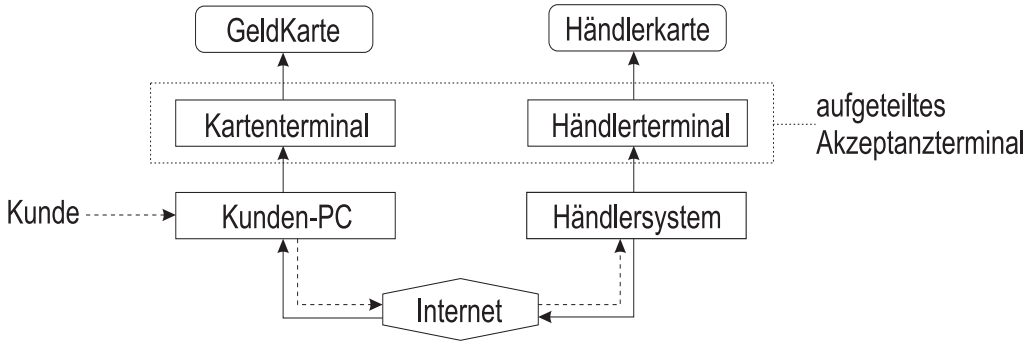

Abbildung 3.5: Verteiltes Kartenterminal

Man kann erkennen, dass bei Internet-GeldKartenzahlungen Kunden- und Händlerseite räumlich getrennt und nur durch das Internet verbunden sind. Wie

erwähnt, muss deshalb das Akzeptanzterminal aufgeteilt werden, so dass sich ein Teil beim Kunden (Kartenterminal) und ein Teil beim Händler (Händlerterminal) befindet. In diesen Terminals befinden sich dann auch die jeweiligen Karten - die GeldKarte im Kartenterminal des Kunden und die Händlerkarte im Händlerterminal.
Aufgrund der räumlichen Trennung der an der Zahlung beteiligten Transaktionskomponenten (Kartenterminals, Karten) spricht man bei Internetzahlungen auch von einem **verteilten Händlersystem**.
Neben dem Akzeptanzterminal existieren ausserdem noch das **Kassenschnittterminal** zur Erstellung eines Tagesdatensatzes und das **Einreichungsterminal** zur Einreichung dieses Tagesdatensatzes an die HEZ. Beide Komponenten bleiben nach der Aufteilung des Akzeptanzterminals auf der Händlerseite und werden üblicherweise in spezielle Paymentserver integriert.

3.2.2 Struktur eines verteilten Händlersystems

Wie ebend dargestellt, wird bei GeldKartenzahlungen im Internet das Akzeptanzterminal und damit das Händlersystem geteilt. Mit der Teilung des Kartenterminals, mussten auch die Aufgaben und Zuständigkeiten neu vergeben werden. Da zwei Parteien beteiligt sind, bot sich eine Master-Slave-Struktur an. Dabei wurde der Händlerseite die Rolle des Masters und der Kundenseite die Rolle des Slaves zugeteilt.
Nachdem also der Kunde mit Hilfe seines Internet-PC den Bezahlvorgang gestartet hat, übernimmt das Händlersystem die Steuerung des kompletten Zahlungsablaufes zwischen Kunden-GeldKarte und der Händlerkarte.

Hier der schematische Aufbau eines verteilten Händlersystems. In dem Schemas 3.6 wird dabei detailierter auf die Kundenseite und deren Zahlungskomponenten eingegangen .

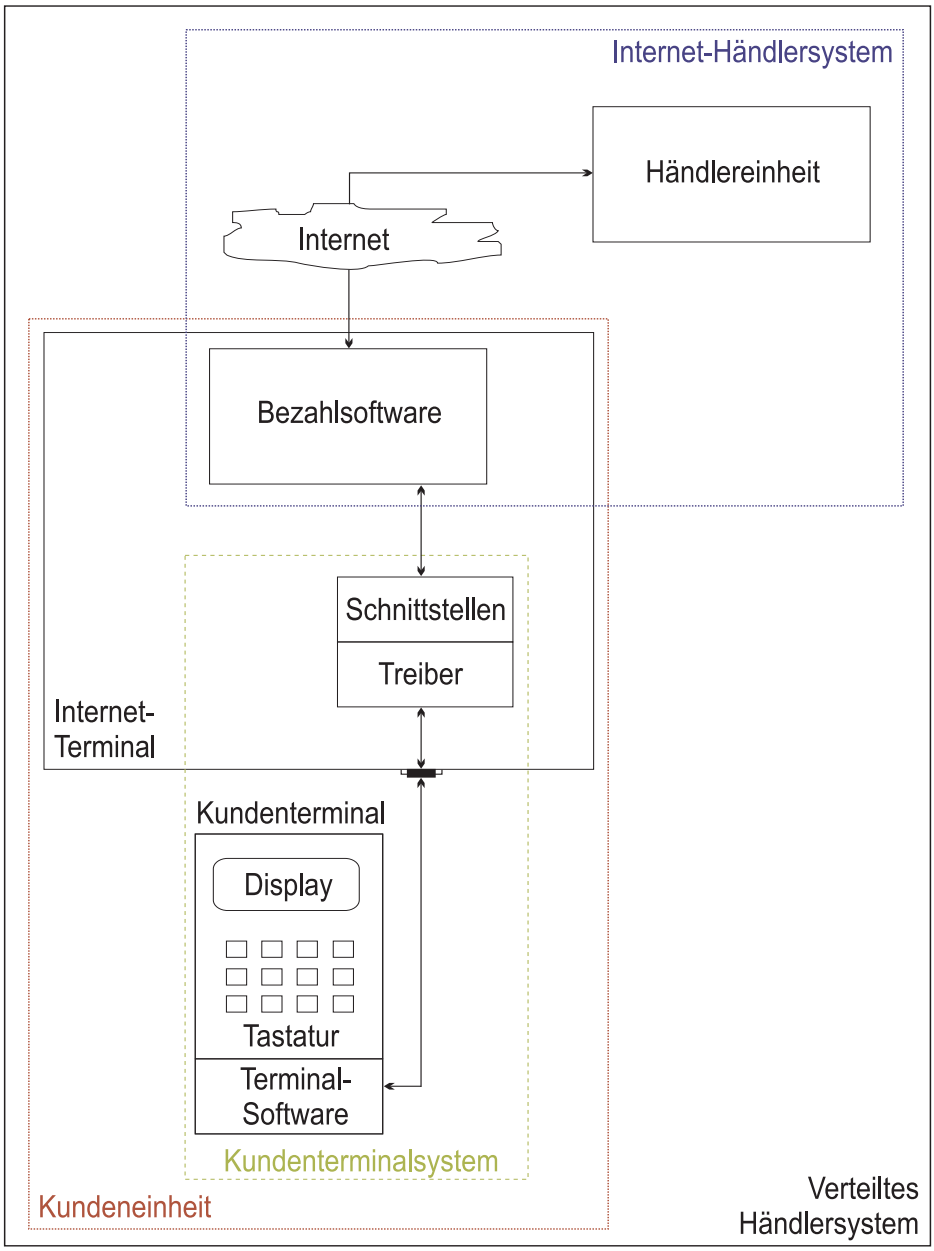

Abbildung 3.6: Verteiltes Händlersystem [39]

Anhand des Schema werden im folgenden die Komponenten eines verteilten Händlersystems erläutert:

- *„Die **Kundeneinheit** ist der beim Kunden befindliche Bestandteil des verteilten Händlersystems. Sie besteht aus dem Internet-Terminal des Kunden mit der darauf befindlichen Bezahlsoftware und dem Kundenterminalsystem.“* [40] In der Praxis handelt es sich bei der Kundeneinheit meist um einen Computer (PC oder Workstation), auf welchem die Bezahlsoftware installiert und an den ein Kartenterminal angeschlossen ist. Aber auch ein Mobiltelefon mit GeldKarteanbindung, sowie vorinstallierter Bezahlsoftware wäre denkbar.
  Hinweis: Kartenterminal und Kundenterminal sind zwei Bezeichnungen für ein und das selbe Gerät, welches umgangssprachlich schlicht „Kartenleser“ genannt wird. Ich werde im folgenden immer von „Kartenterminal“ sprechen.

- *„Die **Händlereinheit** ist der beim Händler befindliche Bestandteil des verteilten Händlersystems. Sie erfüllt die händlerseitigen Anforderungen an ein Akzeptanzterminal und beinhaltet die Funktionalitäten des Einreichungs- und Kassenschnitttterminals.“* [40]

- Das **Internet-Terminal** ist ein System aus Hard- und Software, welches über das Internet die Verbindung mit der Händlereinheit herstellt. Normalerweise ist das Internet-Terminal ein Computer mit Internetanschluss.

- *„Das vom Hersteller des Kartenterminals gelieferte Gesamtsystem, bestehend aus dem Kartenterminal mit der Kartenterminalsoftware, der physischen Anbindung an oder Integration in das Internet-Terminal und der eventuell auf dem Internet-Terminal erforderlichen Software zur Kommunikation mit dem Internet-Terminal wird als **Kundenterminalsystem** bezeichnet.“* [40] Ein Kundenterminalsystem ist also ein Paket bestehend aus:

  - Kartenterminal, welches an den Computer/das Internet-Terminal angeschlossen wird (meist seriell)
  - Anschlusssoftware (Kartenterminal-Treiber)
  - Softwareschnittstelle für das Bezahlen mit der GeldKarte im Internet (GeldKarte-API)

  Eine genaue Darstellung des Kundenterminalsystems erfolgt später in 3.3.

- *„Die Software auf dem Internet-Terminal des Kunden, die der Abwicklung von Internet-Zahlungen dient und insbesondere mit dem Kundenterminalsystem kommuniziert, wird als **Bezahlsoftware** bezeichnet.“* [40] Diese

kann als Plug-In, signiertes Java-Applet oder fest installiertes Programm auf dem Internet-Terminal laufen.

- *„Der vom Händler betriebene Teil des verteilten Händlersystems, bestehend aus der Bezahlsoftware auf dem Internet-Terminal des Kunden und der Händlereinheit wird als Gesamtsystem betrachtet und als* ***Internet-Händlersystem*** *bezeichnet.“* [40]

3.2.3 Notwendige Komponenten und Voraussetzungen für eine GeldKartenzahlung im Internet

Damit ein Kunde bei einem Händler mit der GeldKarte bezahlen kann, müssen die auf beiden Seiten jeweils notwenigen Komponenten des verteilten Händlersystems vorhanden und eingerichtet sein, sowie gewisse Voraussetzungen erfüllt werden. Diese werden im folgenden getrennt nach Händler- und Kundenseite aufgeführt.

Voraussetzungen auf Händlerseite:
Ein Händler muss sich anfangs als GeldKartenzahlungsanbieter bei seinem Kreditinstitut anmelden. Daraufhin wird dem Händler eine Händlerkarte ausgestellt. Desweiteren erhält er eine feste ID-Kennung inklusive Sicherheitsschlüssel zur Authentifikation. Die ID-Kennung wird später nach Überprüfung beim Bezahlvorgang dem Kunden angezeigt.
Damit der Kunde im Internet die Waren des Anbieters per GeldKarte erwerben kann, benötigt der Händler einen Webshop, welcher die Bezahlmethode „GeldKarte“ anbietet. Desweiteren benötigt der Händler einen Paymentserver, der die Transaktionen steuert und in welchem sich dann auch die Händlerkarte befindet.

Bemerkung: Eine Händlerkarte ist nicht in jedem Fall eine physische Karte wie auf der Kundenseite. Der ZKA hat auch „virtuelle Händlerkarten“ spezifiziert. Diese virtuellen Händlerkarten sind geschützte (physisch und elektronisch) Software-Module im Hardware-Sicherheitsmodul des Händlersystems. Diese verhalten sich genauso wie normale Händlerkarten, lassen sich aber platzsparender unterbringen, da keine Chipkartenleser notwendig sind. Es existieren Zahlungsmodule wie die Cipher-Box (Giesecke&Devrient) oder SmartPay (fun communications), in denen solche virtuelle Händlerkarten verwaltet werden. Diese Module sind keine wirkliche Chipkartenlesegeräte, sondern Rechner, welche auf die abgesicherten Händlerdateien zugreifen und mit diesen arbeiten. Es findet also eine Softwareemulation der Händlerkartenfunktionalität statt.

Auf dem Paymentserver muss ein funktionsfähiges Internet-Händlersystem installiert sein. Für grosse Anbieter lohnt es sich, einen oder mehrere eigene Paymentserver zu betreiben. Dabei gibt es Komplettlösungen (z.B. Cipher Box, fun e-Tra), die oftmals mehr als ein Zahlungssystem unterstützen und somit multifunktional eingesetzt werden können. Für kleinere Anbieter oder erst anlaufende Projekte ist der finanzielle Aufwand eines eigenen Paymentservers zu hoch. Diese Händler haben die Möglichkeit, Paymentserviceprovider zu nutzen. Ein solcher Provider betreibt Paymentserver für mehrere Händler, was sich wesentlich wirtschaftlicher gestaltet. Der Provider kann auch die Wartung und Pflege des Systems erledigen. Es ist anzunehmen, dass durch die weitere Verbreitung von Zahlungssystemen im Internet bald jeder normale Internetprovider Zahlungssysteme seinen Kunden zur Nutzung anbietet.
Auf dem Paymentserver läuft der händlerseitige Teil eines bestimmten Internet-Händlersystem. Zum jetzigen Zeitpunkt existieren am Markt 3 solcher Händlersysteme mit etwas unterschiedlichen Eigenschaften, die über die Bezahlsoftware auch Auswirkung auf der Kundenseite zeigen. Deshalb möchte ich die verfügbaren Systeme kurz vorstellen. Die folgende Tabelle 3.1 soll dabei einen ersten Überblick über die Art der Bezahlsoftware, die Lauffähigkeit auf unterschiedlichen Betriebssystemen sowie die angebotenen Abbuchungsmodelle verschaffen.

	X-Pay Brokat	IPS ATOS	SmartPay fun com.
Bezahlsoftware	Online-Applet oder installiertes Applet	Online-Applet und fest installierte Applikation	Online-Applet
Windows Linux Solaris	Ja Nein Nein	Ja Nein Nein	Ja Nein Nein
Abbuchen Inkrement. Ab. Schn. ink. Ab.	Ja Nein Nein	Ja Nein Nein	Ja Nein Nein

Tabelle 3.1: Merkmale der Internet-Händlersysteme

Wie man erkennen kann, ist die Realisierung der Bezahlsoftware unterschiedlich vorgenommen worden. Leider funktionieren alle Lösungen bisher nur unter Windows. Desweiteren wird lediglich das einfach Abbuchen angeboten. Die inkrementellen Zahlungsmethoden, die bei der Nutzung der eVerlage-Bibliothek besonders praktisch wären, werden leider von noch keinem der drei Systeme angeboten.

PaymentWorks X-Pay:
Das erste vom ZKA zertifizierte Internet-Händlersystem ist PaymentWorks X-Pay mit GeldKarte Add-On der Firma Brokat Technologies. Die Bezahlsoftware, das sogenannte X-Pay Wallet, ist als Applet realisiert. Das Applet kann beim Bezahlen online geladen oder einmalig installiert und dann nur noch aufgerufen werden. Die Installation erfolgt sehr einfach und schnell per Mausklick.
Bei Fehlern oder Problemen informiert das System den Nutzer mit recht aussagefähigen Fehlermeldungen. Bei einem Test unter Solaris mit fehlernder GeldKartenschnittstelle (Datei gkapi.so fehlte), meldete das Applet: „The system component to access the card terminal could not be loaded. Aborting payment." In dieser Hinsicht ist PaymentWorks X-Pay den anderen Lösungen überlegen. Eine Linux-/Solarisunterstützung ist theoretisch möglich. Eventuell sind dafür noch nicht einmal sehr viele Änderungen an der Bezahlsoftware notwendig, da das Applet bereits bis zu einem gewissen Punkt auch unter diesen Betriebssystemen funktioniert. Bisher wurde noch nicht über eine Einbindung dieser Plattformen entschieden, da notwendige Schnittstellen fehlten und die Nutzerzahlen zu gering waren.

IPS:
Das InternetPaymentSystem (IPS) von ATOS setzt auf fest installierte Software. Der Kunde muss vor dem Einkauf die „GeldKartenKassette" (aktuell V3.01) aus dem Netz herunterladen, falls er sie nicht anderweitig bekommen hat, und danach installieren.
Am Ende der Installation muss man ein Kartenterminal oder eine CT-API eines nicht aufgeführten Gerätes auswählen. Dieser Schritt ist aus meiner sicht unnötig, da die Bezahlung über die GK-Schnittstelle läuft, welche implizit die korrekte CT-API verwenden sollte. Eine Online-Installation ist scheinbar vorgesehen, funktionierte aber bei einem Test noch nicht.
Beim Bezahlvorgang wird erst ein Java-Applet über das Internet geladen, welches die installierte Bezahlsoftware startet. Dieser Zugriff muss vom Nutzer bestätigt werden. Besser wäre hier sicher die Verwendung eines speziellen Dateityps (MIME-Type) gewesen, bei dessen Aufruf die Applikation ohne zusätzliche Abfrage gestartet wird.

SmartPay Die Paymentplattform „eTra" der Firma „fun communications" wurde bereits bei der Vorstellung der Zahlungssysteme erwähnt. Ein weiteres Modul dieser Paymentarchitektur ist fun SmartPay, mit welchem GeldKartenzahlungen vorgenommen werden können.

Signed Applet oder fest installierte Applikation Wie man aus der Tabelle 3.1 erkennen kann, verfolgen die einzelnen Internet-Händlersysteme unterschiedliche Strategien bei der Realisierung der Bezahlsoftware.
Damit diese Software das Kartenterminal bei einem Zahlungsvorgang ansprechen kann, muss Sie Programmbibliotheksfunktionen aufrufen. Dies ist ein sicherheitskritischer Zugriff auf den Kundenrechner, mit welchem der Nutzer einverstanden sein muss.
Es gibt daher zwei Möglichkeiten, dies zu gewährleisten:

- Der Kunde installiert die Bezahlsoftware wie ein normales Programm. Die Software könnte man z.B. von seinem Kreditinstitut erhalten, damit deren Integrität gewährleistet ist. Die fest installierte Applikation kann ohne weitere Abfragen die Zahlung durchführen. Allerdings muss dieses Programm erst gestartet werden. Dieser Vorgang erfolgt bei der ATOS-Lösung über ein Java-Applet und muss somit trotzdem erst vom Nutzer bestätigt werden. Allerdings kann man diesen Vorgang übernehmen, so dass der Browser die Bestätigung bei späteren Zahlungen nicht mehr benötigt.
  Vorteile:
  - geringere Ladezeit, da Software bereits auf dem Rechner installiert ist
  - nur einmalige Überprüfung der Integrität der Bezahlsoftware notwendig

  Nachteile:
  - plattformgebunden - für jedes Betriebssystem muss eine eigene Bezahlsoftware entwickelt werden
  - erst Besorgung und Installation der Software notwendig
  - Bezahlvorgang ist an den Rechner gebunden, wodurch ein Verlust der Mobilität des Zahlungsverfahrens auftritt
  - Starten der installierte Bezahlsoftware durch Webbrowser muss vom Nutzer genehmigt werden
  - Gefahr von Viren bei nachlässiger Kontrolle dieser Zugriffsberechtigung
- Eine andere Lösung ist die reine Verwendung von signierten Java-Applets. Diese werden beim Bezahlvorgang automatisch geladen und können nach

expliziter Zustimmung des Nutzers auf dessen Hard- und Software zugreifen. Vorteile:

- Verwendung der aktuellsten Version
- keine Installation, automatisches Laden der Software
- kann plattformunabhängig programmiert werden und ist somit flexibler und mobiler einsetzbar

Nachteile:

- Ladezeiten des Applet
- Überprüfung der Signatur und somit der Integrität des Applet notwendig (die Signatur kann aber bis zum Ablaufdatum oder für immer angenommen werden)
- Gefahr von Viren bei nachlässiger Kontrolle der Signatur des Applets
- keine Kontrolle über sicherheitskritische Zugriffe - das Applet kann nur mit allen Konsequenzen angenommen oder abgelehnt werden
- unbemerktes Ausspionieren des Rechners wäre so möglich
- funktioniert evtl. in Netzen mit beschränkten Nutzerrechten nicht

Voraussetzungen auf Kundenseite:
Damit ein Kunde mit der GeldKarte im Internet bezahlen kann, benötigt er eine Vielzahl an Komponenten, die untereinander kompatibel sein müssen. Diese sind:

- Computer mit Internet-Anschluss (Internet-Terminal)
- Webbrowser
- Bezahlsoftware (Teil des Internet-Händlersystem)
- Kundenterminalsystem mit Klasse 3 Kartenterminal

Im Folgenden werden die Komponenten näher erläutert.
Der **Computer** muss auf das Internet zugreifen können, um den Händlershop-Server zu erreichen. Leider sind aber zur Zeit bei weitem nicht alle internetfähigen Plattformen und Betriebssysteme für GeldKartenzahlungen nutzbar, da noch

weitere Kriterien erfüllt sein müssen. Zum einen ist es notwendig, dass ein Kartenterminal physisch an den Rechner angeschlossen werden kann. Ausserdem müssen alle notwendigen Softwarekomponenten (Browser, Bezahlsoftware, Treiber, Geld-Kartenschnittstelle) für die GeldKartenzahlung unter dem Betriebssystem des Computers lauffähig sein.
Die zum jetzigen Zeitpunkt existierenden Lösungen beschränken sich noch auf Rechner mit seriellem Anschluss und dem Betriebssystem Windows. Ziel dieser Diplomarbeit ist es, die GeldKartenzahlung auch auf anderen Plattformen zu ermöglichen.
Ist ein geeigneter Rechner gefunden, so muss auf diesem ein Webbrowser installiert sein. Mit dem Webbrowser kann der Kunde z.B. das Internetangebot des eVerlage-Servers durchsuchen und nutzen. Wenn er fündig geworden ist und bezahlen will, muss der Browser die Bezahlsoftware ordnungsgemäss laden und/oder ausführen können. Wegen der grossen Kompatibilätsprobleme und unterschiedlichen Strukturen der Browser sind oftmals nur der Microsoft Internet Explorer oder der Netscape Communicator verwendbar.
Falls ein Internet-Händlersystem genutzt wird, welches eine vorherige Installation der Bezahlsoftware fordert (zur Zeit noch ATOS IPS), so muss dies geschehen, bevor der Bezahlvorgang gestartet wird.
Damit die Bezahlsoftware die Transaktion mit der Kunden-GeldKarte vornehmen kann, ist es notwendig, dass ein zu dem Computer kompatibles Kundenterminalsystem mit Klasse 3 Kartenterminal angeschlossen und installiert ist. Die Problematik des Kundenterminalsystems wird im nächsten Unterkapitel 3.3 genauer betrachtet.

3.3 Das Kundenterminalsystem als Teilkomponente eines verteilten Händlersystems

Die wichtigste Komponente beim Kunden, der mit der GeldKarte im Internet bezahlen will, ist das Kundenterminalsystem, welches ich in diesem Unterkapitel näher beleuchten will.
Wie bereits in 3.2.2 dargestellt, ist das Kundenterminalsystem ein Komplettpaket bestehend aus Kartenterminal, Kartenterminal-Treiber und der Softwareschnittstelle für das Bezahlen mit der GeldKarte im Internet (GeldKarte-API).
Die Hersteller bieten allerdings eine breite Palette an Kartenterminals an, von

einfachen und billigen Versionen bis zu sicheren, aber teueren Varianten. In [41] wurden daher die Kartenterminals (Chipkartenleser) für den Heimbereich in vier Sicherheitsklassen eingeteilt.

- *Klasse 1: Chipkartenleser, der im wesentlichen eine Kontaktiereinheit darstellt.*
- *Klasse 2: Chipkartenleser, der entweder über eine eigene Tastatur verfügt oder zwischen PC-Tastatur und PC eingeschleift wird, zur Verhinderung des Ausspähens sensitiver Eingabedaten (z.B. PIN).*
- *Klasse 3: Chipkartenleser wie Klasse 2 mit Display zur manipulationssicheren Anzeige sicherheitsrelevanter Daten vor der Übergabe an die Chipkarte (z.B. Abbuchungsbetrag oder Daten, die signiert werden sollen).*
- *Klasse 4: Chipkartenleser wie Klasse 3 mit personalisiertem Sicherheitsmodul mit RSA-Funktionalität zur Authentisierung des Kartenlesers gegenüber anderen Komponenten mittels einer digitalen Signatur.*

Der ZKA hat festgelegt, dass, ausser in den Pilotprojekten, mindestens Klasse 3 Kartenterminals bei GeldKartenzahlungen zum Einsatz kommen dürfen. Es können also nur Kartenterminals verwendet werden, welche über eine separate Tastatur und ein Display verfügen. Erst dadurch ist der Bezahlvorgang auch sicher, denn so kann dem Kunden angezeigt werden (durch diverse Verfahren abgesichert), wieviel Geld an welchen Händler gezahlt werden soll. Durch die intergrierte Tastatur können Nutzereingaben nicht gefälscht und durch das separate Display Händleridentitäten nicht verändert werden.
Diese Kartenterminals müssen schnittstellenkompatibel zu dem verwendeten Computer sein. Die zur Zeit verfügbaren Systeme setzen dabei alle auf die serielle Schnittstelle, aber auch USB-Versionen sind geplant.
Leider sind solche Klasse 3 Kartenterminals noch recht teuer. Deswegen ist es nicht auszuschliessen, dass zukünftig eventuell doch wieder Klasse 2 Terminals zugelassen werden. Dies könnte notwendig werden, wenn den Kunden die hohe Sicherheit nicht so viel Geld wert ist, dass sie bereit sind, ein Klasse 3 Terminal zu kaufen.
Damit das Kartenterminal softwareseitig angesprochen werden kann, muss ein Treiber für das Betriebsystem des Rechners exisiteren und installiert sein. Es gibt dabei verschiedene Schnittstellenstardards am Markt, auf welche ich im nächsten

Kapitel genauer eingehen werde. Somit gibt es für jedes Kartenterminal einen spezifischen Treiber mit einer standardisierten Schnittstelle. Da diese Treiber essentiell für die Kommunikation mit dem Kartenterminal sind, werden z.T. auch Versionen für Betriebssysteme wie Linux oder Solaris angeboten.
Wenn das Kartenterminal angeschlossen und die Treiber installiert wurden, dann kann man auf das Kartenterminal und eine eventuell eingesteckte Chipkarte zugreifen. Für die GeldKartenzahlung fehlt allerdings noch eine weitere Softwarekomponente - die Bezahlsoftware. Bei einer solchen Zahlung kommuniziert das Händlersystem über diese Software mit dem Kundenterminalsystem. Aufgabe des Kundenterminalsystems ist es dabei, Befehle der Bezahlsoftware entgegenzunehmen, zu verarbeiten und zu beantworten.
Damit die Bezahlsoftware eine GeldKartentransaktion mit Hilfe des Kartenterminals abwickeln kann, müssen verschiedene Funktionen wie z.B. „Abbuchen einleiten“ oder „Abbuchen“ in der Kartenterminalsoftware aufgerufen werden, welche ihrerseits wiederum sicherheitskritische Funktionen in der GeldKarte aufrufen. Da die Aufrufe der Funktionen im Kartenterminal leider nicht vom ZKA standardisiert wurden, muss auf dem Computer des Kunden noch eine GeldKartenschnittstelle vorhanden sein, welche die Funktionsaufrufe der Bezahlsoftware in herstellerspezifische Kartenterminalkommandos umwandelt. Diese GeldKartenschnittstelle nennt man „GeldKarte-API“ oder auch einfach „GK-API“.

Verfügbare Kundenterminalsysteme:
Zum jetzigen Zeitpunkt (Juni 2001) haben zwei Kundenterminalsysteme mit Klasse 3 Kartenterminal die ZKA-Zertifizierung erhalten. Dies ist der „KAAN Professional“ der Firma Kobil (Zertifizierung November 2000) und die „CashMouse“ der Firma CpayS/Giesecke&Devrient (Zertifizierung März 2001). In naher Zukunft werden auch noch weitere Hersteller zertifizierte Klasse 3 Kundenterminalsysteme anbieten z.B. Towitoko oder Rainer SCT mit einem USB-Kartenterminal inklusive open source Kartentreiber- und GeldKartenschnittstelle.

Ich möchte jetzt die zwei bereits verfügbaren zertifizierten Systeme mit ihren Merkmalen vorstellen.

KAAN Professional:
Dieses Klasse 3 Kartenterminal besitzt eine Tastatur mit 4x4 Tasten (Zahlen und Funktionstasten wie Abbrechen, Bestätigen etc.) für die Eingabe von PIN-Kodes und Bearbeitung von Zahlungen. Die Anzeige wird über ein Display mit 2 Zeilen a 16 Zeichen gelöst. Dies ist leider etwas wenig, so dass längere Text nur mit Scrolling dargestellt und gelesen werden können.

Das Kartenterminal wird an den Serialport angeschlossen und unterstützt Transferraten zw. 9.600 und 115.000 Baud. Die Stromversorgung erfolgt wahlweise per separatem Netzteil oder über den PS/2-Port. Es sind zwei LEDs als zusätzliche Statusanzeige integriert.
Das Gerät ist in der Lage, abgesichert Upgrades zu laden. Damit können neue Anwendungen in alte Kartenterminals geladen werden, wodurch sich deren Funktionsumfang erhöht. Denkbare wäre z.B. die Erweiterung um ein Modul für die GeldKarteaufladung am privaten PC.
Der KAAN Professional bietet alle wichtigen Kartenterminaltreiberstandards an (CT-API, PC/SC, OpenCardFramework) und kann damit unter sehr vielen Betriebssystemen verwendet werden (Informationen von [42]). Das Gerät kostete 255 DM am 10.8.2001 im Webshop von Kobil.

CashMouse:
Auch die Cash-Mouse hat 16 Tasten und ein zweizeiliges Display je 16 Zeichen. 3 LEDs dienen der weiteren Betriebszustandsanzeige. Die Stromversorgung erfolgt über einen PS/2 Adapter. Es werden, wie beim KAAN Professional alle zur GeldKartenzahlung notwendigen Protokolle wie z.B. die ISO 7816 [43] angeboten. Der Anschluss erfolgt auch hier am Serialport bei Transferraten bis zu 115 KBd. Eine USB-Version ist in Planung.
Als Schnittstellen stehen bis jetzt CT-API und PC/SC zur Verfügung, allerdings nur für die Windows-Plattform. (Informationen aus CashMouse-Prospekt)
Dieses Gerät kostete 169 DM als Sonderangebot am 10.8.2001 im Webshop von innovationtrading.

Wie man erkennen kann, sind sich die beiden Systeme recht ähnlich. Allerdings ist das Schnittstellenangebot bei KAAN Professional schon weiter ausgebaut.

3.4 Notwendige Softwarekomponenten auf der Kundenseite

Wie unter Punkt 3.2.2 und 3.3 dargestellt, werden neben der Hardware auch noch 3 zueinander kompatible Softwarekomponenten auf dem Kundenrechner benötigt, damit eine GeldKartenzahlung abgewickelt werden kann. Diese sind:

- Bezahlsoftware (Teil des Händlersystems)

- GeldKartenschnittstelle GK-API (Teil des Kunderterminalsystems)
- Kartenterminaltreiberschnittstelle (Teil des Kundenterminalsystems)

Diese drei Softwarekomponenten sind bereits verfügbar, so dass eine GeldKartenzahlung im Internet bereits getätigt werden kann. Allerdings sind diese Softwarekomponenten nicht unter allen Betriebssystemen lauffähig. In der folgenden Tabelle 3.2 wird die Verfügbarkeit der notwendigen Softwarekomponenten für die Betriebssysteme Windows, Solaris und Linux aufgeführt (Stand Mai 2001).

	Windows	Solaris	Linux
Bezahlsoftware	X	-	-
GK-API (Kobil KAAN Pro)	X	-	-
GK-API (CashMouse)	X	-	-
Kartenterminaltreiber (Kobil KAAN Pro)	X	X	X
Kartenterminaltreiber (CashMouse)	X	-	-

Tabelle 3.2: Verfügbarkeit der notwendigen Softwarekomponenten

Wie man aus der Tabelle 3.2 erkennen kann, ist die GeldKartenzahlung zur Zeit nur unter Windows möglich, da unter den Unix-Betriebssystemen die Bezahlsoftware nicht funktioniert, und die GK-API fehlt. Zum jetzigen Zeitpunkt ist es noch ungewiss, wann die Kartenterminalhersteller auch für Linux und Solaris eine GK-API anbieten.
Weiterhin kann man erkennen, dass für das Kobil-Kartenterminal bereits auch unter Linux und Solaris Kartenterminaltreiber zur Verfügung stehen. Auch bei der CashMouse werden bald Treiber für diese Betriebssysteme erhältlich sein.
Die Internet-Händlersysteme würden eventuell auch für Linux und Solaris eine Bezahlsoftware anbieten, wenn die dafür notwendige GK-API vorhanden wäre.

Aufgrund dieser Gegebenheiten habe ich im Rahmen meiner Diplomarbeit eine GeldKartenschnittstelle für die Betriebsysteme Solaris und Linux geschreiben. Diese Schnittstelle wurde für den „KAAN Professional“ der Firma Kobil konzipiert, da dieses Gerät bisher als einziges auch Kartenterminaltreiber für zu Windows alternativen Betriebssystemen bereitstellt.
Bevor nach der Vorstellung des GeldKartechips die GeldKartenschnittstelle in Kapitel 5 genau dargestellt wird, möchte ich im folgenden Unterkapitel einige Sicherheitsaspekte bei der GeldKartenzahlung im Internet erläutern, da diese für das tiefere Verständniss notwendig sind.

3.5 Sicherheit

Wenn es um das Thema **Geld** geht, fällt meist im selben Atemzug das Wort **Sicherheit**. Im Interesse aller an der Zahlung beteiligten Parteien muss das Transaktionsschema möglichst vollkommen sicher sein. Gerade bei einer räumlich getrennten Zahlung unter Zuhilfenahme eines offenen Netzes, wie dem Internet, müssen verschiedene Vorkehrungen getroffen werden, um der recht grossen Anzahl an Betrugsmöglichkeiten entgegenzuwirken. So wollen Kunden nicht, dass unrechtmässig Geld von ihrer GeldKarte abgezogen wird. Händler müssen den Zahlungsbestätigungen des GeldKartesystems vertrauen können, um nicht fälschlicherweise Waren auszuliefern. Aber auch die Banken wollen ein sicheres System, da durch Betrugsfälle bei anderen Zahlungssystemen wie z.B. der Kreditkarte ein hoher finanzieller Schaden entsteht und ausserdem die Akzeptanz eines Zahlungssystems massgeblich von dessen Sicherheit abhängt.
Es wurden deshalb die folgenden Techniken angewendet, um ein recht sicheres Transaktionssystem zu schaffen:

- Zertifizierung der an der Zahlung beteiligten Soft- und Hardwarekomponenten
- Händlerauthentifikation mittels Signatur
- MAC-Authorisation
- Protokollierung
- weitere sicherheitstechnische Massnahmen

3.5.1 Zertifizierung

Damit die Transaktion für Kunden und Händler sicher ist, müssen Teile des verteilten Händlersystems sicherheitstechnisch geprüft und vom ZKA zertifiziert werden. Einer Zertifizierungspflicht unterliegen folgende Komponenten:

- Händlerterminal

- Internet-Händlersystem
- Kundenterminalsystem

Diese Zertifizierung soll garantieren, dass die in das Zahlungssystem eingebundene Hard- und Software den Sicherheitsanforderungen genügt.
Leider scheinen die Testverfahren bei der Zertifizierung nicht ausreichend zu sein. Die HTWK Leipzig hat beide zertifizierte Gerätetypen (KAAN Professional, CashMouse) erhalten und getestet. Dabei wurde gleich bei der ersten Testzahlung mit dem KAAN Professional einen Fehler festgestellt. Durch diesen Fehler war es möglich, dass von der GeldKarte Geld abgebucht wurde, obwohl die Zahlung nur im Browser bestätigt wurde. Weder die Händleridentität wurde auf dem Kartenterminaldisplay angezeigt, noch musste an der Kartenterminaltastatur die Zahlung bestätigt werden. Dieser Fehler kam zustande, weil das Internet-Händlersystem noch auf Klasse 1 Kartenterminals im Testmodus zugeschnitten war und das Kartenterminal das Abbuchen-Kommando des Internet-Händlersystem fälschlicherweise im sogenannten „Transparentmodus“ an die GeldKarte weitergeleitet hatte. Dieser unentdeckte grobe Fehler der Kartenterminalsoftware weckt kein grosses Vertrauen in einen Zertifizierungsnachweis vom ZKA. Ich vermute, dass die Zertifizierung nicht sehr gründlich vorgenommen wurde, damit möglichst schnell ein lauffähiges System am Markt vorhanden ist. Allerdings hoffe ich, dass die Tests in Zukunft sicherer und umfassender ausfallen, denn nur dann vertrauen auch die Kunden dem System.
Der Fehler im KAAN Professional konnte, nachdem ihn die HTWK Leipzig gemeldet hat, schnell mittels eines Firmware-Upgrades behoben werden und tritt in den später ausgelieferten Geräten nicht mehr auf.

3.5.2 Händlerauthentifikation

Da bei einer GeldKartenzahlung im Internet der Kunde nicht direkt am Kartenterminal des Händlers bezahlen kann, muss die überprüfte Händleridentität bei dem Bezahlvorgang auf dem Kundenterminal angezeigt werden.
Dazu werden sogenannte „Händlerauthentikationsdaten“ verwendet. Diese bestätigen mit Hilfe eines kryptographischen Schlüssels die Zuordnung einer Händlerkartennummer zu einem Händler nach dem asymetrischen RSA-Public

Key Verfahren. Die Verteilung und Verwendung dieser Schlüssel geschieht in der folgenden Art und Weise:

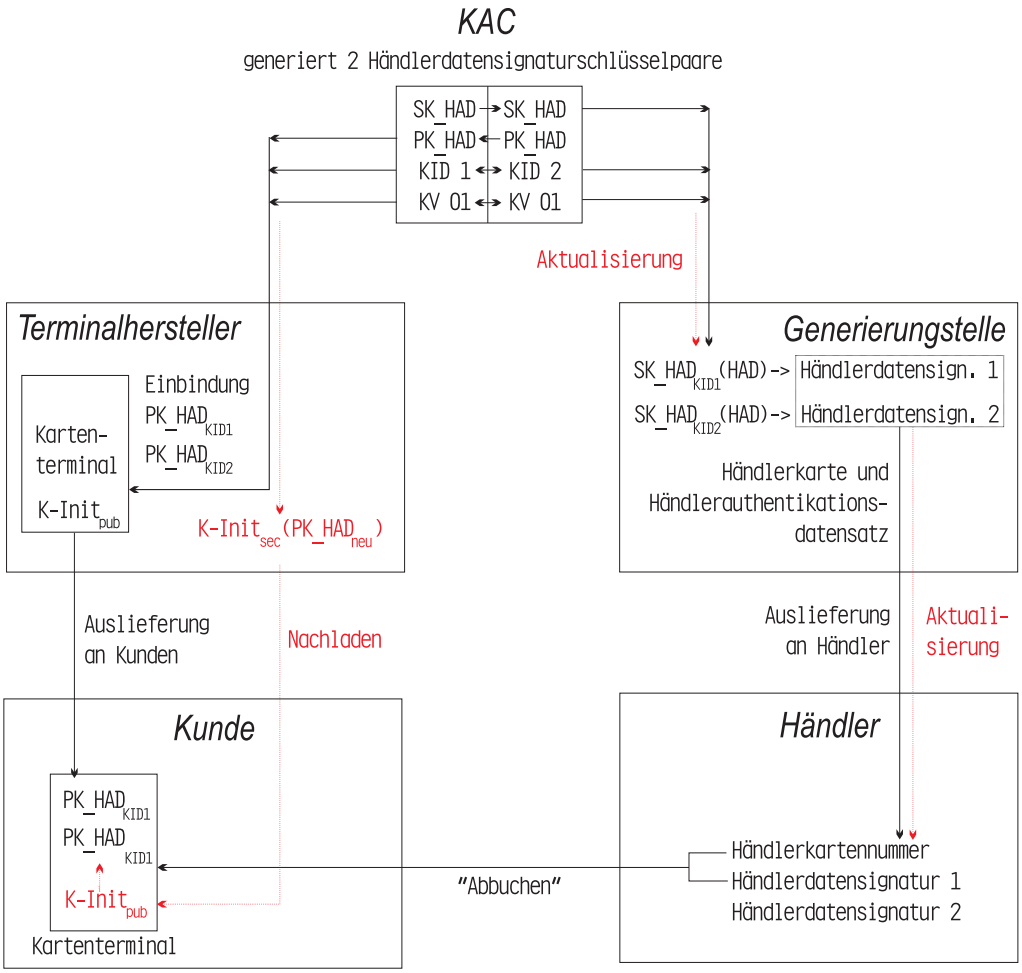

Abbildung 3.7: Schlüsselverwaltung

Schlüsselverteilung:
Die zur Überprüfung der Händlerdatensignatur notwendigen kryptographischen Händlerdatensignaturschlüssel werden von einer zentrale Einrichtung, dem Key Administration Center (KAC), generiert. Es werden jeweils 2 RSA-Schlüsselpaare, bestehend aus Secret- (SK_HAD) und Public-Key (PK_HAD), erstellt und mit

einer eindeutigen Versionsnummer (z.B. KV 01) sowie einer Key-ID (KID1 bzw. KID2) versehen.
Die öffentlichen Schlüssel (PK_HAD) beider Schlüsselpaare werden zusammen mit der Versionsnummer und der ID an die Kartenterminalhersteller sicher übermittelt. Diese integrieren diese Daten in die Kartenterminals, so dass die Schlüssel später zusammen mit der Hardware an den Kunden ausgeliefert werden.
Die geheimen Schlüssel (SK_HAD) werden vom KAC an eine Generierungsstelle weitergeleitet. Diese verschlüsselt damit für eine bestimme Händlerkarte mit einer eindeutigen Händlerkartennummer zwei Händlerauthentikationsdaten (HAD) und fasst die entstehenden Händlerdatensignaturen in einem Händlerauthentikationsdatensatz zusammen. Dieser Datensatz wird zusammen mit der Händlerkarte an dem Händler geliefert, welcher beide Komponenten in seine Paymentlösung integriert.

Überprüfung der Händleridentität beim Bezahlvorgang:
Nach der Schlüsselverteilung ist der Kunde im Besitz der öffentlichen Schlüssel und der Händler im Besitz der privaten Schlüssel. Wird bei einem Bezahlvorgang von der Händlerkarte das Kommando „Abbuchen“ erzeugt, so enthalten die an das Kartenterminal übergeben Daten unter anderem die Händlerkartennummer, die verschlüsselten Händlerauthentikationsdaten (Händlerdatensignatur) und die zur Entschlüsselung notwendige Key-ID und -version. Das Kartenterminal entschlüsselt mit Hilfe des zugehörigen öffentlichen Public-Key die übermittelte Händlerdatensignatur. Aus dem so erhaltenen DSI (Digital Signatur Input) können im Kartenterminal die Händlerauthentikationsdaten aufgebaut werden, welche unter anderem die ASCII-codierte Händleridentität und die Händlerkartennummern enthalten. Im Kartenterminal werden nun die Kartennummern mit der durch das Abbuchen-Kommando übertragenen Nummer überprüft. Bei Gleichheit der Kartennummern wurde der Händler ordnungsgemäss identifiziert und die ASCII-codierte Händleridentität wird auf dem Diplay des Kartenterminal angezeigt.
Somit kann der Kunde sicher sein, an wen er das Geld bezahlt.
Die Händleridentität wird vom Kartenterminal und nicht von der Chipkarte selbst überprüft, da noch kein RSA-Algorithmus in die GeldKarte integriert wurde.

Schlüsselwechsel:
Besteht der Verdacht auf Kompromitierung eines geheimen Händlerschlüssels, so kann dieser nicht mehr verwendet werden. Damit in diesem Fall ein Schlüsselwechsel (rote Teile in Abbildung 3.7) zügig erfolgen kann, wurden gleich zwei

Schlüsselpaare erzeugt. Der Händler braucht also nur die zweite Händlerdatensignatur (KID2) zu verwenden, um wieder eine sichere Zahlung zu gewährleisten. Das KAC kann in dieser Zeit eine neues Schlüsselpaar erzeugen, so dass die Generierungsstelle eine neue Händlerkarte und neu generierte Händlerauthentikationsdaten dem Händler zukommen lassen kann.
Der zweite Grund, einen Schlüsselwechsel durchzuführen, sind regelmässige Wechsel zur Erhöhung der Sicherheit. Der Händler bekommt dabei die Daten wieder über die Generierungsstelle. Damit auch das Kartenterminal die neu erzeugten öffentlichen Schlüssel abgesichert erhalten kann, muss das Gerät einen Mechanismus zum authentischen Nachladen von Schlüsseln besitzen. Dieser Mechanismus darf dabei kryptographisch nicht schlechter als der zur Überprüfung der Händleridentität verwendete RSA-Algorithmus sein. Das Kartenterminal könnte z.B. einen öffentlichen RSA-Schlüssel (K-Init pub) fest in das Kartenterminal integriert haben, so dass bei dem Schlüsselupdate die neuen Händlerschlüssel (PK_HAD neu) mit dem privaten Schlüssel des Kartenterminalherstellers (K-Init sec) verschlüsselt und an das Kartenterminal übertragen werden müssen.

3.5.3 MAC-Autorisation

Da man bei einer Internetzahlung sicherheitskritische Daten über ein offenes Netz versendet, muss deren Integrität sichergestellt werden. Dies bedeutet, dass der Empfänger der Nachricht in der Lage sein muss festzustellen, ob die Nachricht bei der Übertragung manipuliert wurde. Sender und Empfänger sind in diesem Fall die GeldKarte des Kunden und die Händlerkarte.
Dieser Schutz wird bei der GeldKarte durch Verwendung eines Message Authentication Code (MAC) realisiert, welcher eine Art elektronisches Siegel für eine Nachricht darstellt. Dieser MAC, auch KRD-Zertifikat genannt, wird zu der Nachricht mit Hilfe eines kryptographischen Schlüssels gebildet und dann zusammen mit der Nachricht an den Empfänger geschickt, welcher dann mit dem übertragenen MAC die Nachricht auf ihre Integrität verifizieren kann.
Die GeldKarte stellt dabei mehrere MAC-Varianten zur Verfügung, die alle einen 8 Byte langen MAC erzeugen. Bei Verwendung eines 8 Byte langen Schlüssels wird ein einfacher MAC berechnet, bei einem 16 Byte langem Schlüssel wird ein Retail MAC generiert. Beide können mittels DES-Algorithmus im CBC-Mode oder CFB-Mode kodiert werden.
Bei Verwendung eines dynamisierten 16 Byte langen Schlüssels wird ein Retail

CFB-MAC gebildet und mittels des Triple-DES im CBC-Mode kodiert. Die für Verschlüsselung und MAC-Bildung notwendigen kryptographischen Schlüssel der GeldKarte sind in speziellen Datenfeldern auf der Karte abgelegt und werden bei dem Bezahlvorgang ausgewählt.

3.5.4 Protokollierung

Damit die GeldKartenzahlung transparent wird, müssen Transaktionsdaten auch nach dem Bezahlvorgang zur Verfügung stehen. Auch wenn ein Fehler bei der Zahlung aufgetreten ist, muss der Fehlerkode für die spätere Fehlersuche vorhanden sein. Aus diesen Gründen wird die Zahlung auf der Kundeneinheit protokolliert. Dazu werden gesicherte Buchungssätze auf einem Datenträger gespeichert. Diese enthalten alle wichtige Daten über die Zahlung. Damit diese sicherheitskritischen Daten vor Manipulationen geschützt sind, wurden sie von der GeldKarte MAC-verschlüsselt. Möchte der Kunde eine Zahlung beanstanden, so kann dies an einem Bankensonderfunktionsterminal in seinem Kreditinstitut erfolgen. In diesem Terminal ist der zur MAC-Verifizierung notwendige private Schlüssel gespeichert, so dass das Kreditinstitut die zahlungsrelevanten Daten dekodieren kann. Aus diesem Grund muss es möglich sein, die Buchungsdatensätze auf Diskette zu kopieren oder auszudrucken.
Wenn bei der Zahlung ein Fehler aufgetreten ist, so muss versucht werden, einen Fehlerdatensatz auf einem Datenträger zu speichern. Dieser enthält unter anderem Informationen wie aktuelle Transaktionsphase, Fehlerquelle, Fehlerart und Fehlerkode. Auch weitere zahlungsrelevante Daten werden gespeichert, falls sie schon bekannt sind. Somit ist es leichter möglich, Fehlerquellen und -ursachen zu lokalisieren und zu beheben.

3.5.5 Weitere Sicherheitsmassnahmen

Neben den bereits vorgestellten Sicherheitsmechanismen wurden noch weitere Vorkehrungen getroffen, damit die GeldKartenzahlung sicher wird. Diese sind:

1. Verbot des Transparentmodus: Theoretisch ist es möglich, dass eine Software selbstständig Geld von einer GeldKarte abbucht, ohne dass der Kunde die Transaktion bestätigt hat. Dies wäre dann der Fall, wenn die Software

die Abbuchen-Kommandos direkt an die Chipkarte im sogenannten „Transparentmodus“ sendet. Damit solche Betrugsfälle vermieden werden, darf das Kartenterminal sicherheitskritische Kommandos, wie Abbuchen, Laden oder Rückbuchen, nicht von einer externen Software an die Chipkarte weiterleiten.
Einzig das Kartenterminal selbst darf diese Kommandos erzeugen und an die Karte senden. Da die Hardware zertifiziert wurde, kann der Kunde sichergehen, dass keine Software ungewollt Geld von der Karte abbucht.

2. Ablaufsicherheit im Kartenterminal: Ein Zahlungsablauf erfolgt in mehreren Schritten. Dabei ist genau festgelegt, welcher Schritt auf den aktuellen Zustand folgen kann. Somit entsteht ein Zahlungsautomat. Dieser gewährleistet, dass die Schritte in der korrekten Reihenfolge ausgeführt und Abweichungen der Schrittfolge als Fehler erkannt werden.

3. Verwendung von Sicherheitssiegeln: Das Kartenterminal ist ein sicherheitskritisches Gerät. Es hat diverse Schlüssel gespeichert und muss eine sichere Anzeige und Tastatureingabe gewährleisten. Damit die Integrität des Gerätes für den Kunden ersichtlich ist, muss es versiegelt oder so gebaut sein, dass ein unbefugtes Öffnen erkannt wird.

4. Verschlüsselte Kommunikation: Manche Internet-Händlersysteme verwenden einen SSL-Verschlüsselungskanal, damit der Datenstrom beim Versand über das Internet zusätzlich geschützt wird.

Kapitel 4

Der GeldKartechip

In diesem Kapitel wird der GeldKartechip näher vorgestellt. Es wird seine zugrundeliegende Spezifikation genannt, die logische Struktur erläutert und der Zugriff auf den Chip erklärt. Desweiteren wird auf die unterschiedlichen Kartentypen eingegangen, wobei in diesem Zuge auch das Problem der Euroumstellung und die daraus folgenden Auswirkungen erwähnt werden.

4.1 Spezifikation der GeldKarte

4.1.1 Chipkarte nach ISO/IEC 7816-Norm

Die GeldKarte ist eine sogenannte „Prozessorkarte“, welche der Norm ISO/IEC 7816 „Identification cards - Integrated circuit(s) cards with contacts“ [43] folgt. Diese Spezifikation besteht zur Zeit (August 2001) aus 10 Teilen:

- ISO/IEC 7816-1 : Physikalische Charakteristika der Karte
  In Teil 1 wird eine Chipkarte (Plastikkarte) nach Grösse (siehe Grafik 4.1), Aussehen, Biegefestigkeit und Resistenz gegenüber Umwelteinflüssen festgelegt.
- ISO/IEC 7816-2 : Dimensionen, Lage und Funktion der Kontakte werden in Teil 2 spezifiziert.

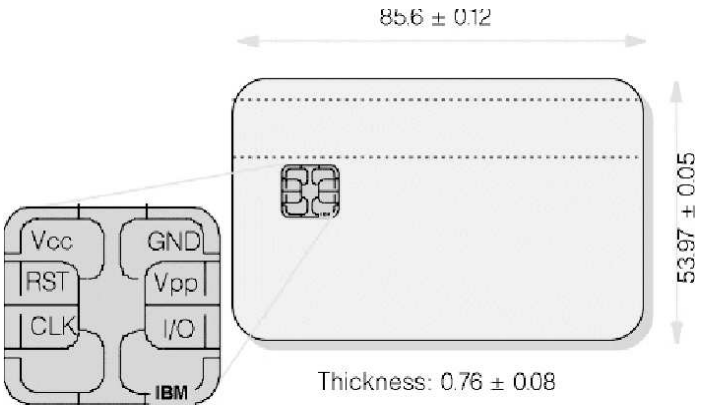

Abbildung 4.1: Chipkarte nach ISO 7816 [44]

Vcc=Versorgungsspannung
RST=Resetleitung
CLK=Takt
GND=Masse
Vpp=Programmierspannung
I/O=Datenleitung
Wie man erkennen kann, exisitert nur eine Datenleitung. Daher wird die serielle I/O im Halbduplexmodus betrieben. Die Chipkarte stellt dabei den Slave dar, der auf Kommandos des Kartenterminals, welches den Master darstellt, antwortet.

- ISO/IEC 7816-3 : In Teil 3 werden die elektronischen Signale und Übertragungsprotokolle festgelegt. Chipkarten bieten dabei normalerweise nur eines der beiden folgenden Protokolle an:
  - T=0, asynchrones, halb-duplex zeichenorientiertes Übertragungsprotokoll
  - T=1, asynchrones, halb-duplex blockorientiertes Übertragungsprotokoll

  Kartenterminals müssen natürlich beide Protokolltypen unterstützen. Bei der GeldKarte kommt laut ZKA das T=1 Protokoll zum Einsatz. Die serielle Kommunikation erfolgt über einen I2C-Bus (Beschreibung auf beiligender CD). Wird die Karte in den Slot des Kartenterminals gesteckt, so führt das

Kartenterminal eine Power-On-Reset-Sequenz auf der Chipkarte durch und erhält in dem zurückgelieferterten ATR (Answer To Reset) die Daten für den Kommunikationsaufbau.

- ISO/IEC 7816-4 : Standardisierte Kommandos zum Datenaustausch. Diese werden noch vorgestellt.
- ISO/IEC 7816-5 bis 10 : Die restlichen Teile behandeln weitere Zusatzkommandodefinitionen, standardisierte Datenelemente und mehr.

Die in dieser Norm spezifizierten Chips sind recht komplex organisiert. Sie besitzen einen Prozessor, ein Dateisystem, flüchtigen und nichtflüchtigen Speicher sowie einen Satz an Befehlen, welcher das Betriebssystem der Karte darstellt. Meist ist zusätzlich ein Kryptocoprozessor vorhanden. Für die Kommunikation kommt ein serielles Interface zum Einsatz. Es besteht sogar die Möglichkeit Applikationen, also neue Befehlssätze, nachzuladen. Solch ein Chip kann also durchaus als Mikrocomputer angesehen werden. In Bild 4.2 wird der schematische Aufbau einer Prozessorkarte ohne Kryptocoprozessor dargestellt.

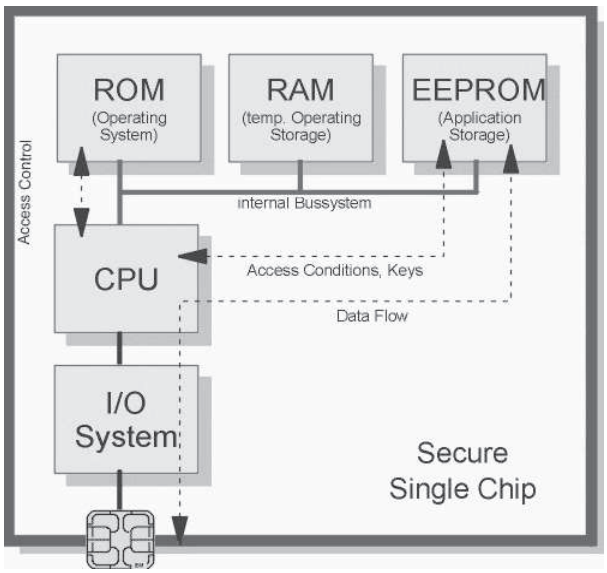

Abbildung 4.2: Schema einer Prozessorkarte [44]

4.1.2 Die zwei Varianten der ZKA-Chipkarte

Der ZKA hat auf Basis der vorgestellten ISO/IEC-Norm den GeldKartechip der ZKA-Chipkarte entwickelt. ZKA-Chipkarte nennt man eine Chipkarte, die der ZKA-Spezifikation „Datenstrukturen und Kommandos“ [45] folgt. Darin werden die Struktur der Daten, Sicherheitsarchitektur sowie die Standard- und Administrationskommandos des GeldKartechips beschrieben. Dieser Chip existiert in zwei Varianten:

1. Enthält der Chip nur die Zusatzapplikation „elektronische Geldbörse“, dann handelt es sich um eine kontoungebundene „weisse GeldKarte“ bzw. „Wertkarte“.
2. Enthält der Chip die Zusatzapplikationen „elektronische Geldbörse“ **und** „electronic cash“, dann liegt eine GeldKarte mit Kontobezug vor, welche auch „Börsenkarte“ genannt wird.

Man kann die ZKA-Chipkarten also in kontogebundene (Börsenkarten) und kontoungebundene Karten (Wertkarten) unterteilen. Beide Kartenarten enthalten die Applikation „elektronische Geldbörse“, welche für GeldKartenzahlungen benötigt wird. Eine guter Vergleich zwischen Börsen- und Wertkarte, auch im Hinblick auf die Anonymität, ist in der ct 11/98 [46] zu finden: *„Ein Vorteil der Geldkarte ist die Anonymität, die jedoch vom Geldkartentyp abhängig ist. Bei der Kombination aus EC- und Geldkarte handelt es sich um sogenannte Börsenkarten, die sich am Automaten durch eine Kontoabbuchung mit Bargeld aufladen lassen. Neben den Börsenkarten, die über 95 Prozent der Geldkarten stellen, gibt es nur ein paar hunterttausend Wertkarten. Dieser Geldkartentypus ist nicht an ein Konto gebunden und wird mit Bargeld am Bankschalter aufgeladen. Der Vorteil einer Wertkarte ist die maximale Anonymität: Bei einer Börsenkarte könnte die Bank Geldkartenzahlungen anhand von Ladevorgängen und Kartenschlüsseln zurückverfolgen, während die Wertkarte keinerlei Rückschlüsse auf den Kunden zulässt.“*

4.2 Problem Euroumstellung

Da am 1.1.2002 der Euro eingeführt wird, wurden schon im Vorfeld weitreichende Aktionen unternommen, den Zahlungsverkehr von DM auf Euro umzustellen.

Von diesen Veränderungen ist auch die GeldKartenzahlung im Internet betroffen. Die Ursache liegt darin begründet, dass, neben der Unterteilung in Börsen- und Wertkarten, zwei GeldKartentypen existieren, die mit unterschiedlicher Währung arbeiten. Zum einen gibt es die alten **Typ 0** GeldKarten, welche auf DM-Basis funktionieren und zum zweiten die neuen **Typ 1** GeldKarten, die den Euro als Währungsgrundlage haben. Typ 1 GeldKarten besitzen ausserdem unter anderem ein verbessertes Sicherheitskonzept.
Da zum jetzigen Zeitpunkt (2001) beide GeldKarten im Umlauf sind, müssen alle Zahlungskomponenten mit zwei Kartentypen arbeiten können. Dadurch entsteht z.B. bei einem Kartenterminalhersteller ungefähr der doppelte Aufwand bei der Programmierung der GeldKartenapplikation. Auch eine eventuelle Währungsumrechnung (Währungskennzeichen der Händlerkarte ist anders als das der Kunden-GeldKarte) muss vom Kartenterminal erledigt werden.
Da im Sommer 2001 aber immer noch Pilotprojekte laufen, und ab 1.1.2002 keine GeldKarten Typ 0 mehr im Umlauf sein werden, war dieser Aufwand der doppelten Integration im Bereich Internetzahlung im Prinzip nicht notwendig.

Ein Kartenterminal muss also mit vier verschiedenen Kartentypen/-arten umgehen können - Wertkarten Typ 0 und Typ 1 sowie Börsenkarten Typ 0 und Typ 1. Diese unterscheiden sich in den Kommando- bzw. Antwortnachrichten zum Teil erheblich.

4.3 Das Dateisystem einer ISO/IEC 7816-4 konformen Chipkarte

In diesem Unterkapitel soll das Dateisystem einer ISO/IEC 7816 Teil 4 konformen Chipkarte, welches auch bei der ZKA-Chipkarte Verwendung findet, erläutert werden.
Das Dateisystem einer Chipkarte ähnelt dem eines Computers. Es gibt zwei verschiedene Dateitypen:

- Dedicated File (DF) - Verzeichnis
- Elementary File (EF) - Datei

4.3.1 Aufbau des Dateisystems

Auf jeder Chipkarte exisitert mindestens ein Verzeichnis, das sogenannte Master File (MF). Dies kann man mit dem Root-Verzeichnis eines Computers vergleichen. In diesem Hauptverzeichnis können sich Dateien oder Unterverzeichnisse befinden. Jedes Unterverzeichnis kann, wie bei einem Computerdateisystem, weitere Unterverzeichnisse und Dateien enthalten. Die folgende Grafik 4.3 soll den möglichen Aufbau eines Chipkartendateisystems veranschaulichen.

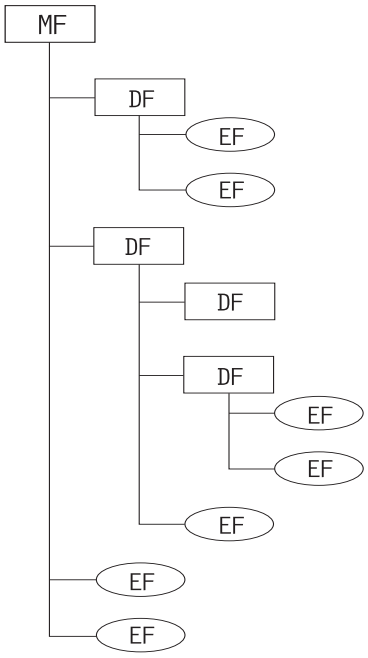

Abbildung 4.3: Dateisystem einer ISO/IEC 7816 Chipkarte

4.3.2 Referenzierung von Dateien und Verzeichnissen

Die Dateien und Verzeichnisse können mit einer von vier Methoden für den Zugriff selektiert werden:

1. Implizite Selektion, wenn das EF im aktuell verwendeten DF liegt.
2. Über einen 2 Byte langen „file identifier“, den jedes DF und EF besitzt.
3. Über den Pfad. Dieser kann im Root-Verzeichnis beginnen (MF-DF-DF-EF) oder im aktuell selektierten Verzeichnis (DF-DF-EF).
4. EFs können unter bestimmten Umständen auch über einen 5 Bit langen „short EF identifier“ (SFI) ausgewählt werden.

4.3.3 Elementarer Aufbau der Dateien

Die Dateien eine Chipkarte können in zwei Klassen geteilt werden. Zum einen gibt es transparente EFs. Diese besitzen keine besondere Struktur, sondern sind als reines Datenfeld zu verstehen. Die zweite Klasse stellen Dateien mit einer Recordstruktur (Datensatzstruktur) dar. Dabei gibt es drei Typen von Recordstrukturen:

1. Lineare EFs mit fester Datensatzgrösse
2. Lineare EFs mit variabler Datensatzgrösse
3. Zyklische EFs (Ring) mit fester Datensatzgrösse

Will man auf die einzelnen Records einer Datei zugreifen, so müssen diese referenziert werden. Dies kann über eine Recordnummer oder über einen „record identifier“ erfolgen. Bei einem Zugriff über die Recordnummer muss nur die Nummer des Datensatzes (Reihenfolge der Erstellung) angegeben werden. Bei linearen EFs besitzt der zuerst erstellte Datensatz die Nummer 1. Bei zyklischen Records besitzt der zuletzt erstellte Datensatz die Nummer 1.
Bei der Referenzierung mittels „record identifier“ werden Daten des Records zur Identifizierung verwendet. Dazu wird das erste Byte des Datenfeldes genutzt. Reicht dies für eine eindeutige Unterscheidung nicht aus, so werden weitere Daten herangezogen. Bei dieser Referenzierungsmethode muss zusätzlich eine logische Position angegeben werden. Diese spezifiziert, ob der gesuchte Record das erste oder letzte Auftreten bzw. das nächste oder vorherige Auftreten des „record identifier“ darstellt.

Will man auf transparente Dateien zugreifen, so werden dazu „data units“ adressiert. Diese sind üblicherweise 1 Byte gross. Transparente Dateien werden allerdings von der ZKA-Chipkarte noch nicht unterstützt.

Jede Datei besitzt eine „File Control Infomation “ (FCI). Diese enthält unter anderem die „File Control Parameters“, welche Informationen zu der Datei in TLV-Struktur bereitstellen. Diese Parameter besitzen auch ein „File Descriptor Byte“, welches den Dateizugriff, -typ und die verwendete EF-Struktur der Datei definiert. Die FCI können bei Auswahl der Datei mittels des Basic interindstry commandos SELECT FILE zurückgegeben werden.

4.4 Basic interindustry commandos

Jeder Chip, welcher nach ISO/IEC Norm 7816 Teil 4 entworfen wurde, bietet einen Grundbefehlssatz an. Dieser stellt die folgenden Befehle bereit:

- READ BINARY - aus Datei lesen
- WRITE BINARY - in Datei durch logische Verknüpfung schreiben
- UPDATE BINARY - Datei schreiben
- ERASE BINARY - Datei zum löschen freigeben
- READ RECORD(S) - Datensatz lesen
- WRITE RECORD - Datensatz schreiben
- APPEND RECORD - Datensatz an Datei anhängen
- UPDATE RECORD - Schreiben eines Datenobjektes
- GET DATA - Datenobjet lesen
- PUT DATA - Schreiben eines Datenobjektes durch logische Verknüpfung
- SELECT FILE - Datei selektieren
- VERIFY - PIN Vergleich

- INTERNAL AUTHENTICATE - Authentifizierung der Chipkarte durch das Kartenterminal
- EXTERNAL AUTHENTICATE - Authentifizierung des Kartenterminals durch die Chipkarte
- GET CHALLENGE - Zufallszahl erzeugen
- MANAGE CHANNEL - logischen Kanal der Chipkarte steuern

Diese Befehle stellen Sicherheitsfunktionen dar oder dienen der Dateimanipulation. Die genaue Befehlsbeschreibung ist in der ISO/IEC 7816-4 zu finden.

4.5 Die Applikation „elektronische Geldbörse“

Wie bereits in 4.1.2 erläutert, benötigt eine ZKA-Chipkarte die Applikation „elektronische Geldbörse“ [47], damit sie die GeldKartenfunktionalität anbieten kann. Eine Applikation ist dabei als Sicht auf die Dateien der ZKA-Chipkarte zu verstehen. Wurde eine Applikation selektiert, so stehen die Applikationskommandos dem Kartenterminal zur Verfügung und die applikationseigenen Dateien und Verzeichnisse sind sichtbar. Auch die short EF identifier sind erst jetzt gültig. Die Karte befindet sich zusätzlich in einem entsprechenden Applikationskontext. Zur internen Ablaufkontrolle muss die ZKA-Chipkarte diesen aktuell verwendeten Applikationskontext speichern.
Damit eine Applikation genutzt werden kann, muss das zugehörige Applikationsverzeichnis (ADF - Application Dedicated File) mittels des Standardkommandos SELECT FILE ausgewählt werden.
Das ADF der Applikation „elektronische Geldbörse“ besitzt bei Typ 0 GeldKarten den Namen DF_BÖRSE, und bei Typ 1 GeldKarten den Namen DF_BÖRSE_NEU. Die Selektion erfolgt über die, vom ZKA spezifizierte, Application-ID (AID). War die Selektion erfolgreich, so bietet die ZKA-Chipkarte, in diesem Fall eine kontobezogene Börsenkarte, folgende Dateien an:

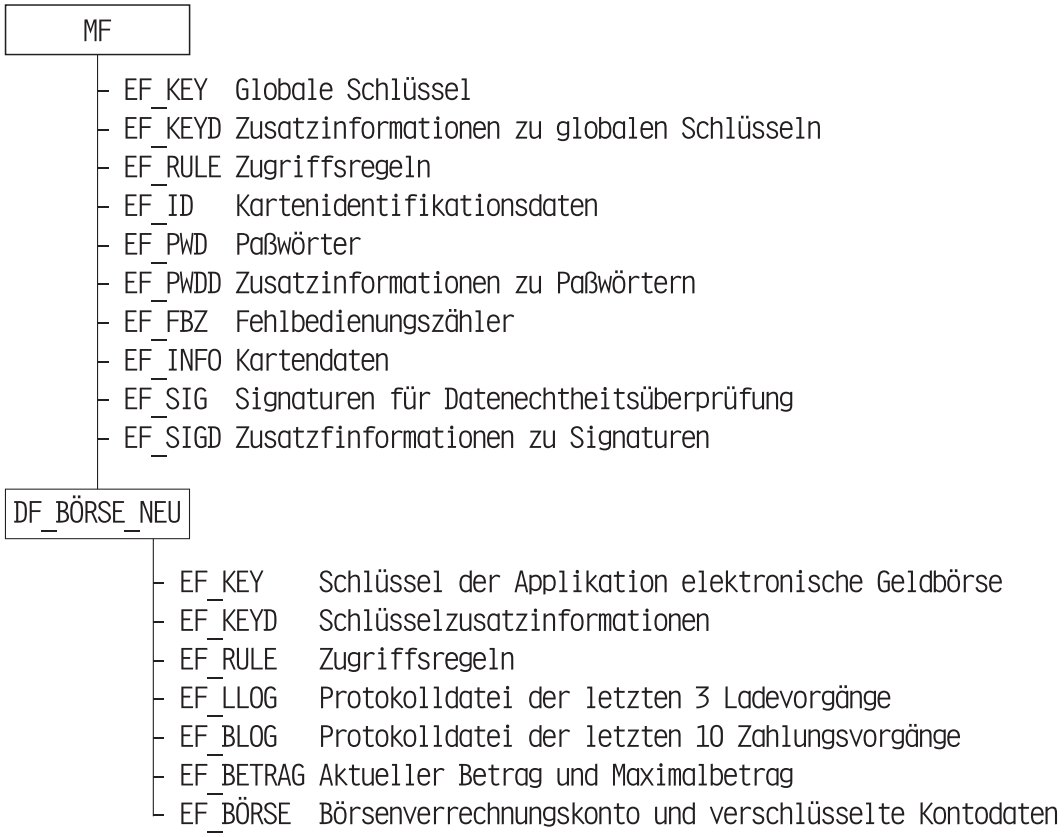

Abbildung 4.4: Dateien der „elektronische Geldbörse“ einer Börsenkarte

Kontofreie Wertkarten besitzen im Gegensatz zu Börsenkarten keine EF_PWD, EF_PWDD, EF_FBZ und EF_INFO Dateien im MF.

Neben diesen für die GeldKartenzahlung notwendigen Dateien bietet die Applikation „elektronische Geldbörse“ zahlungsrelevante Ergänzungskommandos an, welche in Tabelle 4.1 aufgeführt werden. Diese Ergänzungskommandos existieren zum Teil in zwei Varianten. Die erste leitet einen Vorgang ein, während die zweite Variante den Vorgang ausführt.

Ergänzungskommando	Funktion(en)
LADEN	Laden einleiten, Betrag in Geldbörse laden
ENTLADEN	Entladen einleiten, gesamten Betrag auf Kundenkonto buchen
ABBUCHEN	Abbuchen einleiten, Betrag abbuchen
RÜCKBUCHEN	Letzten Abbuchungsbetrag zurückbuchen
ANTWORT WIEDERHOLEN	Prüfung, ob vorheriges Kommando korrekt ausgeführt wurde

Tabelle 4.1: Ergänzungskommandos der Applikation „elektronische Geldbörse“

Bemerkung: Wertkarten werden nicht ENTLADEN, da das Geld nicht auf ein Konto gebucht werden kann. Aus diesem Grunde wird bei Wertkarten am Bankschalter durch ein ABBUCHEN-Kommando der Restbetrag von der Karte entfernt und bar ausgezahlt.

4.6 Kommunikation zwischen Kartenterminal und GeldKartechip

Bei einer GeldKartenzahlung muss das Kartenterminal in einer bestimmten, vom ZKA festegelegten, Reihenfolge ISO-Standardkommandos sowie Ergänzungskommandos der Applikation „elektronische Geldbörse“ mit den korrekten Parametern aufrufen. In diesem Unterkapitel soll daher die Kommunikation zwischen Kartenterminal und GeldKartechip betrachtet werden.

Die Kommandos (Standard- und Ergänzungskommandos) einer ISO/IEC 7816-Chipkarte bestehen aus einer fest vorgegebenen Struktur, welche Kommando-APDU (command-APDU) genannt wird. Auch die Antwortdaten der Chipkarte werden in einer APDU-Struktur verpackt, welche Antwort-APDU (response-APDU) genannt wird. Beide Strukturen wurden als einfache Bytefolge mit bestimmten Kommando- und Antwortkodes realisiert. Wenn man also eine Kommando-APDU an die Chipkarte überträgt, so wird einfach eine Bytefolge in einem festgelegten Format übergeben. Nach Kommandoabarbeitung wird eine Antwortbytefolge zurückgesandt. Die Abbildung 4.5 zeigt den Aufbau beider APDU-Strukturen.

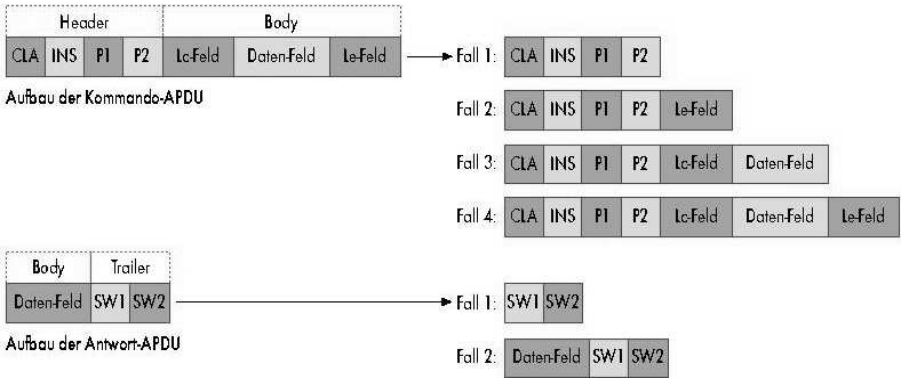

Abbildung 4.5: APDU-Aufbau [48]

Eine Kommando-APDU besteht aus den Teilen **Header** und **Body**. Dabei beschreibt der Header das auszuführende Kommando, so dass er essentiell notwendig ist. Der optionale Body enthält Daten, die an das Kommando übergeben werden, und ist demnach unterschiedlich lang.
Der Header besteht immer aus den 4 Bytes: CLA, INS, P1 und P2. CLA gibt die Klasse der auszuführenden Instruktion an. INS spezifiziert die Instruktion, der mit CLA ausgewählten Klasse, womit der Befehl festgelegt ist. P1 und P2 sind die Parameter des ausgewählten Kommandos und können auch „00h“ betragen.
Der Body kann 4 verschiedene Formen besitzen:

- Fall 1: Der Body ist leer. Es wird nur das, durch den Header spezifizierte Kommando, ausgeführt, ohne dass Daten zurückgegeben werden.
- Fall 2: Der Body besteht aus dem 1 Byte grossen Längenfeld Le (length expected). Somit wird das Kommando ausgeführt und Le-Bytes zurückgeliefert bzw. werden alle Bytes zurückgegeben, falls Le=00h beträgt.
- Fall 3: Der Body enthält das 1 Byte grosse Längenfeld Lc (length commanddata), in welchem die Länge in Byte des nachfolgenden Datenfeldes steht, und das Lc-grosse Datenfeld selbst. Es wird also das Kommando des Headers ausgeführt, wobei dem Kommando das Längen- und Datenfeld übergeben werden. Es werden keine Daten zurückgegeben.
- Fall 4: Wie Fall 3, nur dass sich nach dem Datenfeld noch das Längenfeld Le der erwarteten Daten befindet. Somit werden nach Ausführung des

Kommandos Le-Bytes oder alle Daten (Le=00h) zurückgegeben.

Die Antwort-APDU ist wesentlich einfacher aufgebaut. Sie besteht aus einem optionalen **Body**, welcher die Antwortdaten enthält und einem **Trailer**, welcher aus zwei Resultatbytes besteht, die über Erfolg oder aufgetretene Fehler informieren. Es gibt zwei Typen von Antwort-APDUs:

- Typ 1: Es wird nur der Trailer zurückgeliefert. Dies ist der Fall, wenn keine Daten erwartet wurden, keine Daten zurückgegeben werden oder ein Fehler aufgetreten ist.
- Typ 2: Es wird der Body mit den Antwortdaten und daran anschliessend der Trailer zurückgegeben.

Der Trailer informiert über das Resultat der Ausführung des Kommandos. Dabei existieren die Fehlergruppen:

- Korrekte Ausführung des Kommandos, kein Fehler
- Warnung bei der Ausführung
- Fehler bei der Ausführung
- Fehler in der Kommando-APDU

Die Kodes im Body, der Aufbau des Headers und Bodyinhalt der Antwort-APDU der Standardkommandos werden von der ISO/IEC-Norm 7816 festgelegt. Die Kodes und Header der Ergänzungskommandos sowie deren Antwortdatenbody wurden vom ZKA unter Beachtung der ISO-Norm spezifiziert.

4.7 Sicherheitsmechanismen der ZKA-Chipkarte

Die Sicherheitsmechanismen einer ZKA-Chipkarte werden unter anderem in [49] und [45] genauer erläutert.

Beschränkter Zugriff:
Die ZKA-Chipkarte enthält viele vertrauliche Daten wie Schlüssel und Passwörter. Auf diese sicherheitskritischen Daten dürfen daher nur bestimmte Komponenten unter definierten Bedingungen zugreifen. Wichtig dabei ist, dass auch einige Ergänzungskommandos der Applikation „elektronische Geldbörse“, wie z.B. das finale Abbuchen, nur vom Kartenterminal an die Chipkarte gesendet werden dürfen. Das Kartenterminal muss also dafür Sorge tragen, dass es diese Kommandos nicht im Transparentmodus von einer externen Software an die Chipkarte weiterleitet, sondern abweist.

Integrierte kryptographische Verfahren:
Neben diesem Zugriffsschutz bietet die ZKA-Chipkarte diverse kryptographische Sicherungsverfahren an, welche einen chipkarteninternen Zufallsgenerator benutzen können. Die Kryptofunktionen werden von einem speziellen Kryptoprozessor, der mit im Chip integriert ist, bereitgestellt. Die angeboten Verfahren sind:

- Authentisierung (Identitätsnachweis):
  - Karte gegenüber Aussenwelt
  - Aussenwelt gegenüber Karte
  - gegenseitige Authentisierung
- MAC-Sicherung (Sicherung der Integrität übermittelter Nachrichten)
- symmetrische Verschlüsselung nach DES oder Triple-DES (Vertraulichkeit der Daten) mit:
  - nicht-flüchtig gespeicherten Schlüsseln
  - flüchtig gespeicherten, abgeleiteten Schlüsseln
  - ausgehandelten Schlüsseln (Session Keys)
- asymmetrische Verfahren wie RSA sind in Planung

Sicherheitszustand:
Die Chipkarte oder ein Verzeichnis befinden sich in einem Sicherheitszustand. Diese Zustände können sich nur nach Authentisierung der externen Welt, durch Passwörter oder Schlüssel, verändern. Bei der GeldKartenzahlung wird so z.B.

sichergestellt, dass kein Abbuchen-Kommando ohne ein vorheriges Abbuchen-einleiten-Kommando ausgeführt wird. Durch die definierten Zustandswechsel ist es also möglich, Automaten der erlaubten Programmabläufe zu erstellen.

Dateischutz:
Jede Datei besitzt Zugriffsbedingungen, welche für einen Zugriff erfüllt sein müssen. So kann verlangt werden, dass die Nachricht zum Auslesen einer Datei in einer bestimmten Weise abgesichert ist (z.B. MAC-gesichert). Eine andere Möglichkeit bieten die Sicherheitszustände. So kann auf bestimmte Dateien nur dann zugegriffen werden, wenn ein gewisser Sicherheitszustand vorliegt.

Sicherheitsumgebungen:
Die vorgestellten Sicherungsmechanismen können nur lokal für ein Verzeichnis oder global für den kompletten Chip aktiv sein. Bei lokalen Sicherungsmechanismen besitzt das Verzeichnis eigene Schlüssel oder Passwörter, die zur Anwendung kommen. In sogenannten Sicherheitsumgebungen (Security Environments) wird spezifiziert, welche Schlüssel oder Passwörter ein Sicherheitsmechanismus benutzt, welche Zugriffsregeln angewandt werden und welches Datenformat für Übermittlungen verwendet wird. Ein Verzeichnis kann zwar mehrere solcher Security Environments besitzen, allerdings ist zu einem bestimmten Zeitpunkt immer nur eine dieser Sicherheitsumgebungen aktiv.

4.8 Weitere Informationen

Die in diesem Kapitel dargestellten Informationen haben das Thema Chipkarte im Allgemeinen und GeldKarte im Speziellen nur grob abgehandelt, damit dem Leser die Problematik der Chipkartenprogrammierung bekannt ist, bevor im nächsten Kapitel die GeldKartenschnittstelle vorgestellt wird. Wer sich mit Chipkarten genau auseinandersetzen will, sollte sich die ISO/IEC 7816 Norm [43] zumindest in Teilen besorgen. Teil 1-3 befinden sich als ASCII-Version auf der beiligenden CD. Für das Thema GeldKarte ist die ZKA-Spezifikation essentiell (Bezugsadresse siehe Anhang). Sehr interessant ist die „EMV 96 Integrated Circuit Card - Specification for Payment Systems“. Diese behandelt elektromechanische Charakteristiken, logisches Interface, Übertragungsprotokolle, Datenelemente und Kommandos, Applikationsauswahl und Sicherheitsaspekte für Chipkarten in Zahlungssystemen. Die Spezifikation ist auf der beiligenden CD zu finden.

Kapitel 5

Die GK-API als Schnittstelle zwischen Internet-Händlersystem und Kartenterminal

Dieses Kapitel beschreibt eine GeldKartenschnittstelle für das Kartenterminal „KAAN Professional“ der Firma Kobil. Die Schnittstelle wurde von mir neben einer Windowsversion auch für die Betriebssysteme Solaris und Linux entwickelt. Sie stellt damit die erste exisiterende GeldKartenschnittstelle für diese beiden Betriebssysteme dar und soll mit dafür sorgen, dass die GeldKartenzahlung bald auch auf diesen Plattformen möglich ist.

5.1 Grundinformation zu der GK-API

Damit verschiedene Internet-Händlersysteme (incl. Händlersystem-spezifischer Bezahlsoftware) mit diversen Kartenterminals, welche eine individuelle Befehlskodierung besitzen, kommunizieren und so eine GeldKartenzahlung abwickeln können, bedarf es einer standardisierten Schnittstelle. Diese Schnittstelle wird GeldKarte-API oder kurz GK-API genannt.
Sie gehört zum Softwareumfang des Kundenterminalsystem und bietet alle Funktionen, die für eine GeldKartenzahlung notwendig sind, nach aussen hin an.

Der ZKA hat spezifiziert, welche Schnittstellenfunktionen die GK-API zur Verfügung stellen muss, wie sie aufgerufen werden und was diese zu leisten haben.

Wie bereits in Kapitel 3.4 erwähnt wurde, existieren bis jetzt die GeldKarteschnittstellen nur für das Betriebssystem Windows. Andere Betriebssysteme, wie Linux, Solaris oder OS2, werden noch nicht unterstützt. Aus diesem Grund habe ich GK-API zu schreiben, welche auch auf den Betriebssystemen Solaris und Linux lauffähig ist, so dass in Zukunft für diese Systeme eine GeldKartenschnittstelle angeboten werden kann.

Abbildung 5.1: GK-API als Schnittstelle

Die GK-API wird als dynamisch ladbare Bibliothek (shared library) umgesetzt. Dies hat folgende Gründe:

- Die GK-API muss einfach aufzufinden sein. Da es eine shared library ist, braucht der Nutzer sie nur in das Systemverzeichnis für Bibliotheken zu kopieren bzw. den Bibliothekspfad korrekt zusetzen. Danach kann die Bezahlsoftware die GK-API einfach durch Anfordern der Bibliothek „gkapi“ laden. Das jeweilige Betriebssystem sucht die Bibliothek dann eigenständig im richtigen Verzeichnis.
- Ein zweiter Grund ist die einfache Kommunikation mit einer Bibliothek. Nachdem die Bezahlsoftware die Bibliothek geöffnet hat, kann sie einfach über Funktionsaufrufe mit ihr kommunizieren. Über den Austausch von Zeigern auf Strukturen und Felder können auch komplexe Daten übergegeben werden werden.

Die Bibliothek wird für das zertifizierte Klasse 3 Kartenterminal „KAAN Professional“ in der Programmiersprache C geschrieben. Die Entscheidung fiel auf den KAAN Professional, weil für dieses Gerät bereits die Kartenterminaltreiberschnittstelle für Solaris und Linux existiert. C wurde gewählt, da die Header-Dateien der Treiberschnittstelle für diese Sprache verfügbar waren.

5.1.1 Notwendige Tools und Spezifikationen

Damit man eine GeldKartenschnittstelle entwickeln kann, benötigt man einige Informationen und Programme, die man im Vorfeld besorgen sollte. Nachfolgend werden die wichtigsten Entwicklungstools und Spezifikationen vorgestellt.

Wie bereits erwähnt wurde, hat der Zentrale Kreditausschuss die GK-API spezifiziert. An diese Spezifikation muss man sich bei der Programmierung halten, da sonst die Schnittstelle keine Zertifizierung erhält und somit nicht eingesetzt werden kann. Die „Schnittstellenspezifikationen der ZKA-Chipkarte“ kann direkt beim ZKA (Adresse siehe Anhang) für eine Schutzgebühr von zur Zeit 400 DM bestellt werden. Dafür erhält man eine CD mit der gesamten Spezifikation zur ec- und GeldKarte.
Ausserdem benötigt man die Spezifikation des verwendeten Kartenterminaltreibermodells. In diesem Fall wurde [50] verwendet. Weitere wichtige Spezifikationen sind im Anhang aufgeführt.

Damit die erstellte GK-API getestet werden kann, benötigt man ein Testsystem. Dafür kann man ein funktionsfähiges Internet-Händlersystem benutzen. Es existieren Testshops, bei denen auch Pfennigartikel gekauft werden können, wodurch sich der finanzielle Verlust bei Testkäufen in Grenzen hält. Leider funktionieren die bisherigen Lösungen nur auf Windows-Systemen, so dass man z.B. unter Linux kein Internet-Händlersystem zum Testen besitzt. Desweiteren bietet zu Zeit kein Händlersystem die Funktionalitäten des (Schnellen) Inkrementellen Abbuchens an, so dass auch diese nicht getestet werden können.
Die zweite Möglichkeit ist die Verwendung eines speziellen Testprogrammes. Dieses heisst „bssim“ und kann bei der VÖB-Zvd Bank für Zahlungsverkehrsdienstleistungen (Adresse siehe Anhang) für 150 DM bestellt werden, wobei der Support inklusive ist. Mit der bssim-Software kann man Zahlungen mittels editierbarer Scripte simulieren. Somit können beliebige Testfälle generiert werden, so dass man z.B. leichter in der Lage ist, Fehlzahlungen zu simulieren. Im Gegensatz zum Test mit realen Internet-Händlersystemen kann mit der bssim-Software auch das (Schnelle) Inkrementelle Abbuchen überpürft werden. Die Lauffähigkeit wird nur bei Verwendung von Windows NT 4.0 als Testbetriebssystem garantiert, allerdings scheint das Programm auch unter Windows 95/98 problemlos zu funktionieren. Leider gibt es auch von der bssim-Software keine Version für andere Betriebssysteme. Dies ist ein grosses Hemmniss bei der Entwicklung der GK-API für andere Plattformen.

Damit die bssim-Software richtig eingesetzt werden kann, benötigt man zusätzlich Test-GeldKarten und Händlerkartenschlüssel. Die GeldKarten kann man für 50 DM pro Stück ebenfalls von der VÖB-Zvd Bank für Zahlungsverkehrsdienstleistungen beziehen. Die entsprechenden Händlerkartenschlüssel bekommt man bei entsprechenden Voraussetzungen von der Bank-Verlag GmbH (Adresse siehe Anhang).

5.2 Die GK-API als Vermittler zwischen Bezahlsoftware und Kartenterminal

Wie bereits aufgezeigt, wird eine standardisierte GeldKartenzahlungsschnittstelle benötigt, damit ein Internet-Händlersystem über ein Kartenterminal und der darin eingesteckten GeldKarte eine Zahlung abwickeln kann.
Dazu Grafik 5.2, welche die auf Kundenseite an einer Zahlung beteiligten Hard- und Softwarekomponenten sowie die Kommunikation zwischen diesen Komponenten verdeutlichen soll.

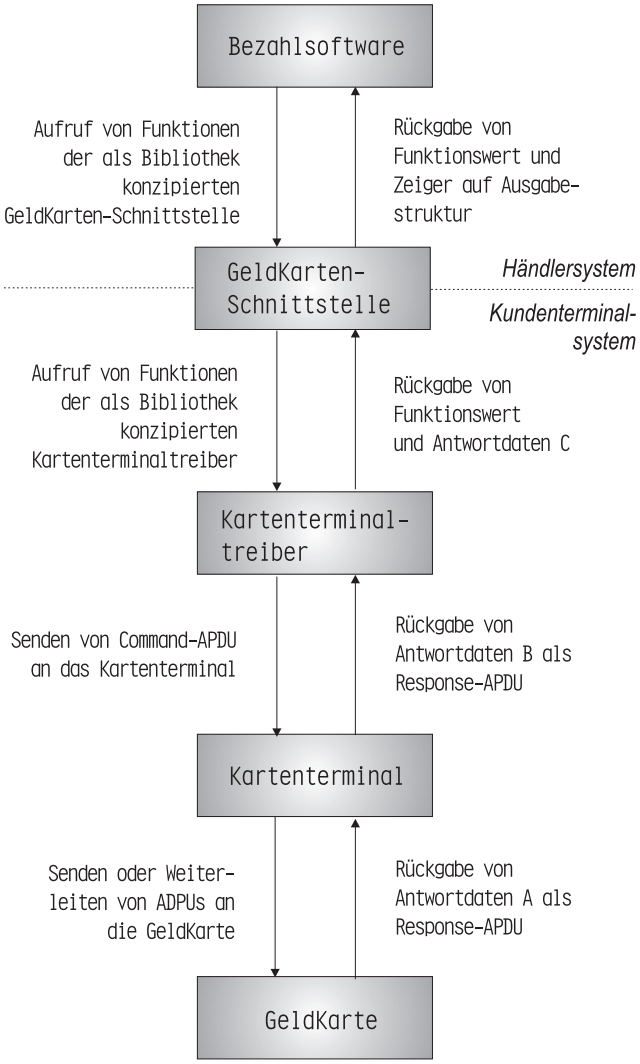

Abbildung 5.2: Kommunikationskette

Wie man erkennen kann, handelt es sich um eine 2-Wegeverbindung. Zum einen von Bezahlsoftware über diverse Komponenten zur GeldKarte und zum zweiten von der GeldKarte wieder über die selben Komponenten zurück zur Bezahlsoftware. Dabei funktioniert die Kommunikation im Halbduplexmodus, d.h. es wird von jeder Komponente jeweils eine Nachricht versandt und dann auf eine Antwort gewartet.

Internetzahlungen werden immer vom Internet-Händlersystem angestossen und kontrolliert. Die Bezahlsoftware ist der Teil auf dem Internet-Terminal, der zum Internet-Händlersystem gehört und der deshalb die führende Rolle auf dem Internet-Terminal einnimmt.
Wenn die Bezahlsoftware einen Zahlungsvorgang initiert, so ruft sie in fest vorgeschriebener Reihefolge Kommandos der GeldKartenschnittstelle auf. Der Schnittstelle erhält dabei verschiedene Eingabedaten, welche teils von der Bezahlsoftware und teils direkt von der Händlerkarte stammen. Aufgabe der GK-API ist es, aus den übergebenen Eingabedaten Kommandonachrichten (command-APDUs) zu generieren und mit Hilfe eines Kartenterminaltreibers (KT-Treiber) an das Kartenterminal zu senden. Dies ist notwendig, da, wie bei den Sicherheitsmechanismen in Kapitel 3.5 und 4.7 beschrieben, nur das Kartenterminal sicherheitskritische Kommandos an die GeldKarte senden darf. Die Bezahlsoftware kann also nicht selbst Geld abbuchen, sondern nur den Vorgang veranlassen.
Aus diesem Grund enthält jedes für die GeldKartenzahlung ZKA-zertifizierte Kartenterminal einen Befehlssatz an KT-Kommandos, dessen Befehle von der GK-API über die generierten Kommandonachrichten aufgerufen werden können. Hinter jedem dieser Aufrufe verbirgt sich auf Seiten des Kartenterminals ein kleines Programm, welches, wie bereits in 4.6 erklärt, auch über APDUs vielfältige Kommandos von der GeldKarte ausführen lässt und aus den Antwortdaten der Karte eine Antwortnachricht (Response-APDU) für die GK-API erstellt.
Leider hat es der ZKA versäumt, die Kommando-APDUs des Kartenterminals für die GeldKartenzahlung festzulegen und das Antwort-APDU-Format zu spezifizieren. Somit kann jeder Hersteller eigene Kommandokodes für die Aufrufe der Zahlungsfunktionen festlegen. Dafür werden die nach ISO 7816-4 frei verwendbaren Kodes benutzt. Auch die Antwortdaten werden in einem herstellerspezifischen Format vom Kartenterminal zurückgeliefert.
Damit eine Bezahlsoftware nicht Kommandokodes und Auswertungsroutinen für alle zertifizierten Geräte beinhalten muss, wurde die GK-API als Standardschnittstelle spezifiziert, welche allgemeingültige Kommandos der Bezahlsoftware in herstellerspezifische Kommandos des Kartenterminal umwandelt. Die Schnittstelle hat somit hauptsächlich die Aufgabe einer Übersetzungsfunktion. Allerdings wurde die GK-API auch entworfen, damit eine Bezahlsoftware unabhängig von den in diesem Kapitel vorgestellten Treibermodellen ist.
Theoretisch wäre auch eine standardisierte GK-API möglich. Dafür müssten sich allerdings die Kartenterminalhersteller auf feste CLA und INS-Kodes der command-APDUs zum Aufruf der Zahlungsfunktionen im Kartenterminal, sowie auf standardisierte response-APDUs vom Kartenterminal an die GK-API einigen.

Für eine standardisierte GeldKartenschnittstelle ist es zum jetzigen Zeitpunkt wahrscheinlich bereits zu spät, da die Entwicklungen schon zu weit fortgeschritten sind und somit umfangreichere Firmwareupdates notwendig wären.

In der folgenden Grafik 5.3 soll noch einmal der Kommunikationsablauf des Kommandos „Kartendaten lesen“ (READ CARD DATA) die Rolle der GK-API als Bindeglied zwischen Bezahlsoftware und Kartenterminal genauer verdeutlichen.

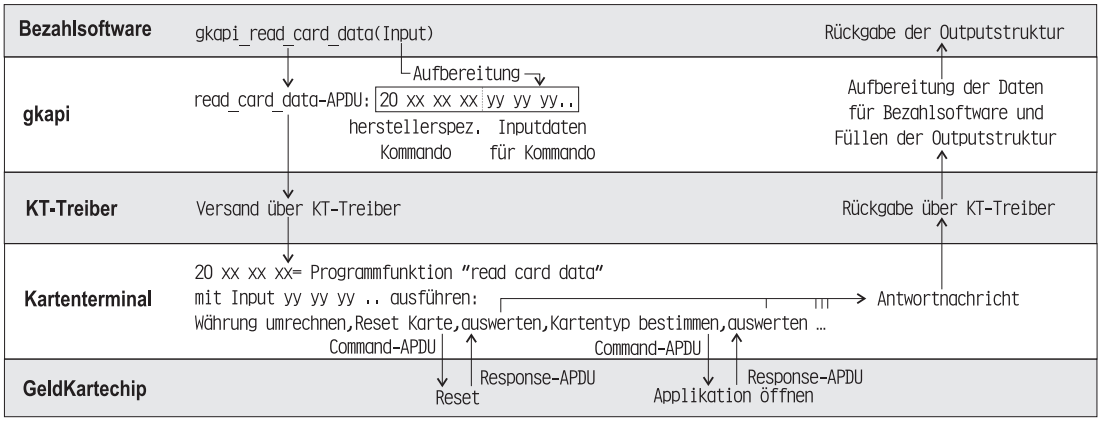

Abbildung 5.3: Kommunikationsablauf

5.3 Kommunikation zwischen Bezahlsoftware und GK-API

Damit die Bezahlsoftware mit Hilfe der GeldKartenschnittstelle eine Zahlung abwickeln kann, werden spezifizierte Funktionen in der Bibliothek aufgerufen.
Dazu muss die Bibliothek als erstes von der Bezahlsoftware geöffnet werden. Dieser sicherheitskritische Zugriff auf die Software des Kunden-PC erfordert zumindest eine einmalige Bestätigung von Seiten des Kunden.
Dann werden alle Funktionen der Bibliothek verbunden, bevor die Bezahlsoftware diese aufrufen kann. Die GK-API muss dabei diese notwendigen Funktionen bereitstellen. Die folgende Tabelle 5.1 führt alle von der GK-API nach aussen hin anzubietenden Funktionen mit einer kurzen Beschreibung auf.

Funktionsname	Beschreibung
gk_api_init	Initialisierung von GK-API, Kartenterminal
gk_api_close	Verbindung zum Kartenterminal schliessen und Freigabe von belegten Ressourcen
gk_api_read_card_data	Kartendaten lesen
gk_api_read_card_data_schnell	Kartendaten lesen bei schneller Zahlung
gk_api_abbuchen_einleiten	Abbuchung einleiten
gk_api_abbuchen_einleiten_schnell	Abbuchung einleiten bei schneller Zahlung
gk_api_abbuchen	Betrag abbuchen
gk_api_abbuchen_ie	Betrag abbuchen ersten Schritt
gk_api_abbuchen_ie_schnell	Betrag abbuchen ersten Schritt bei schneller Zahlung
gk_api_abbuchen_iw	Betrag abbuchen weiterer Schritt
gk_api_abbuchen_iw_schnell	Betrag abbuchen weiterer Schritt bei schneller Zahlung
gk_api_fini	Zahlung beenden
gk_api_fini_i	inkrementelle Zahlung beenden
gk_api_error	Übergabe von Fehlerkodes

Tabelle 5.1: Schnittstellenfunktionen

Mit diesen Funktionen der GK-API können alle drei vom ZKA spezifizierten Zahlungsvarianten Abbuchen, Inkrementelles Abbuchen und Schnelles inkrementelles Abbuchen realisiert werden, indem die Funktionen von der Bezahlsoftware mit den korrekten Inputdaten nacheinander aufgerufen werden. Dabei werden manche Funktionen, wie z.B. „gk_api_init“, von mehreren Varianten verwendet, während andere, wie „gk_api_abbuchen_iw_schnell“, nur von einer Zahlungsvariante benutzt werden.
Bei einem Bezahlvorgang muss die Bezahlsoftware die Funktionen in der korrekten Reihenfolge aufrufen, da sonst die kartenterminalinternen Ablaufsicherheitsmassnahmen einen Fehler liefern würden.

Hier die Reihenfolge der Funktionsaufrufe bei den drei möglichen Zahlungsvarianten.
Abbuchen:

1. gk_api_init

2. gk_api_read_card_data
3. gk_api_abbuchen
4. gk_api_fini
5. gk_api_close

Inkrementelles Abbuchen:

1. gk_api_init
2. gk_api_read_card_data
3. gk_api_abbuchen_ie
4. gk_api_abbuchen_iw (eventuell mehrfacher Aufruf)
5. gk_api_fini_i
6. gk_api_close

Schnelles inkrementelles Abbuchen:

1. gk_api_init
2. gk_api_read_card_data_schnell
3. gk_api_abbuchen_ie_schnell
4. gk_api_abbuchen_iw_schnell (eventuell mehrfacher Aufruf)
5. gk_api_fini_i
6. gk_api_close

Eine Ausnahme bietet die Fehlerfunktion „gk_api_error“. Diese kann immer von der Bezahlsoftware aufgerufen werden, sobald ein Fehler aufgetreten ist, damit die GK-API den Fehlerkode an das Kartenterminal weiterleiten kann.

Damit die Bezahlsoftware mit der GK-API beim Funktionsaufruf Daten austauschen kann, wurden vom ZKA Input- und Outputstrukturen nach C-Syntax festgelegt [51]. Diese besitzen den folgenden Aufbau:

Inputstruktur von gk_api_init:

```
typedef struct {
    char  version[10];
} gk_api_init_input, *p_gk_init_input;
```

gk_api_init bekommt von der Bezahlsoftware in der Inputstruktur die erwartete GK-API-Version geliefert.

Outputstruktur von gk_api_init:

```
typedef struct {
    char  version[10];
    char  vendorstring[80];
} gk_api_init_output, *p_gk_init_output, ** const pp_gk_init_output;
```

gk_api_init liefert die GK-API-Version sowie einen Herstellerstring zurück an die Bezahlsoftware.

Inputstruktur von zahlungsrelevanten Funktionen:

```
typedef struct {
    unsigned int   length_data;
    unsigned char  *data_block;
} gk_api_kt_input, *p_gk_kt_input;
```

Alle von der GK-API angebotenen Zahlungsfunktionen ausser gk_api_init, gk_api_error und gk_api_close verwenden die gleiche Inputstruktur. In dieser werden von der Bezahlsoftware ein Datenblock sowie dessen Länge in Byte an die GK-API übergeben.

Outputstruktur von zahlungsrelevanten Funktionen:

```
typedef struct {
    unsigned int   length_data;
    unsigned char  *data_block;
    unsigned int   length_error;
    unsigned char  *error_code;
} gk_api_kt_output, *p_gk_kt_output, ** const pp_gk_kt_output;
```

Alle zahlungsrelevanten Funktionen besitzen im Gegenzug auch eine einheitliche Outputstruktur. Diese enthält einen Antwortdatenblock und dessen Länge in Byte, sowie eventuell einen Fehlerdatenblock und dessen Länge in Byte.

Inputstruktur von gk_api_error:

```
typedef struct {
    unsigned char  error_input[6];
} gk_api_error_input, *p_gk_error_input;
```

gk_api_error wird von der Bezahlsoftware mittels der Inputstruktur ein 6 Byte lange Fehlerkode übergeben.

Outputstruktur von gk_api_error:

```
typedef struct {
    unsigned int   length_error;
    unsigned char  *error_code;
}gk_api_error_output,*p_gk_error_output,** const pp_gk_error_output;
```

Im Gegenzug gibt gk_api_error eine Fehlerdatenblock und dessen Länge in Byte an die Bezahlsoftware zurück.

gk_api_close erhält nur einen Input- und Outputpointer. Diese werden zur Zeit zwar nicht verwendet, wurden aber aus Kompatibilitätsgründen für mögliche spätere Erweiterung bereits eingeplant.

Neben den Outputstrukturen liefern alle GK-API-Funktionen einen Funktionswert an die Bezahlsoftware zurück. Dieser ist für jede Funktion als

`unsigned int` definiert und liefert Informationen über Erfolg oder Nichterfolg bei der Ausführung der GK-API-Funktion.

Da üblicherweise aus Performancegründen keine kompletten Strukturen an Funktionen übergeben werden, benutzt man statt dessen Zeiger auf die Strukturen. Es wurden folgende Festlegungen getroffen:

- Die Bezahlsoftware reserviert den Speicher für die Inputstrukturen sowie Speicherplatz für die Zeiger auf die Outputstrukturen und gibt diese später wieder frei.
- Bei einem Funktionsaufruf werden der GK-API die Adresse des Inputstrukturspeichers (AIS) sowie die Adresse des Speicherplatzes des Zeiger auf die Ausgabestruktur (AOP) übergeben.
- Die GK-API muss den Speicherplatz für die Outputstruktur reservieren und später wieder freigeben. Dem bereits von der Bezahlsoftware reservierten Zeigerspeicherplatz wird die Adresse der Outputstrukur zugewiesen, so dass die Bezahlsoftware nach dem Funktionsaufruf die Outputstruktur von dieser Adresse auslesen kann.

In der folgenden Abbildung 5.4 wird die Speicherverwaltung, der Austausch der Zeiger und der Datenfluss grafisch dargestellt. Das Schema gilt für alle In- und Outputstrukturen gleichermassen.

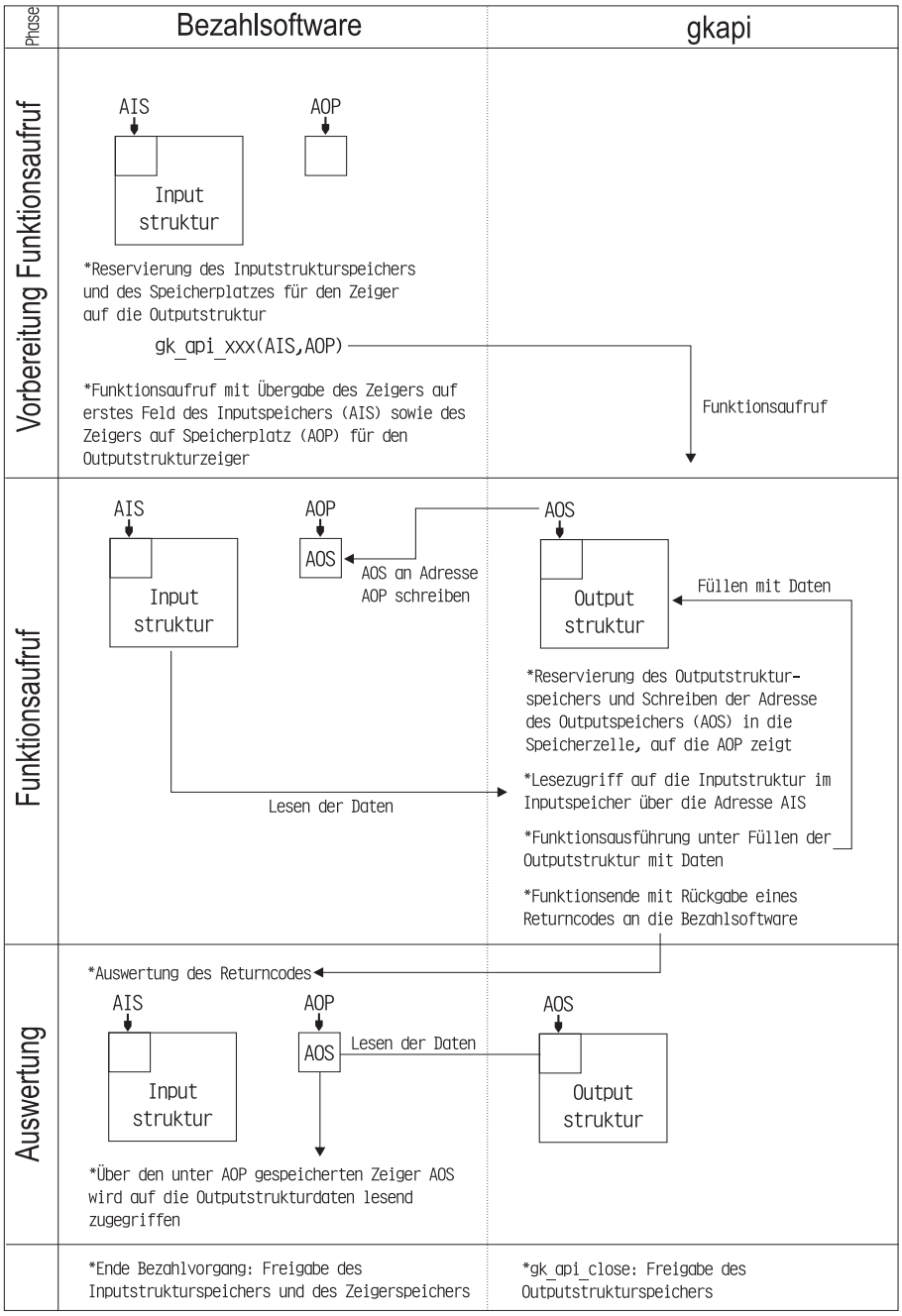

Abbildung 5.4: Speichermanagement bei Datenübergabe

5.4 Kommunikation der GK-API mit dem Kartenterminal

Nachdem die Bezahlsoftware eine Funktion der GK-API-Bibliothek aufgerufen hat, muss die GK-API mit dem Kartenterminal kommunizieren können, um die entsprechende Zahlungsfunktion im Kartenterminal aufzurufen und im Gegenzug die Antwortdaten zu erhalten.
Damit diese Kommunikation mit dem Kartenterminal zustande kommt, wird ein Kartenterminaltreiber benötigt. Dieser ist als standardisierte API konzipiert, welche folgende Hauptaufgaben besitzt:

- Herstellen einer Verbindung zu dem Kartenterminal
- Senden von Daten an das Kartenterminal und Rückgabe der Antwortdaten
- Schliessen der Verbindung zum Kartenterminal

Daneben kann die API noch eine Reihe von Funktionen, z.B. zur Verwaltung der Kartenterminals, bereitstellen.

Bisher haben sich drei grosse Kartenterminal-API etabliert, welche recht unterschiedliche Eigenschaften besitzen. Daher werden sie an dieser Stelle etwas näher beschrieben.

CT-API (CardTerminal-Application Programming Interface):
Die CT-API wurde von der deutschen Telekom, TÜV Informationstechnik GmbH, GMD Forschungszentrum Informationstechnik GmbH und TeleTrusT Deutschland e.V. zusammen entwickelt. Diese Schnittstelle ist daher vor allem in Deutschland im Einsatz. Sie wird zur Zeit nur für die Programmiersprache C angeboten, ist dafür aber plattformunabhängig. Es werden Speicher- und Prozessorkarten unterstützt. Die CT-API bietet nur einen Befehlssatz für die drei Funktionen Initialisierung, Datenversand und Verbindungsende, welcher aber meist ausreichend ist.

PC/SC:
PC/SC (http://www.pcscworkgroup.com) steht für PersonalComputer/SmartCard und ist eine Schnittstelle, welche von einem Konglomerat vieler Firmen entwickelt und anerkannt wurde, unter anderem: Bull, Sun

Microsystems, Microsoft, Hewlett Packard, IBM, Siemens, Gemplus und Schlumberger. Die Schnittstelle ist im Prinzip plattformunabhängig, allerdings muss in das Betriebssystem bereits als Kern ein sogenannter „Resource-Manager“ eingebunden sein. Dies ist bisher nur bei Windows der Fall, aber auch unter Linux wird an einer Umsetzung durch das M.U.S.C.L.E.-Projekt (http://www.linux.net.com/index.html) gearbeitet.
PC/SC arbeitet nur mit Prozessorkarten und nicht mit Speicherkarten. Dafür werden komfortable Funktionen, wie z.B. eine Auflistung der angeschlossenen Kartenterminals, angeboten.

OpenCard:
OpenCard (http://www.opencard.org) ist eine auf Java basierende Kartenterminal-API und daher auf allen Plattformen verwendbar, die eine Javaunterstützung anbieten. Es können neue Terminalklassen integriert werden, so dass sich die OpenCard-Funktionalität erweitert. Auch PC/SC soll als Terminalklasse vorhanden sein. Es können Prozessor- und Speicherkarten angesprochen werden, wobei auch hier OpenCard mehr Funktionen als die CT-API anbietet. Auch dieser Standard wird von grossen Firmen gefördert, wie z.B. IBM, Sun Microsystems, Gemplus, Siemens, Schlumberger und Bull. Wie man erkennen kann, stützen sich diese Firmen also parallel auf PC/SC und OpenCard.

Obwohl noch weitere Kartenterminalstandards existieren, implementieren Hersteller für ein neues Gerät als erstes bevorzugt den CT-API-Standard und bieten ihn auf verschiedenen Systemen an. Dies liegt sicher in der relativ einfachen Integration und dem geringen Funktionsumfang der CT-API im Vergleich zu den anderen Kartenterminaltreiberstandards begründet.
Auch für das KAAN Professional Kartenterminal stand am Anfang die CT-API und PC/SC zur Verfügung. Da ich eine GeldKartenschnittstelle für Solaris/Linux entwickeln wollte, konnte PC/SC nicht verwendet werden, da dieser Standard zur Zeit nur unter Windows integriert ist. Aus diesem Grund habe ich festgelegt, dass die GeldKartenschnittstelle auf der CT-API aufsetzen wird, um mit dem Kartenterminal und der Karte zu kommunizieren.

Zum besseren Verständniss werden an dieser Stelle die Funktionen der CT-API erklärt und der Kommunikationsmechanismus beschrieben. Die Informationen zu den Funktionen basieren auf der CT-API-Spezifikation [50].

5.4.1 Die CT-API

Wie bereits erwähnt bietet die CT-API lediglich 3 Funktionen. Diese sind:

CT-API-Funktion	Aufgabe
CT_init	Initiieren einer Verbindung zum Kartenterminal
CT_data	Senden eines Kommandos an Kartenterminal oder Chipkarte und Rückgabe einer Antwortnachricht
CT_close	Schliessen einer Verbindung zum Kartenterminal

Tabelle 5.2: Funktionen der CT-API

CT_init:
`CT_init(ctn,port);`
Mit CT_init wird eine Verbindung zu einem Kartenterminal an einem bestimmten seriellen Port (port) (Windows:COM1/COM2 bzw. Unix:ttya/ttyb) geöffnet. Dieser Verbindung wird eine vom Programmierer wählbare ID (ctn=0..65535) zugewiesen. Diese ID wird wie ein filedescriptor in C verwendet.

CT_data:
`CT_data(ctn,dad,sad,lenc,command,lenr,response);`
Über die Funktion CT_data kann man ein Kommando (command) mit einer bestimmten Länge (lenc) an eine Kartenterminalverbindung (ctn) senden. Dabei wird die Quelladresse (sad) als Host oder Remote-Host sowie die Zieladresse (dad) übergeben. Ziel kann dabei das Kartenterminal selbst oder ein bestimmter Chipkartenslot (falls das Kartenterminal mehrere besitzt) sein. Die Funktion liefert nach Absetzen des Kommandos an das Kartenterminals eine Antwort (response) mit einer bestimmten Länge (lenr) an das aufrufende Programm zurück.
Die Kommando- und Antwortnachrichten zwischen externer Software und Kartenterminal wurden wie bei der Kommunikation zwischen Kartenterminal und Chipkarte als APDUs realisiert. Das command-Feld von CT_data ist daher eine command-APDU, während das response-Feld die response-APDU enthält. Die APDU-Strukturen wurden bereits in Unterkapitel 4.6erläutert.

CT_close:
`CT_close(ctn);`
Beim Aufruf dieser Funktion wird eine bestehende Verbindung (ctn) zu einem

Kartenterminal abgebaut. Diese Funktion muss daher am Ende aufgerufen werden.

Alle diese Funktionen liefern ein Resultatbyte zurück, welches einen eventuellen Fehlerkode zwischen -128 und 127 enthält. Neben diesen Funktionen der CT-API gibt es noch das „CT-BCS - Anwendungsunabhängiger CardTerminal Basic Command Set für Chipkartenanwendungen“ [52] von TeleTrusT und GMD, welches Zusatzfunktionen wie z.B. REQUEST ICC oder PERFORM VERIFICATION anbietet.
Die CT-API Spezifikation kann als Teil 3 der MKT-Spezifikation kann unter http://www.darmstadt.gmd.de/~eckstein/CT/mkt.html heruntergeladen werden.

5.4.2 Verarbeitung eines Kommandos im Kartenterminal

Sendet man mit CT_data eine Kommando-APDU, so wird die übergebene Zieladresse `dad` des Kommandos geprüft.
Fall 1 (APDU an Kartenterminal): Beträgt dieser Wert 01h, ist das Kartenterminal der Empfänger, so dass es das in `command` gespeicherte Kommando abarbeiten sowie eine Antwort-APDU erzeugen und übergeben kann. Auf die Chipkarte wird dabei normalerweise nicht zugegriffen. Ein Beispiel für solch ein Kommando ist RESET_CT. Dieses Kommando veranlasst das Kartenterminal zu einem Reset mit Rückgabe des kartenterminalinternen Herstellerstrings.

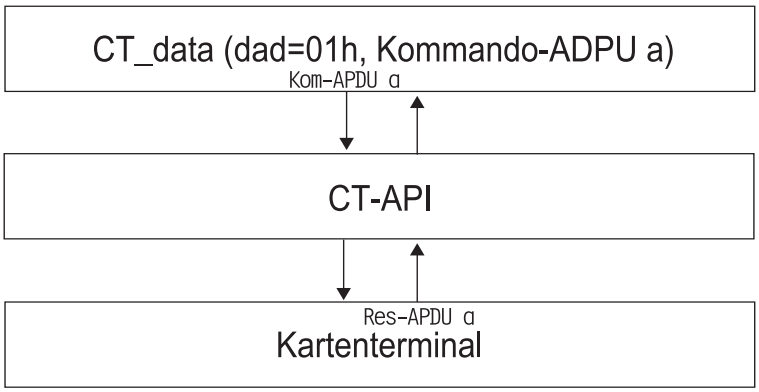

Abbildung 5.5: Fall 1: APDU an Kartenterminal

Fall 2 (APDU an Chipkarte): Ist die Zieladresse des Kommandos hingegen 00h oder grösser als 02h, so ist die Chipkarte im entsprechenden Slot das Ziel (00h=Karte in Slot 1, 02h-0Eh=Karte in Slot 2-14). In diesem Fall bearbeitet das Kartenterminal die APDU nicht weiter, sondern leitet sie im sogenannten „Transparentmodus“ an die Chipkarte weiter. Die Karte verarbeitet das Kommando, wobei zum Teil recht komplexe Operationen wie Verschlüsselung ausgeführt werden, und erzeugt die Antwort-APDU, welche sie an das Kartenterminal liefert. Dieses leitet die Antwort dann an die Applikation weiter. Ein Beispiel für ein Chipkartenkommando wäre die Selektierung einer Datei der Karte mit SELECT_FILE oder eine komplexe Operation wie ABBUCHEN (Variante: Vorgang einleiten).

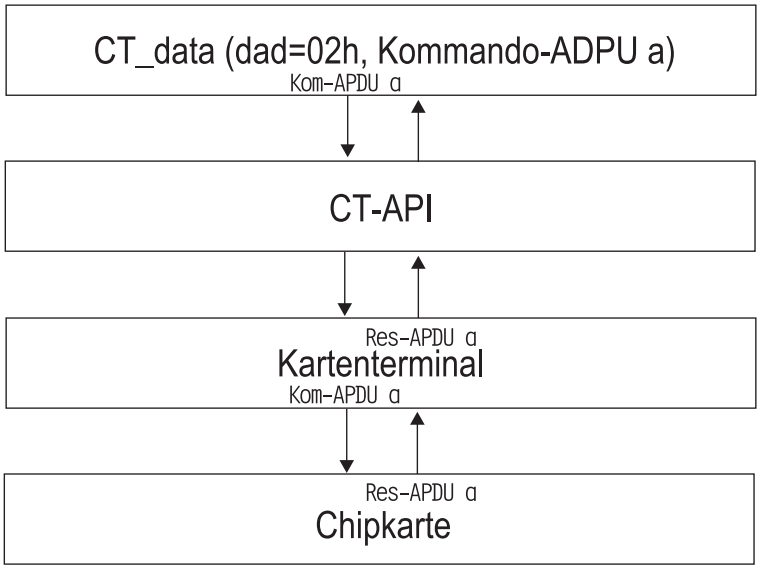

Abbildung 5.6: Fall 2: APDU an Chipkarte

Fall 3 (APDU an Kartenterminal mit Chipkartenzugriff): Als drittes gibt es noch eine Mischung aus beiden Varianten. Dies ist dann der Fall, wenn eine mitunter komplexe Applikation (KT-Kommando), welche im Kartenterminal (dad=01h) bereits integriert ist, aufgerufen wird. Diese kartenterminalinterne Applikation kommuniziert mehrfach über selbst generierte APDUs mit der Chipkarte und bearbeitet deren Antworten. Am Ende stellt die kartenterminalinterne Applikation eine Antwort-APDU aus den Teilantworten der Karte und weiteren Informationen zusammen und sendet diese an das aufrufende Programm.

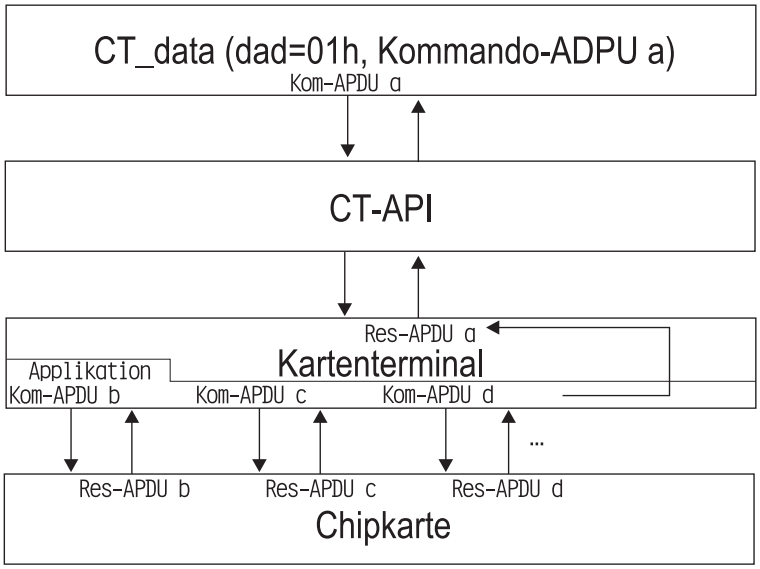

Abbildung 5.7: Fall 3: APDU an Kartenterminal mit Chipkartenzugriff

5.4.3 GK-API-Funktionen und KT-Kommandos

Wie bei den Sicherheitsmassnahmen in Punkt 3.5 erläutert, darf ein Programm nicht wie in Fall 2 sicherheitskritische Kommandos (LADEN, ABBUCHEN, RÜCKBUCHEN etc.) an die Chipkarte senden. Da bei der GeldKartenzahlung allerdings solche Kommandos verwendet werden, muss man die Kommunikation wie in Fall 3 gestalten. Es werden also spezielle kartenterminalinterne Zahlungskommandos benötigt, welche von der GK-API verwendet werden und die sicherheitskritischen Zugriffe ausführen. Auch diese KT-Kommandos für die (inkrementelle) Internetzahlung wurden komplett vom ZKA spezifiziert.
Ein Kundenterminal für GeldKartenzahlungen muss laut ZKA folgende KT-Kommandos integriert haben:

- READ CARD DATA
- READ CARD DATA SCHNELL
- ABBUCHEN EINLEITEN

- ABBUCHEN EINLEITEN SCHNELL
- ABBUCHEN
- ABBUCHEN IE
- ABBUCHEN IE SCHNELL
- ABBUCHEN IW
- ABBUCHEN IW SCHNELL
- FINI
- FINI I
- ERROR

Wenn man diese KT-Kommandos mit den durch die GK-API anzubietenden Funktionen aus Punkt5.3 vergleicht, so bemerkt man, dass alle Funktionen ausser gk_api_init und gk_api_close ein zugehöriges KT-Kommandos besitzen, z.B. gk_api_abbuchen_einleiten und das KT-Kommando ABBUCHEN EINLEITEN. Dabei übernehmen die KT-Kommandos die Hauptarbeit bei einem Zahlungsvorgang: Sie müssen Daten aus der Karte lesen, Texte auf dem Display anzeigen, Tastatureingaben verarbeiten, Antwortdaten überprüfen, Währungen umrechnen und vieles mehr. Um die Komplexität, die sich hinter einem Funktionsaufruf im Kartenterminal verbirgt, zu verdeutlichen, befindet sich auf der CD zur Diplomarbeit ein Programmstück, welches die Funktionalitäten der Kartenterminalfunktionen „read_card_data“ und „abbuchen_einleiten“ nachbildet.
Theoretisch hätte man nicht alle Aufgaben in das Kartenterminal verlegen müssen, da manche GK-API-Funktionen, wie z.B. gk_api_read_card_data, keinen Zugriff auf sicherheitskritische Kommandos der Chipkarte darstellen. Allerdings erhält man so eine einheitliche und übersichtliche Schnittstelle. Einzig die APDU-Kodes zum Aufruf der KT-Kommandos und das Format der Datenübergabe wurden, wie bereits erwähnt, nicht zentral spezifiziert, sondern liegen in der Hand der Kartenterminalhersteller.

Die Hauptaufgabe der GK-API liegt also, neben dem Öffnen und Schliessen der Verbindung zum Kartenterminal, in einer Umsetzung der Aufrufe der Bezahlsoftware in kartenterminalabhängige APDUs, sowie in der Auswertung der Antwortdaten und deren standardisierte Ausgabe an die Bezahlsoftware.

Bei einem Bezahlvorgang rufen die einzelnen GK-API-Funktionen die zugehörigen KT-Kommandos auf und übergeben ihnen im Datenfeld des Body der APDU die zahlungsrelevanten Daten, welche von der Bezahlsoftware geliefert wurden. Desweiteren wird die Antwort des Kartenterminals ausgewertet, die Daten in die Outputstruktur gefüllt und diese an die Bezahlsoftware zurückgegeben.

5.5 Realisierung der plattformübergreifenden Programmierung

Die GeldKartenschnittstelle wurde parallel unter Windows und Solaris in der Programmiersprache C entwickelt. Danach wurde die Solarisversion auch unter Linux getestet.
Unter Windows wurde mit „Microsoft Visual C++ 6.0“ entwickelt. Unter Solaris und Linux kam der „gcc“ in Verbindung mit „configure“ und „make“ sowie weiteren kleinen Tools zum Einsatz.
Es wurde nur eine Quellkodedatei verwendet. Mit der Hilfe von `#define` wird in einer betriebssystemspezifischen Headerdatei (global.h) das jeweilige Betriebssystem festgelegt. Für Windows wird `#define __WIN32` und für Solaris/Linux `#define __UNIX` definiert. Bei der Kompilierung werden durch Abfrage der Betriebssystemvariable nur die jeweils für das Betriebssystem gültigen Teile übersetzt. Das folgende Beispiel demostriert diesen Mechanismus an Hand der betriebssystemspezifischen Angabe des CT-API-Dateinamens.

```
#ifdef __WIN32   //wird unter Windows kompiliert
    char  dllname[256]="Ct32.dll";
#endif
#ifdef __UNIX    //wird unter Solaris/Linux kompiliert
    char  dllname[256]="libct_b1.so";
#endif
```

Der Anteil der betriebssystemspezifischen Programmteile ist allerdings sehr gering. So muss lediglich die Initialisierung (Öffnen, Funktionen verbinden) und Freigabe (Schliessen) der CT-API-Bibliothek unterschiedlich realisiert werden. Auch die Pfadposition der beiden Protokolldateien ist betriebssystemspezifisch. Alle anderen Funktionen und Anweisungen funktionieren systemübergreifend.

Die Umsetzung von Solaris auf Linux konnte recht schnell erfolgen. Dazu musste nur die CT-API ausgetauscht und ein configure-Lauf gestartet werden. configure erkennt die neue Systemumgebung und passt die für die Kompilierung notwendigen Hilfsdateien an.

Unter Windows musste noch eine Datei „libgkapi.def“ erstellt werden, in welcher die von der GK-API zu exportierenden Funktionen aufgeführt sind.

5.6 Methoden der GK-API

In diesem Unterkapitel werden die Funktionen der GK-API genauer dokumentiert und erläutert. Dabei wird auf folgende Programmteile näher eingegangen:

1. Headerdateien
2. Globaler Teil der GK-API
3. Funktion gk_api_init
4. Funktion gk_api_close
5. Funktion gk_api_read_card_data stellvertretend für alle zahlungsrelevanten KT-Kommandos
6. Funktion gk_api_error
7. Hilfsfunktion ctapi_init
8. Hilfsfunktion fill_output_structure
9. Zusatzfunktion gk_api_config

1. Headerdateien:
Es wurden zwei Headerdateien (*.h) angelegt, welche in der GK-API Verwendung finden. Zum einen ist dies die Datei **global.h** und zum anderen die **libgkapi.h**.

Die „global.h“ ist recht kurz und für Windows und Solaris/Linux unterschiedlich. In der Datei wird festgelegt, für welches Betriebssystem die GK-API kompiliert

werden soll. Wurde `#define __WIN32` definiert, so werden die betriebssystemspezifischen Programmteile der GK-API für Windows kompiliert. Ist hingegen das Betriebssystem als `#define __UNIX` festgelegt, so wird der Unix-spezifische Teil verwendet.
Desweiteren kann mit Hilfe der global.h der Debugmodus über `#define __DEBUG` eingeschalten werden. In diesem Modus schreibt die GK-API spezifischere Daten in die Logdatei „gkapisessionlog.txt“. Es werden die Aufrufsparameter der GK-API-Funktionen, die von der GK-API daraus erzeugten APDUs, sowie die Antwortnachricht und deren Auswertung protokolliert. Somit ist es wesentlich einfacher, Fehlerquellen und -ursachen zu finden.

In der Datei „libgkapi.h“ werden als neue Typen die im Unterkapitel 5.3 beschriebenen Input- und Outputstrukturen definiert und die notwendigen Zeigertypen festgelegt.

2. Globaler Teil der GK-API:
Im globalen Teil der GK-API werden neben Standardheaderdateien für Ein- und Ausgabe, Zeitfunktionen usw. (stdio.h, stdlib.h, time.h, fcntl.h) die Dateien „global.h“ und „libgkapi.h“, sowie die Headerdatei der ctapi „ctapi.h“ mittels `#include` eingebunden. Weiterhin werden unter Windows noch die „windows.h“ und „conio.h“, sowie unter Solaris/Linux die „dlfcn.h“ für die Arbeit mit Bibliotheken verwendet.

Danach folgt eine Definition der Baudrate zum Kartenterminal. Diese wird zur Zeit auf maximal 115 KBd gesetzt.

Betriebssystemspezifisch wird dann ein Handle auf die CT-API festgelegt, sowie deren Funktionstypen und Instanzen auf diese Typen spezifiziert. Unter Windows werden dazu für jede CT-API-Funktion die Funktionstypen als Zeiger definiert (z.B. `typedef char (FAR WINAPI *CTAPI_CLOSE)(unsigned short);`), und die Instanzen werden als solche Zeigertypen definiert (z.B. `CTAPI_CLOSE MCT_close;`). Bei Solaris/Linux werden die Funktionstypen als normale Funktionen festgelegt (z.B. `typedef char CTAPI_CLOSE(unsigned short);`). Die Instanzen sind dann als Zeiger auf diese Funktionstypen definiert (z.B. `typedef CTAPI_CLOSE *MCT_close;`).

Anschliessend werden die globalen Variablen vereinbart. Dabei sind die meisten Variablen der GK-API global verfügbar, weil so gut wie alle Funktionen auf diese zugreifen. Eine genaue Beschreibung der einzelnen Variablen erfolgt im Anhang.

3. Funktion gk_api_init:
Diese Funktion ist für die Initialisierung verantwortlich und muss von der Bezahlsoftware als erstes aufgerufen werden (Ausnahme gk_api_config).

Zu Beginn werden einige Variablen festgelegt. Dabei werden neben einer Zeitstruktur auch Pfad und Dateiname von zwei Logdateien angegeben. Datei 1 („gkapi_transactionlog.txt“) erfüllt die Aufgabe der in Unterkapitel 3.5.5 geforderten Protokollierung von Buchungssätzen und Fehlerkodes. Bei jeder Zahlung wird ein Datensatz im ASCII-Format an das Ende der Datei geschrieben. Bei Problemen kann sich der Kunde mit dieser Datei an sein Kreditinstitut wenden. Die zweite Datei („gkapi_sessionlog.txt“) wird bei jeder Zahlung neu angelegt. Sie dient der genaueren Kontrolle des Bezahlvorganges bei Problemfällen. In dieser Datei werden Probleme, wie Speicherplatzmangel, fehlerhafte Initialisierung der CT-API oder fehlende Verbindung zum Kartenterminal, wiederum im ASCII-Format protokolliert. Ausserdem werden die einzelnen Funktionsaufrufe („Kartendaten lesen“oder „Abbuchen“) und deren Resultate („fertig“oder „Problem aufgetreten“) mitgelogt. Somit kann ein Kunde erkennen, in welcher Phase der Transaktion ein Fehler aufgetreten ist. Wurde die GK-API mit `__DEBUG` kompiliert, so werden zusätzlich alle wichtigen Kommando- und Antwortdaten mit in der Datei gespeichert.

Beispiel für gkapi_sessionlog.txt bei erfolgreicher Zahlung:

```
=Start GeldKartezahlung=====================================
| 19.07.2001 , 16:41:01                                    |
| Speicher reserviert                                      |
| gkapi-Version ok                                         |
| Kartenterminaltreiber (ct-api) geladen: CT32.dll         |
| Reset des Kartenterminal ok                              |
| ct\_init ok: Kartenterminal an Port 2 gefunden           |
| Baudrate setzen ok                                       |
| Herstellerangaben holen ok                               |
| Initialisierung fertig (gk\_init)                        |
| Kartendaten gelesen (read\_card\_data)                   |
| Abbuchen eingeleitet (abbuchen\_einleiten)               |
| Abbuchen fertig (abbuchen)                               |
```

```
| Zahlung abgeschlossen (fini)                                    |
| Karte deaktivieren ok                                           |
| Verbindung zu Kartenterminal getrennt                           |
=Ende GeldKartezahlung=====================================
```

Beispiel für gkapi_transactionlog.txt mit einer erfolgreichen Zahlung (Buchungsdatensatz hier verfremdet) und einer Fehlzahlung.

```
=Start GeldKartezahlung=====================================
| 02.08.2001 , 18:06:51                                         |
Zahlung erfolgreich                                             |
Buchungsdatensatz:                                              |
02 C2 00 06 77 52 51 16 02 19 00 29 7D 9D 82 00 00 3E 00 00
20 06 00 20 20 08 09 00 01 20 01 80 02 27 01 24 23 7C 6C 18
37 21 BE C9 2A 01 77 52 36 12 00 A1 04 81 66 4D 34 11 01 C4
04 04 80 45 52 01 61 2B 00 01
=Ende GeldKartezahlung=====================================

=Start GeldKartezahlung=====================================
| 02.08.2001 , 15:08:30                                         |
Kartenterminal konnte nicht initialisiert werden.
ct-init-Fehler=-1 , Port=1 und 2 getestet
=Ende GeldKartezahlung=====================================
```

Zuletzt werden noch die von gk_api_init verwendeten APDUs in Feldern vom Typ `unsigned char` abgelegt.

Danach führt die Funktion folgende Aufgaben aus:

- Anlegen der beiden Logdateien
- Holen der Systemzeit und Schreiben von Datum und Uhrzeit in Logdateien
- Speicherreservierung für Outputstrukturen von gk_api_init (Typ gk_api_init_output) und der KT-Kommandos (Typ gk_api_kt_output)
- Schreiben der GK-API-Version (1.0.0) in Outputstruktur

- Test der erwarteten GK-API-Version
- Aufruf der Hilfsfunktuion ctapi_init
- Suche des Kartenterminals mittels RESET-APDU an COM1/COM2 bzw. ttya/ttyb oder Suche an dem mit der Zusatzfunktion gk_api_config gesetzten Port
- Setzen der Baudrate über SET_BAUD-APDU
- Holen der Herstellerinformationen mittels GET_VENDOR-APDU
- Rückgabe der Outputstruktur

4. Funktion gk_api_close:
Die Funktion gk_api_close ist das Gegenstück zu der gerade dargestellten Initialisierungsfunktion. Sie sorgt für einen sauberen Abschluss der Internetzahlung und die Freigabe aller von der GK-API belegten Ressourcen. Dazu werden folgende Aktionen ausgeführt:

- Als erstes wird eine DEACTIVATE_CARD-APDU an das Kartenterminal gesendet.
- Danach gibt die Funktion den Speicher für die Outputstrukturen (init_output, kt_output, error_output) wieder frei, falls er belegt wurde.
- Die Verbindung zum Kartenterminal wird über die Funktion CT_close der CT-API getrennt.
- Daraufhin wird die CT-API-Bibliothek mit betriebssystemabhängigen Funktionen wieder freigegeben.
- Zum Schluss werden die beiden Logdateien geschlossen.

5. Funktion gk_api_read_card_data:
An dieser Stelle soll der Ablauf der Funktion gk_api_read_card_data stellvertretend für alle hier nicht extra aufgeführten Funktionen erläutert werden. All diese Funktionen sind gleich aufgebaut. Der einzige Unterschied zwischen den Funktionen ist das an das Kartenterminal gesendete Kommando (Kommando-APDU) und die spezifische Textausgabe in die Logdateien.

- In jeder Funktion wird als erstes die funktionsspezifische APDU-Body in einem `unsigned char` -Feld abgelegt (z.B. READ_CARD_DATA_CMD). Die APDU-Kodes sind herstellerspezifisch und unterliegen der Geheimhaltung. Aus diesem Grund werden die jeweiligen Kommandobytes im veröffentlichten Quelltext durch „xx“ ersetzt.
- Funktionsname wird in das Sessionlogfile geschrieben.
- Ausgabe der Inputdaten der Funktion in Sessionlogfile, falls __DEBUG definiert wurde.
- Ablegen der Adresse des Outputstrukturspeichers in die von der Bezahlsoftware reservierte Speicherstelle. Initialisieren der Variable `datapos` .
- Erstellen der funktionsspezifischen Kommando-APDU aus dem APDU-Body (unsigned char-Feld) und den von der Bezahlsoftware übergebenen Inputdaten (`kt_input->data_block`). Im Debug-Modus wird die generierte Kommando-Bytefolge in das Sessionlogfile geschrieben.
- Senden der erstellten Kommando-APDU an das Kartenterminal.
- Aufruf der Hilfsfunktion „fill_output_structure“.
- Funktionsende in Sessionlogfile speichern.
- Funktion verlassen.

6. Funktion gk_api_error:
Die Funktion gk_api_error hat eine Sonderstellung. Sie dient der Übermittlung von Fehlern von der Bezahlsoftware zum Kartenterminal. Gab es z.B. Probleme mit dem Händlersystem, so kann dies über die Funktion gk_api_error dem Kartenterminal mitgeteilt werden. Dieses kann dann den Bezahlvorgang abbrechen und von der Chipkarte einen gesicherten Buchungsdatensatz erzeugen lassen, welcher an die GK-API zurückgeliefert und dort protokolliert wird. Die Funktion arbeitet folgende Schritte ab:

- Der APDU-Body des Error-Kommandos wird in einem unsigned char-Feld abgelegt.

- Der übergebene Fehlerkode wird in das Sessionlogfile geschrieben. Zukünftig soll die GK-API diesen Fehlerkode zusätzlich auswerten, so dass eine für den Nutzer leicht lesbare Fehlermeldung erzeugt werden kann, wie z.B. „Fehler beim Abbuchen-Einleiten. Fehler auf der Händlerseite.“

- Der Speicher für die Outputstruktur (error_output) wird angefordert und die Adresse an die von der Bezahlsoftware übergebene Adresse der Speicherzelle geschrieben.

- Erstellen der Kommando-APDU aus dem APDU-Body und dem von der Bezahlsoftware übergebenen Errorkode (`in->error_input`). Im Debug-Modus wird das generierte Kommando in das Sessionlogfile geschrieben.

- Senden der erstellten APDU an das Kartenterminal.

- Auswerten der Antwortnachricht vom Kartenterminal. Die Nachricht wird untersucht und die extrahierten Daten an die jeweils zugehörige Stelle in der Outputstruktur kopiert. Der Aufbau der Antwort ist herstellerspezifisch. Aus Geheimhaltungsgründen kann deshalb nicht näher auf die Struktur der Antwortnachricht und deren Auswertung eingegangen werden. Im Quellkode wird dieser Teil weggelassen.

- Mit `fflush` werden die Dateipuffer geschrieben. Durch diese Massnahme wird sichergestellt, dass die mitgelogten Daten in die Dateien geschrieben werden. So können trotzdem die Fehler erkannt werden, selbst wenn die Bezahlsoftware die Funktion gk_api_close nicht aufruft, welche die Dateien ordnungsgemäss schliesst und somit ein Schreiben der Dateipuffer auslöst.

- Funktion verlassen.

7. Hilfsfunktion ctapi_init:
Die Hilfsfunktion ctapi_init wird nur von gk_api_init verwendet. Sie sorgt für die Initialisierung der CT-API als Kartenterminaltreiberschnittstelle. Dies ist die einzige Funktion die betriebssystemspezifisch ist. Zum einen sind die CT-API Namen von Betriebssystem zu Betriebsystem verschieden. Desweiteren unterscheiden sich die Funktionen zum Öffnen und Schliessen einer Bibliothek, sowie die Bindung der Bibliotheksfunktionen.

- Zuerst werden die Dateinamen der CT-API in char-Feldern abgelegt. Diese Dateinamen sind herstellerabhängig und betriebssystemspezifisch. Unter Windows wird die Datei „Ct32.dll“ verwendet, unter Solaris/Linux die Datei „libct_b1.so“.

- Betriebssystemabhängiges Laden der CT-API-Bibliothek unter Windows mit `LoadLibrary` und unter Solaris/Linux mit `dlopen`.

- Danach wird eine Verbindung zu den Bibliotheksfunktionen hergestellt. Unter Windows wird dazu `GetProcAddress` und unter Solaris/Linux `dlsym` verwendet.

- Wenn bei der Verbindungserstellung ein Fehler aufgetreten ist, wird dies protokolliert.

8. Hilfsfunktion fill_output_structure:
Die Hilfsfunktion wird von jeder zahlungsrelevanten Funktion (siehe Punkt 5) verwendet. Der Aufruf erfolgt nach Absetzen eine Kommandos an das Kartenterminal und Erhalt der Antwortnachricht. Aufgabe dieser Hilfsfunktion ist es, die in dem Feld `response` gespeicherten Antwortdaten auszuwerten, Teile zu extrahieren und damit die Outputstruktur (kt_output) zu füllen.
Der Aufbau der Antwortdaten ist wie bei gk_api_error herstellerspezifisch und unterliegt ebenfalls der Geheimhaltungspflicht. Aus diesem Grund wird der Auswertungteil dieser Funktion nicht mit im öffentlichen Quellkode erscheinen. Die Funktion führt die folgenden Aktion durch:

- Schreiben der rohen Antwortdaten in das Sessionlogfile im Debug-Modus.

- Überprüfung der Länge der Antwortdaten. Bestehen diese nur aus zwei Byte, so wurden nur die Statusbytes SW1 und SW2 zurückgeliefert, was auf ein Problem bei der Ausführung des zahlungsrelevanten Kommandos im Kartenterminal oder einen Übertragungsfehler hinweist.

- Auswertung der Daten und Füllen der Outputstruktur (geheim).

- Wie bei gk_api_error werden mit `fflush` die Dateipuffer geschrieben. Somit ist sichergestellt, dass vor Beendigung einer zahlungsrelevanten Funktion die gelogten Daten in die beiden Protokolldateien geschrieben werden, so dass diese stets hinreichend aktuell sind.

9. Zusatzfunktion gk_api_config:
Mit dieser sehr kurzen Zusatzfunktion kann der Port an dem sich das Kartenterminal befindet konfiguriert werden. Dadurch kann die Initialisierung (gk_api_init) des Kartenterminals schneller vorgenommen werden, da nicht die Ports durchgeprüft werden müssen. Diese Funktion kann durch die Bezahlsoftware aufgerufen werden, gehört aber nicht zu den Standardfunktionen der GK-API.

5.7 Test der GK-API

Die entstandene Bibliothek (gkapi.dll unter Windows, libgkapi.so unter Solaris/Linux) wurde auf verschiedene Weisen getestet.

Windows:
Unter Windows konnte ein Praxistest durchgeführt, da alle zahlungsrelevanten Softwarekomponenten vorhanden sind. Die hier beschriebene GeldKartenschnittstelle wurde über Testkäufe (Kugelschreiber bzw. Broschüre) mit den Internethändlersystemen der Firmen fun (SmartPay) und Brokat(X-Pay) erfolgreich getestet. Die Firma fun hat sogar wie im realen Leben die Ware ausgeliefert. Leider können die inkrementellen Abbuchungsvarianten in der Praxis nicht getestet werden, da die Internethändlersysteme diese Methoden noch nicht unterstützen. Desweiteren wurde das Testprogramm bssim verwendet, damit auch die inkrementellen Verfahren überprüft werden können. Da wegen fehlender Händlerschlüssel diese Tests nur bis zu den kritischen Abbuchungsfunktionen durchgeführt werden konnten, wird die GK-API zur Zeit von der Firma Kobil mit der bssim-Software getestet. Da ein normaler Bezahlvorgang funktioniert und sich die GK-API-Funktionen des inkrementellen Abbuchens bis auf Kommando-APDUs und Protokollierung nicht unterscheiden, sollten dabei allerdings keine Probleme auftreten.

Solaris/Linux:
Bei diesen beiden Betriebssystemen waren Test schon erheblich schwieriger zu bewerkstelligen, da man mit diversen Problemen konfrontiert ist. Zum einen funktionieren die Applets der Internethändlersysteme noch nicht und zum zweiten exisitiert keine bssim-Testsoftware für diese Plattformen.
Der einzige Ausweg war daher die Erstellung eines eigenen Testprogrammes. Dieses öffnet wie ein Händlersystem die GK-API-Bibliothek (libgkapi.so) und ruft

deren Funktionen auf. Wie in Punkt 5.3 aufgezeigt, werden dabei mitunter recht umfangreiche Parameter im Datenblock übergeben. Dem Testprogramm war es allerdings unmöglich, wie eine Bezahlsoftware, diese Daten zu erzeugen, da keine Händlerkarte im Hintergrund exisitierte. Aus diesem Grund wurden unter Windows die GK-API-Funktionsaufrufe einer erfolgreichen Zahlungen protokolliert. Aus der so entstandenen Logdatei wurden die Aufrufe extrahiert und in die Testsoftware integriert.
Durch diesen kleinen Trick konnte sich die Testsoftware begrenzt wie ein Internethändlersystem verhalten und einen Bezahlvorgang simulierten. Leider werden bei einer GeldKartenzahlung auch zu verwendende Schlüssel festgelegt und Daten ausgetauscht, die bei einer späteren Transaktionen nicht mehr gültig sind. Aus diesem Grunde konnte die Testzahlung nur bis zur Anzeige der Händleridentität durchgeführt werden, da danach das Kartenterminal einen Fehler wegen inkorrekter Daten meldete. Bis auf gk_api_fini wurden dabei allerdings schon alle Funktioen einer normalen Abbuchung verwendet. Für die Funktionen der inkrementellen Zahlungsvarianten gelten die selben Bemerkungen wie bei Windows.

Kapitel 6

Abschlussbetrachtungen

Neue Möglichkeiten durch die GK-API:
Mit der im Rahmen dieser Diplomarbeit entwickelten GeldKartenschnittstelle bietet das Klasse 3 Kundenterminalsystem KAAN Professional alle, für eine GeldKartenzahlung im Internet vom ZKA geforderten, Softwarekomponenten unter den Betriebssystemen Windows, Solaris und Linux an.
Leider ist zum jetzigen Zeitpunkt auf Solaris und Linux trotzdem noch keine GeldKartenzahlung möglich, da auf diesen Systemen die Bezahlsoftware der Internethändlersysteme noch nicht korrekt arbeitet. Da jetzt aber alle grundlegenden Schnittstellen auch unter Linux und Solaris vorhanden sind, bleibt zu hoffen, dass die Hersteller die Bezahlsoftware auch für diese Betriebssysteme in nächster Zeit anpassen. Es ist weiterhin wünschenswert, dass Internethändlersysteme auch die, vom ZKA spezifizierten, inkrementellen Abbuchungsverfahren umsetzen. Gerade diese Varianten sind für wiederkehrenden Zahlungen im Micropaymentbereich, wie sie auch bei eVerlage auftreten, sehr wichtig.

Probleme der GeldKartenzahlung im Internet:
Zur Zeit hat die GeldKartenzahlung im Internet noch mit verschiedenen Problemen zu kämpfen. Diese sind:

- unzureichende Information potentieller Nutzer
- noch recht geringe Akzeptanz auf der Händlerseite
- Kosten der Klasse 3 Kartenterminals

- noch nicht ausgereifte Bezahlsoftware
- z.T. nicht ausreichende Verbreitung der Ladeterminals - ein Ladevorgang am PC wäre daher sehr vorteilhaft
- konkurierende Zahlungsverfahren

Mögliche Anwendungen der GeldKarte:
Sollte es die GeldKarte trotzdem schaffen, sich als Bezahlsystem im Internet durchzusetzen, so könnte sie in recht vielen Anwendungsgebieten zum Einsatz kommen. Möglichkeiten wären:

- Angebot von kostenpflichtigen Informationen
- Download und Bezahlung von Software bzw. Freischaltungsschlüsseln
- kostenpflichtiges Musik- und Videostreaming (PayTV)
- Suche in speziellen Datenbanken und Archiven, welche nicht kostenlos sind

Desweiteren können in die GeldKarte noch weitere Zusatzanwendungen integriert werden. So wäre z.B. eine Nutzung als elektronischer Fahrschein oder Rabattkarte denkbar. Günstig wäre es weiterhin, wenn der ZKA das Laden von GeldKarte gegen GeldKarte spezifizieren würde. Damit wären Transaktionen unter allen GeldKartenbesitzern möglich, so dass, in Anbetracht des günstigen Disagio, fast von einem Bargeldersatz gesprochen werden könnte.
Die Standardisierung durch den ZKA und die Zertifizierungspflicht für GeldKartenzahlungskomponenten sorgen für einheitliche Schnittstellen und Kompatibilität. Leider werden dadurch im Gegenzug auch innovative Lösungen, wie das Laden der Karte am PC zu Hause (siehe http://www.kuk.net), zum Teil etwas behindert. Das führt dazu, dass aktuelle technologische Entwicklungen oft erst zeitverzögert im GeldKartenzahlungssystem verwendet werden.

Fazit:
Der GeldKartechip bietet schon heute faszinierende Möglichkeiten an, die unseren Umgang mit Geld z.T. verändern könnten. Bleibt zu hoffen, dass das Potential hinter dieser Technologie ausgenutzt wird und über günstige Marketingstrategien die tatsächlichen Kunden- und Händlerzahlen steigen, so dass sich die GeldKarte fest als Zahlungssystem etabliert.

Glossar

API (Application Programming Interface: Programmierschnittstelle. Definierte Softwareschnittstelle zum Aufruf von vorgefertigten Programmroutinen.

APDU (Application Protocol Data Unit): Datenstruktur zur Kommunikation mit Kartenterminal oder Chipkarte. Diese kann entweder einen Kommandoaufruf inklusive Parametern enthalten (Command-APDU), welcher an das Kartenterminal oder die Chipkarte gesendet und dort verarbeitet wird. Oder es handelt sich um eine Response-APDU, die die Antwortdaten von der Chipkarte/Kartenterminal enthält, welche an die aufrufende Software übergeben werden.

Application: Applikationsprotokoll (Befehlssatz) zwischen Kartenterminal und Chipkarte und die dazugehörigen Daten.

Bezahlsoftware: Software, welche die Zahlung auf dem Kunden-PC abwickelt und per Webbrowser gestartet wird.

Börsenkarte: Die Börsenkarte ist eine, an ein bestimmtes Konto gebundene, Geldkarte und kann von diesem aufgeladen werden z.B. EC-Karte.

Chipkarte: Plastikkarte mit integriertem Mikroprozessor und Kontaktierfläche. Es exisitieren Speicherkarten zur Datenspeicherung und Prozessorkarten für komplexe Programmfunktionen. engl. ICC - Integrated Circuit Cards

CT-API: Einfache aber effiziente API zur Kommunikation zwischen Anwendung und Kartenterminal bzw. Chipkarte. Besonders in Deutschland verbreitet.

DES (Data Encryption Standard): DES (DES und Triple-DES) ist ein symmetrisches Verschllusselungsverfahren. Es werden 64 Bits Klartext in 64 Bits Schlüsseltext und umgekehrt transformiert. Der DES-Algorithmus ist nach ANSI-Standard normiert (ANSI X3.92-1981) und findet häufig im Finanzsektor Anwendung.

DF (Dedicated File): Verzeichnis einer Prozessorkarte

DF (Elementary File): Datei einer Prozessorkarte

GeldKarte: Die GeldKarte ist eine Chipkarte, welche als elektronische Geldbörse genutzt werden kann. Es gibt Börsen- und Wertkarten.

GK-API (GeldKarte-API): Schnittstelle welche Zahlungsfunktionen für eine Bezahlsoftware anbietet.

Händlerkarte: Gegenstück zur Kunden-GeldKarte. Auf die Händlerkarte wird bei einer Transaktion Geld gebucht.

Händlersystem: Hintergrundsystem zur Steuerung eines Zahlungsablaufes.

ICC (Integrated Circuit Card): Integrierter Schaltkreis. Chip welcher einen bestimmten Funktionsumfang anbietet.

IEC (International Electrotechnical Commision): Standardisierungsgremium

Internet-Händlersystem: Händlersystem für die GeldKartezahlung im Internet.

Internet-Kundenterminal: Heimischer PC oder Workstation mit Internetanschluss.

ISO (International Organisation for Standardization): Normierungsstelle, welche Standards festlegt.

Kartenterminal: Gerät mit welchem man Chipkarten benutzen kann. Das Kartenterminal kann Daten an die Chipkarte senden und Daten von ihr empfangen. Manche Kartenterminals bieten noch Zusatzfunktionen, wie eine Tastatur und ein Display.

KT-Kommando (Kartenterminal-Kommando): Kartenterminalinternes Kommando, welches von einer Software aufgerufen werden kann. Hinter einem solchen Kommando verbirgt sich meist ein kleines Programm (Applikation).

Kundenterminal: ZKA-Bezeichnung für Kartenterminal.

Kundenterminalsystem: Gesamtpaket aus Kundenterminal, Kartenterminaltreiber und GeldKartenschnittstelle.

MD5: Hashfunktion, welche einen 128-Bit Hashwert erzeugt. Von Ron Rivest entwickelt.

MF (Master File): Hauptverzeichnis einer Prozessorkarte

OCF (Open Card Framework): Ein weiterer Standard für die Kommunikation zwischen Anwendung und Kartenterminal bzw. Chipkarte, welcher auf Java basiert.

PC/SC: Standard für terminalunabhängige API zur Kommunikation zwischen Anwendung und Kartenterminal bzw. Chipkarte.

PIN (Personal Identification Number): Geheimzahl zur Authentifizierung.

Prozessorkarte: Chipkarte mit integriertem Microcontroller. Die GeldKarte ist eine Prozessorkarte mit CPU, RAM, Datei- und Betriebssystem.

RSA (Rivest, Shamir, Adlema): asymmetrisches Public-Key-Verfahren, welches sowohl zur Verschlüsselung als auch zum Erzeugen von Signaturen verwendet werden kann. Beruht auf der Schwierigkeit die Primfaktoren einer groen Zahl zu bestimmen.

Smart Card: siehe Chipkarte.

SMS (Short Message System): Sehr populärer Service zur Übermittlung von Kurznachrichten per Mobiltelefon.

Speicherkarte: Chipkarte zur Speicherung von Daten

TAN (TransActionNumber): Transaktionsnummer, welche zur Ausführung genau einer Transaktion berechtigt und danach verfällt.

TLV (Tag-Length-Value): Struktur zur Speicherung von Daten (Value) mit vorangestelltem Datentyp (Tag) und Datengrösse (Length).

Transparentmodus: Modus, in dem das Kartenterminal Kommandos direkt an die Chipkarte weiterleitet.

weisse GeldKarte: siehe Wertkarte

Wertkarte: kontoungebundene GeldKarte, welche vollkommene Anonymität gewährleistet

ZKA (Zentraler KreditAusschuss): Gremium, welches Hard- und Software für die GeldKartezahlung sowie die GeldKarte selbst spezifiziert hat.

Literaturverzeichnis

[1] Knud Böhle und Ulrich Riehm, ITAS. *„Elektronisches Geld und Internet-Zahlungssysteme. Innovationen, Mythen, Erklärungsversuche“*. TA-Datenbank-Nachrichten, Nr.2, 7. Jahrgang - Juni 1998, S. 40-54. http://www.itas.fzk.de/deu/tadn/tadn298/bori298b.htm [2.8.2001]

[2] *„Sicherheit und elektronische Zahlungssysteme im Internet Eine Kurzbersicht“* von http://www.electronic-commerce.org/zahlungssysteme/allgemeine/tabelle.html [4.12.2000]

[3] Jürgen Seegers. *„Kritische Masse“* Artikel der Zeitschrift iX, Ausgabe 3/2001, S. 48-57. Verlag Heinz Heise

[4] Holger Reibold. *„Zur Kasse bitte! “* Artikel der Zeitschrift Internet Professional, Ausgabe Mai 2001, S.84-87. VNU Business Publications Deutschland GmbH

[5] Prof. Dr. Karl-Heinz Ketterer. Zusammenfassung der IZV4-Studie des Institut für Wirtschaftspolitik und Wirtschaftsforschung der Universität Karlsruhe. http://iww.uni-karlsruhe.de/IZV4/auswertung/showalle.html [23.7.2001] (Tabelle 8)

[6] Kostentabelle eops Cards/Transactions von http://www.eops.de/german/htdocs/produkte/includes/pdf/pl_e-ca-tra.pdf [13.8.2001]

[7] Kosten Net900 von: http://www.in-medias-res.com/netkosten.htm [13.8.2001]

[8] FAQ zu Net900 *„Wie funktioniert Net900 Kontopass ?“*, von: http://www.net900.de/support/lan_faq/faq/funktion.html [13.8.2001]

[9] Produktbeschreibung von eops-Webseite: http://www.eops.de/german/htdocs/produkte/cards.php [13.8.2001]

[10] *„Verfahrensanleitung für das Bezahlen mit der Kreditkarte im Internet“* von Sparkassenwebseite: http://ecommerce.sparkasse.de/down/anleitung_set.doc [13.8.2001]

[11] *„CyberCash CashRegister - How It Works...“* , von CyberCash Webseite: http://www.cybercash.com/cashregister/howitworks.html [13.8.2001]

[12] *„Purchasing CashRegister Service“* , von CyberCash Webseite: http://www.cybercash.com/cashregister/sales.html [13.8.2001]

[13] Eigenschaften des Zahlungssystems von aposto-Webseite: http://www.aposto.de/archi1_0.html?head=archi&sub=1 [13.8.2001]

[14] Zahlungsablauf von aposto-Webseite: http:/www.aposto.de/archi2_0f.html?head=archi&sub=2/ [13.8.2001]

[15] Kostenauflistung von der paybox Firmenwebseite: http://www.paybox.de/costs.html [13.8.2001] ,

[16] *„Produktbeschreibung zu Street Cash“* per e-Mail erhalten

[17] Funktionsweise Street Cash , http://www.streetcash.de/html/funktionsweise.html [13.8.2001]

[18] Sicherheitsinformation der paybox AG , http://www.paybox.de/security.html [13.8.2001]

[19] Preisliste per e-Mail erhalten. Siehe CD

[20] Zahlungsablauf von aposto-Webseite: http://www.eops.de/german/htdocs/produkte/mobile.php [13.8.2001]

[21] Eigenschaften des Zahlungssystems und Zahlungsablauf von eops-Webseite: http://www.eops.de/german/htdocs/produkte/pin.php [13.8.2001]

[22] Webseite monkeybank http://www.monkeybank.com [13.8.2001]

[23] *„Das Handy wird zur Geldbörse“* , Informationen von Payitmobile-Webseite: http://www.payitmobile.com/Kunden/kunden.html [13.8.2001]

[24] Webseite Mondex: http://www.mondex.com/webcode/common/contents.asp?ID=45 [13.8.2001]

[25] ECIN (Electronic Commerce Info Net) Webseite „electronic commerce“: http://www.ecin.de/zahlungssysteme/voraussetzung/index-2.html [13.8.2001]

[26] FAQ zu click&buy von Firstgate-Firmenwebseite: http://www.firstgate.de/info/template_faq_wasist.html [13.8.2001]

[27] Kostenübersicht von Firstgate-Firmenwebseite: http://www.firstgate.de/wasist/kosten.html [13.8.2001]

[28] Funktionsweise net900 classic von in media res-Webseite http://www.net900.de/support/faq/funktion.html [13.8.2001]

[29] Zahlungsablauf von eops-Webseite: http://www.eops.de/german/htdocs/produkte/call.php [13.8.2001]

[30] Kostenlisten eops-Connector PerClick von eops-Webseite: http://www.eops.de/german/htdocs/produkte/includes/pdf/pl_e-connector_PerCall.pdf [13.8.2001]

[31] Funktionsweise PurePay von Firmenwebseite: http://www.purepay.de/einkaufen/sofunktioniert.html [13.8.2001]

[32] paysafecard-FAQ von Firmenwebseite: http://www.paysafecard.com/de/de/m_fragen.shtml [13.8.2001]

[33] *„Virtuelle Münzen - elektronisches Geld“* , Beitrag ECIN-Webseite: http://www.ecin.de/zahlungssysteme/voraussetzung/index-4.html [13.8.2001]

[34] *„Verrechnung mit elektronischen Schecks“* , Beitrag ECIN-Webseite: http://www.ecin.de/zahlungssysteme/voraussetzung/index-3.html [13.8.2001]

[35] eVerlage Zahlungssystemhilfe: http://birne.offis.uni-oldenburg.de/eVerlage/help/helpzs.html [9.7.2001]

[36] Dr. Tschangiz Scheybani/GIESECKE&DEVRIENT GmbH. *„Die Bedeutung der virtuellen Händlerkarte für das Zahlen mit der Geld-Karte im Internet“* Information aus Vortrag (Stand Oktober 1999) entnommen

[37] Auflistung der Institute aus GeldKarte-Verrechningsinformationen von: http://ecommerce.sparkasse.de/openworx.php?def=liste_tunnel&vlg=liste_tunnel&id=124 [13.8.2001]

[38] Nachzeichnung des Ablaufbildes von ECIN-Webseite, http://www.electronic-commerce.org/zahlungssysteme/smartcards/images/geldkarte.gif [4.12.2000]

[39] Nachzeichnung der Grafik aus der ZKA-Spezifikation: *„GeldKarte: Händlersysteme, Internet Kundenterminal“*

[40] ZKA-Spezifikation: *„GeldKarte: Händlersysteme, Internet Kunden-terminal“* Vers. 3.2

[41] Informatikzentrum der Sparkassenorganisation GmbH: *„Anforderun-gen an Chipkartenleser für den Heimbereich aus Sicht der SKO“*, Version 1.0, September 1997

[42] Produktinformation Kobil KAAN Professional von Firmenwebseite: http://www.kobil.de/seiten/d/ct/kaan_pro.htm [13.8.2001]

[43] ISO/IEC 7816 - Industrienorm für Chipkarten. Diese kann unter http://www.iso.ch erworben werden. Eine Übersicht der Teil-Normen

und ihrer Beschreibung findet sich ebenfalls auf der ISO-Webseite oder auf der CD zur Diplomarbeit

[44] Chipkartenbild (Kontakte.gif) und Prozessorkartenschema (secusinglechip.gif) von: http://www.hardwareecke.de/specials/chipkarten.html [17.8.2001]

[45] ZKA-Spezifikation: *„Datenstrukturen und Kommandos“* V4.1

[46] Ralf Gladis. *„Netzkröten - EBusiness mit der Geldkarte“* Artikel der Zeitschrift ct, Ausgabe 11/1998, S. 52. Verlag Heinz Heise

[47] ZKA-Spezifikation: *„GeldKarte - Applikation elektronische Geldbörse“* V4.3.1

[48] Kai-Uwe Mrkor. *„Kartenspiele - Grundlagen der Chipkartenprogrammierung“* Artikel der Zeitschrift ct, Ausgabe 8/2000, S. 211. Verlag Heinz Heise

[49] ZKA-Spezifikation: *„Kurzdarstellung des Betriebssystems der ZKA-Chipkarte“* V4.1

[50] CT-API 1.1 (Teil 3) *„Anwendungsunabhängiges CardTerminal Application Programming Interface für Chipkartenanwendungen“*. Die CT-API-Spezifikation (mktp3v09.doc) kann unter folgender URL heruntergeladen werden: http://www.darmstadt.gmd.de/~eckstein/CT/mkt.html [13.8.2001]

[51] ZKA-Spezifikation: *„GK-API zum Bezahlen mit der GeldKarte im Internet“* V1.4

[52] CT-BCS 0.9 (Teil 4) *„Anwendungsunabhängiger CardTerminal Basic Command Set für Chipkartenanwendungen“* CT-BCS (mktp4v09.doc) kann unter folgender URL heruntergeladen werden: http://www.darmstadt.gmd.de/~eckstein/CT/mkt.html [13.8.2001]

Abbildungsverzeichnis

Tabellenverzeichnis

Anhang A

Bezugsadressen und Spezifikationen

Die **„Schnittstellenspezifikationen der ZKA-Chipkarte“** ist für eine Schutzgebühr von 400 DM beim Bundesverband Öffentlicher Banken e.V. (ZKA-Mitglied) erhältlich:
Bundesverband Öffentlicher Banken e.V.
Lennestrasse 17
10785 Berlin
Tel.: 030/81 92-1 81
Fax: 030/81 92-1 89

Test-GeldKarten können von der VöB-ZVD Bank für Zahlungsverkehrsdienstleistungen GmbH angefordert werden:
E-Mail: zvd@voed-zvd.de
Internet: www.voeb-zvd.de
Postanschrift:
VöB-ZVD
Postfach 26 01 32
53153 Bonn

oder

VöB-ZVD
Bank für Zahlungsverkehrsdienstleistungen GmbH

Godesberger Allee 88
53175 Bonn

Die **Testsoftware bssim** ist ebenfalls beim VöB-ZVD von der Abteilung „Funktionstest“ erhältlich:
E-Mail: funktionstest@voeb-zvd.de
Postanschrift:
VöB-ZVD
Bank für Zahlungsverkehrsdienstleistungen GmbH
Funktionstest
Godesberger Allee 88
53175 Bonn

Testschlüssel Zur bssim-Software werden noch Testschlüssel (KGKRD) benötigt. Bei Erfüllung bestimmter Voraussetzungen sind diese auf Anfrage vom Key Administration Center erhältlich:
Bank-Verlag GmbH (BV)
Melatengürtel 113
50825 Köln
Tel: 0221/5490-0
Fax: 0221/5490-120
Für die Schlüssel sollten Kosten von ca. 450 DM einkalkulieren werden.

CT-API Spezifikation Die CT-API Spezifikation kann als Teil 3 der MKT-Spezifikation unter http://www.darmstadt.gmd.de/~eckstein/CT/mkt.html heruntergeladen werden.

ISO/IEC-Norm 7816 Die ISO/IEC Norm 7816 zu Chipkarten ist bei http://www.iso.ch erhältlich. Die Spezifikation besteht aus 10 Teilen (siehe Webseite oder CD)

EMV 96 „EMV 96 Integrated Circuit Card - Specification for Payment Systems“ von Europay, MasterCard und Visa. Spezifikation auf beiligender CD.

eZI-L Das ECIN (Electronic Commerce Info Net) bietet mit „eZI-L“ (elektronische Zahlungssysteme im Internet) eine interessante Mailingliste an, auf welcher auch neue Trends zu erfahren sind.
http://www.ecin.de

Anhang B

Weitere interessante Zahlungsverfahren

Hier werden weitere interessante Zahlungsverfahren vorgestellt. Die Gruppierung erfolgt wie in Kapitel 2.

B.1 Kontobasis

Bankeinzug/Lastschrift:
Von der eops AG existiert auch ein elektronisches Lastschriftverfahren namen „eops-Transactions“, welches die Kontodaten des Kunden gesichert an die Bank des Händlers übermittelt und eine Liquiditätsprüfung des Kunden unternimmt. Dafür muss der Händler je nach Anzahl der Transaktionen pro Monat zwischen 0.40-0.65 DM zzgl. Mehrwertsteuer pro Transaktion und 55 DM/Monat zzgl. Mehrwertsteuer zusätzliche Grundgebühr an die eops AG bezahlen (Kosten aus [6]).

Net900 Kontopass:
Mit der Variante „Kontopass“ integriert auch Net900 das Bezahlen mittels Bankeinzug. Der Kunde muss sich mit seiner Kontoverbindung anmelden und erhält danach von Net900 eine Überweisung, wobei im Verwendungszweck dem Kunden eine PIN mitgeteilt wird. Mit dieser kann man einmalig die Bankeinzugsermächtigung freischalten.

Trifft man auf einen kostenpflichtigen Inhalt, so wird der NET900-Server angewählt und der Kunde über den Preis informiert. Dieser kann zwischen 0.29 und 25 DM liegen (Zahlungsbereich aus [7]). Bestätigt man diese Information mit Nutzernamen und Kennwort, so wird der Betrag bei Net900 registriert und der Inhalt freigegeben. Die Summe aller getätigten Zahlungen wird einmal monatlich vom Konto des Kunden eingezogen (Funktionsweise aus [8]).
Für den Händler fallen 75 DM Einrichtungs- sowie 7.50 DM Monatsgebühren an. Desweiteren muss eine Umsatzpauschale von 15-30 % abgeführt werden (Kosten aus [3]).

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Anonym gegenüber Händler
Micropayment möglich	Ja	Ab 0.29 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	
Mobilität hoch	Ja	Durch Nutzerauthentifizierung überall nutzbar
Kosten Händler gering	Mittel	Relativ hohe Umsatzprovision
Kosten Kunde gering	Ja	Nur Kontoführungskosten
Einstieg einfach	Nein	Anmeldung,PIN-Freigabe
Bedienung einfach	Ja	Nur Authetifikation
Zahlungsbereich gross	Nein	Bis 25 DM sinnvoll
Akzeptanz hoch	Nein	Zu Neu
Tranparenz hoch	Ja	

Tabelle B.1: Bewertung Net900 Kontopass

Eignung für eVerlage: Da Net900 Kontopass noch recht neu ist, besteht das Akzeptanzproblem seitens Kunden und Händler. Trotzdem ist das Verfahren brauchbar, da es anonym ist und in Echtzeit Micropayment ermöglicht. Deshalb wäre über eine Implementierung, evtl. im Rahmen eines Pilotprojektes, nachzudenken. Es ist allerdings zu prüfen, ob die Transaktionskosten vertretbar sind.

B.2 Kreditkarte

eops-Cards:
Das Kreditkartenabrechnungsverfahren „eops-Cards“ der eops AG soll hier nur kurz vorgestellt und nicht weiter bewertet werden, da nur der Händler richtig eingebunden ist und der Kunde weiterhin seine Kartendaten über das Internet versenden muss. eops empfiehlt dem Händler, ein SSL-Zertifikat einzubinden, da nur die Kommunikation zwischen Händlershop und eops-Gateway abgesichert ist. eops-Cards ist weniger ein eigenständiges Zahlsystem als vielmehr eine Erweiterung des Kreditkartenverfahrens, welches eine einfache Liquiditätsprüfungen und Abrechnungen ermöglicht. Dadurch kann von einem Echtzeit-Clearing gesprochen werden, und der Händler hat noch weniger Aufwand mit der Einreichung der Kreditkartendaten. Ausserdem wird eine Vielzahl von Kartentypen unterstützt.
Die eops AG beschreibt ihr Produkt wie folgt: *„eops-Cards ist ein professionelles Abrechnungsverfahren, mit dem Internet-Händler online Geschäfte über alle gängigen Kreditkarten abwickeln können. VISA, EuroCard/MasterCard, American Express, Diners Club und JCB schaffen diesem Macropayment-System den Background für finanzielle Transaktionen im Internet. Mit eops-Cards steht jeder Online-Shop in direkter Verbindung zum Autorisierungssystem des jeweiligen Finanzdienstleisters. Die sichere Zahlungsabwicklung wird durch die Verschlüsselung aller Daten garantiert. Das Ergebnis der Buchungen liegt Händler und Kunden bereits nach wenigen Sekunden vor.“* [9]
An Kosten entstehen dem Händler pro Transaktion zwischen 0.40-0.65 DM zzgl. Mehrwertsteuer, sowie weitere 55 DM zzgl. Mehrwertsteuer Grundgebühren im Monat (Kosten aus [6]).

Eignung für eVerlage: Erweiterungen des Kreditkartensystem wie eops-Cards lohnen sich erst ab einer gewissen Anzahl von Transaktionen, oder wenn besonders viele internationale Kartentypen akzeptiert werden sollen.

CashRegister:
CyberCash löst mit CashRegister das Datensicherheitsproblem bei Kreditkartenzahlungen (Visa, MasterCard, American Express, Discover/Novus und mehr) etwas anders. Wie bei SET muss auf dem Kundenrechner eine Software installiert werden. Bei einer Transaktion verschlüsselt diese die Kreditkarteninformationen des Kunden mittels RSA (Public Key), DES (symmetrischer Algorithmus) und MD5 (Hash) und schickt die Nachricht über das Internet an den Händlerserver. Der Server erweitert die für ihn nicht komplett lesbare Nachricht um die Händ-

leridentifikationsdaten und sendet sie dann signiert an den CyberCash-Server. Dieser prüft die Liquidität des Kunden und veranlasst bei erfolgreicher Prüfung die Kreditkartenzahlung bei den Bankinstituten (Funktionsweise aus [11]).
Das Clearing soll nahezu in Echtzeit ablaufen.
Dem Händler entstehen Kosten von 495 US$ für die Einrichtung, 20 US$ monatliche Grundgebühr und 0.20 US$ Transaktionsgebühr. Micropayment ist damit nur begrenzt möglich (Preise aus [12]).
Bei CyberCash gab es finanzielle Probleme, so dass sie jetzt von VeriSign übernommen wurden. Die CyberCash Softwareproduktlinie wurde von First Data Merchant Services aufgekauft.

Eignung für eVerlage: Das Bezahlsystem ist recht teuer und nur begrenzt im Micropaymentbereich einsetzbar. Auch die Zukunft dieser Lösung ist ungewiss. Daher lohnt eine Integration vorerst nicht.

aposto:
aposto von der Firma Aposto akzeptiert ec-, Kreditkarten, Flottenkarten und Kundenkarten. Das Bezahlsystem ist eine Mischung aus Kreditkartenzahlung und SET. Dabei versucht aposto genau wie SET, einige negative Eigenschaften der normalen Kreditkartenzahlung, wie mangelnde Anonymität und Sicherheit, zu beseitigen.
So sollen die Kartendaten verschlüsselt übertragen werden und für den Händler nicht einsehbar sein. Der Kunde ist also anonym gegenüber dem Händler, aber nicht gegenüber dem Zahlungssystemanbieter (Aposto). Durch die Verschlüsselung sind die Kreditkartendaten sicherer. Ausserdem reserviert aposto die Zahlungen, wodurch dem Händler eine Zahlungsgarantie entsteht (Eigenschaften aus [13]). Der Kunde ist allerdings darauf angewiesen, dass der Händler die Auslieferung der Ware wahrheitsgemäss an aposto mitteilt.
Der Händler muss sich wie bei eops-Cards nicht mehr um die Zahlungsabwicklung kümmern.
Der Zahlungsablauf [14] geschieht in der folgenenden Art und Weise:

1. *Der Kunde füllt seinen Warenkorb und will mit Karte bezahlen.*
2. *Die Funktion Kartenzahlung löst einen Weblink zu aposto aus.*
3. *aposto schickt ein verschlüsseltes Formular an den Kunden.*
4. *Der Kunde gibt seine Kartendaten bzw. seine Bankdaten ein.*

5. *Das Formular wird ein weiteres Mal verschlüsselt.*
6. *aposto stellt eine Autorisierungsanfrage.*
7. *aposto schickt dem Kunden o.k.-Bestätigung und Tracking-ID.*
8. *aposto schickt dem Händler die Transaktionsbestätigung.*
9. *Der Händler bestätigt dem Kunden den Auftrag.*
10. *Der Händler teilt aposto die Auslieferung der Ware mit.*
11. *aposto wandelt die reservierte Zahlung in eine Buchung um.*
12. *aposto löst die Gutschrift auf das Konto des Händlers aus.*

Eignung für eVerlage: Genau wie SET könnte man sich aposto genauer ansehen, um eine für Kunden und Händler sicherere Kreditkartenzahlung zu gewährleisten. Es muss allerdings vor der Integration überprüft werden, ob der Kunde den Mehraufwand für eine sicherere Zahlung überhaupt akzeptiert.

B.3 Mobiltelefon

eops-Mobile:
Ein weiteres Mitglied aus der eops-Familie ist „eops-Mobile“. Nachdem sich ein Kunde bei dem Zahlungsprovider angemeldet hat, kann er folgendermassen das Bezahlsystem nutzen [20]:

1. *Der Kunde kauft ein Produkt und wählt zur Zahlung das eops-Mobile System.*
2. *Der Händler initiiert die Transaktion mit einem Link zum eops-Mobile System und das Eingabefenster erscheint auf dem Bildschirm des Kunden.*
3. *Der Kunde gibt seine Mobilfunknummer für die darauffolgende Autorisierung ein. Der Shop-Server übermittelt diese Nummer über eine sichere Datenverbindung an den eops-Server.*

4. *Das eops-Mobile Sprachsystem ruft binnen weniger Sekunden das Mobiltelefon des Kunden an und teilt ihm den Händler und den zu zahlenden Preis mit.*

5. *Durch die Eingabe des eops-Mobile Code autorisiert der Käufer den Zahlvorgang. Anschliessend wird der fällige Betrag entweder von der Kreditkarte oder dem Girokonto des Kunden abgebucht und dem Lieferanten gutgeschrieben.*

6. *Nach erfolgreichem Abschluss des Vorgangs erhält der Händler die Bestätigung der Zahlung.*

7. *Der Kunde wird auf die Bestätigungsseite des Händlers geleitet.*

Wie auch paybox oder Street Cash ist eops-Mobile nicht nur im Internet nutzbar, sondern es können auch Transaktionen zwischen eops-Mobile Kunden und Zahlungen an weitere autorisierte eops-Mobile Partner erfolgen. Das System ist noch nicht so weit fertig, dass Preislisten bekannt sind. Per e-Mail konnte ich erfahren, dass Transaktionskosten und evtl. eine monatliche Grundgebühr für den Händler anfallen werden.

Eignung für eVerlage: Das System ist noch nicht ganz marktreif, könnte aber, wie paybox und Street Cash, für eVerlage geeignet sein.

eops-PIN:
„Für eops-PIN wurde die Technologie von eops-Mobile noch einmal um einige Funktionen erweitert. So entstand eine zusätzliche Möglichkeit, Rechnungen im Internet mit dem Handy zu begleichen. Das Auslösen der Zahlungen durch eops-PIN erfolgt wahlweise per SMS, e-Mail oder Voice-Message. Vor der ersten Transaktion ist lediglich eine einmalige Registrierung erforderlich. Dabei entscheidet der Kunde, ob die Beträge über sein Girokonto oder über seine Kreditkarte abgebucht werden.“ [21]
Der Zahlungsablauf [21] gestaltet sich folgendermassen:

1. *Der Kunde kauft ein Produkt und wählt zur Zahlung das eops-PIN System.*

2. *Der Händler initiiert die Transaktion mit einem Link zum eops-PIN System, und das Eingabefenster erscheint auf dem Bildschirm des Kunden*

3. *Der Kunde gibt seine Mobilfunknummer für die darauffolgende Autorisierung ein. Der Shop-Server übermittelt diese Nummer über eine sichere Datenverbindung an den eops-Server.*

4. *Das eops-PIN Delivery-System sendet dem Kunden per SMS, Voice-Message oder e-Mail umgehend eine Transaktionsnummer (TAN), die nur für diese Transaktion gültig ist. Der Händler und der zu zahlende Betrag werden wiederholt.*

5. *Der Kunde bestätigt die Zahlung durch Eingabe der TAN und seiner persönlichen eops-PIN im Eingabefenster. Anschliessend wird der fällige Betrag entweder von der Kreditkarte oder vom Girokonto des Kunden abgebucht und dem Lieferanten gutgeschrieben.*

6. *Nach erfolgreicher Zahlung wird dem Händler die Zahlung bestätigt.*

7. *Der Kunde wird auf die Bestätigungsseite des Händlers geleitet.*

Auch eops-PIN ist nicht fertiggestellt, so dass keine genaueren Informationen verfügbar waren.

Eignung für eVerlage: Das System ist noch nicht ganz marktreif, aber scheint, wie eops-Mobile, theoretisch für eVerlage geeignet zu sein.

moneybox:
Die monkey AG bietet mit dem System moneybox eine Mobiltelefonlösung, die die paybox-Lösung (Bestätigung per Anruf) und die Street Cash-Lösung (Bestätigung per SMS) in einem Produkt vereint. Der Zahlungsablauf erfolgt analog zu paybox bzw. Street Cash. Dabei existieren jeweils die Lösungen „Internet 2 moneybox“ für den Interneteinkauf sowie „moneybox 2 moneybox“ für Geldtransfer zwischen zwei moneybox-Nutzern.
Dem Kunden entstehen 10 DM Grundgebühren pro Jahr, und der Händler muss ein Disagio von 2 % an die monkey AG abführen.
Die geheime, vierstellige PIN wird dem Kunden nach dessen Anmeldung über das SMS-System zugeteilt (Informationen von [22]). Der Zahlungsbereich steht noch nicht ganz fest, soll aber von 1 Cent bis zu 100.000 US$ reichen (Information per e-Mail). Allerdings scheint mir diese Zahlungsspanne unrealistisch.

Eignung für eVerlage: Auf eine e-Mail-Anfrage hin erhielt ich die Nachricht, dass das System voraussichtlich erst in einem Jahr einsatzfähig sein soll. Daher sollte über eine Integration erst nachgedacht werden, wenn das System marktreif ist.
Payitmobile:
Payitmobile und die Gesellschaft für Zahlungssysteme (GZS) wickeln mit dem Verfahren „Payitmobile“ verschiedene Zahlungsverfahren für den Kunden über Mobiltelefonab.
Dieser muss sich als erstes registrieren lassen und seine Kontodaten angeben. All diese Daten werden geprüft. Danach kann man virtuell einkaufen, wobei sich der Ablauf [23] folgendermassen gestaltet:

1. Klick auf den Button „Payitmobile “
2. Angabe der eigenen Mobiltelefonnummer
3. Eine SMS mit dem Rechnungsbetrag wird an dieses Mobiltelefon geschickt
4. Der Kunde antwortet seinerseits mit einer SMS, in welcher die persönliche Payment-PIN angegeben wird, um die Zahlung zu bestätigen.
5. Eine virtuelle Geldbörse wird geöffnet, in der man auswählen kann, mit welchem Zahlungsmittel (Kreditkarte, Lastschriftverfahren, andere Bezahlverfahren) bezahlt werden soll.
6. Die Bezahlung wird durchgeführt, wobei keine sicherheitskritischen Daten, wie Kreditkartennummern, übertragen werden.

Payitmobile arbeitet also als Datenverwalter und nutzt die Vorteile des Mobilfunknetzes, um das Problem der mangelnden Kreditkartendatensicherheit zu minimieren. Leider muss man dafür die bei Street Cash aufgezeigten Nachteile des SMS-Systems in Kauf nehmen.
Für den Kunden fallen ausser eventuellen SMS-Kosten keine weiteren Gebühren an. Der Händler muss Fixkosten und Transaktionsgebühren bezahlen, über deren Höhe auf der Payitmobile-Webseite leider keine Informationen erhältlich waren. Diese erhält man nur nach Kontakt.

Eignung für eVerlage: Payitmobile lohnt sich zur Zeit nur dann, wenn die Fixkosten für den Händler sehr gering sind, da die Nutzerzahlen noch zu klein sind.

B.4 Chipkarte

Mondex:
In einigen Ländern wie Grossbritannien, Frankreich oder Norwegen existiert das Verfahren Mondex. Es wird von ansässigen Banken unterstützt. Mondex ist eine Mischung aus Smartcard und elektronischem Geld. Der Kunde erwirbt von der Bank per Telefon oder Internet signierte virtuelle Geldscheine, welche auf der Chipkarte in einer von fünf verschiedenen Währungen gespeichert werden. Diese Zahlungseinheiten können nun beliebig an andere Personen (Händler oder Privatpersonen) weitergegeben werden. Dabei soll keine Banküberprüfung notwendig sein, so dass keinerlei Transaktionsgebühren anfallen. Auch eine Anmeldung oder Authentifizierung entfällt auf Grund des sicheren Transfers der Geldeinheiten (Informationen von [24]).
Ein Händler kann die erhaltenen Einheiten jederzeit wieder mittels Telefon oder Internet in Bargeld umwandeln lassen. Da die Zahlungseinheiten oft ihren Besitzer wechseln ist dieses Verfahren so anonym wie richtiges Bargeld.
Das Zahlungssystem ist so effizient, dass selbst Kleinstbeträge von 1 US Cent wirtschaftlich transferiert werden können (Informationen von [25]). Leider benötigt der Kunde ein Mondex-kompatibles Kartenterminal, um mittels Computer auf die Karte und somit das Geld zugreifen zu können. Es soll aber auch eine Miniversion geben, die man als point-of-sale z.B. in einem Taxi verwenden kann und welche sehr geringe Abmessungen.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Wanderung der Zahlungseinheiten
Micropayment möglich	Ja	Ab 1 US Cent
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	
Mobilität hoch	Ja	Karte transportabel, tragbares POS-Terminal
Kosten Händler gering	-	Unbekannt
Kosten Kunde gering	Ja	1.50 Pfund pro Monat [25]
Einstieg einfach	Nein	Zusatzhardware notwendig
Bedienung einfach	-	Unbekannt
Zahlungsbereich gross	Ja	1 US Cent bis landesabhängige Grenze
Akzeptanz hoch	Ja	Durch Bankunterstützung
Tranparenz hoch	-	Unbekannt

Tabelle B.2: Bewertung Mondex

Eignung für eVerlage: Zur Zeit wird das System noch nicht in Deutschland genutzt. Ausserdem ist fraglich, ob sich ein System mit elektronischem Geld wirklich durchsetzen kann. Andererseits ist die Unterstützung auch von Seiten der Banken scheinbar schon recht hoch. Sollte sich das System am Markt etablieren und in Deutschland eingeführt werden, so wäre eine Integration in eVerlage denkbar.

B.5 Telefonrechnung

eops-Call:
Mit „eops-Call“ hat die eops AG ihre Zahlungssystemfamilie um ein Abrechnungsverfahren per Telefonrechnung erweitert. Die wichtigsten Fakten von der eops Webseite [29]: *„eops-Call ist ein speziell für den Internet-Bereich entwickeltes Micropayment-System. Der Kunde benutzt einfach das Telefon, um Waren und Dienstleistungen im Wert von DM 0,49 bis DM 99,- abzurufen. Das Inkasso erfolgt anschliessend in jedem Fall über seine Telefonrechnung. eops-Call stellt somit eine Verbindung zwischen dem Anruf des Käufers und Ihrer Internetseite her. Basis dieser Verbindung ist eine kostenpflichtige Servicerufnummer. Ihr Kunde benötigt keine Software für den Verbindungsaufbau, sondern kann spontan*

mit seinem Telefon einkaufen. Daher eignet sich eops-Call besonders für LANs.“
Die Zahlung mit eops-Call funktiontiert folgendermassen: *„Der Kunde öffnet Ihre kostenpflichtige Seite im Internet, auf der der eops-Call Button installiert ist. Die Zahlung beginnt mit dem Klick des Endkunden auf den vom Händler implementierten Link zu eops-Call. Das eops-Call System zeigt dem Endkunden eine dynamisch generierte Transaktionsnummer (TAN) und eine dem gewünschten Transaktionspreis entsprechende Servicerufnummer an.*
Der Kunde ruft die angezeigte Telefonnummer an und folgt den Anweisungen des Sprachcomputers. Dieser fordert ihn auf, die TAN einzugeben. Bei richtiger Eingabe wird der Kunde anschliessend aufgefordert, mit der Maus erneut auf den eops-Call Button zu klicken. Das Telefonat ist damit abgeschlossen und er gelangt zum zahlungspflichtigen Content, bzw. hat ein Produkt aus dem Warenkorb bezahlt. Die Inkassodaten verbleiben beim Netzcarrier des Kunden, ausserdem listet die Telefonrechnung sämtliche mit eops-Call durchgeführten Transaktionen auf. “
Es werden die zwei Abrechnungsverfahren „pay-per-click“ und „pay-per-minute“ angeboten.

Eignung für eVerlage: Wegen der hohen Grundgebühr lohnt eine Integration nur bei Vorhandensein entsprechender Nutzerzahlen. Zur Zeit dürfte dieses Kriterium noch nicht erfüllt sein.

eops-Connector:
eops-Connector funktioniert ähnlich wie Net900 classic. Eine vom Kunden zu installierende Software unterbricht beim Bezahlvorgang die Internetverbindung, baut eine kostenpflichtige Telefonverbindung über eine Servicenummer auf und informiert den Kunden über die abgebuchten Beträge.
Wie bei eops-Call können DM 0.49 bis DM 99 mittels „pay-per-click“ und „pay-per-minute“ abgerechnet werden. Allerdings betragen die Transaktionskosten eines Produktes für 1 DM je nach Gesamtumsatz 50-60% [30] und liegen damit relativ hoch.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Bei eops-Call nur Servicenummernanruf
Micropayment möglich	Ja	Ab 0.49 DM
Echtzeit-Clearing	Ja	
Sicherheit hoch	Ja	
Mobilität hoch	Nein	An festen Telefonanschluss gebunden
Kosten Händler gering	Nein	Bei pay-per-click 99 DM/Monat Recht hohe Transaktionskosten
Kosten Kunde gering	Ja	
Einstieg einfach	Mittel	Anmeldung bzw. Softwareinstallation
Bedienung einfach	Ja	
Zahlungsbereich gross	Ja	Bis 99 DM
Akzeptanz hoch	Nein	
Tranparenz hoch	Mittel	Auflistung der Beträge erst bei der nächsten Telefonrechnung

Tabelle B.3: Bewertung eops-Call/eops-Connector

Eignung für eVerlage: Hier gelten die selben Bemerkungen wie zu eops-Call.

PurePay:
Die PurePay AG bietet mit dem Zahlungssystem „PurePay“ ein Micropaymentverfahren für Zahlungen zwischen 0.25 und 10.00 Euro in 0.25 Euro-Schritten an. Grundlage ist ein Gutscheinsystem und die Abrechnung per Telefonrechnung.
Der Kunde muss vor der ersten Transaktion ein Plug-In für seinen Browser (Netscape, Internet Explorer, Opera) installieren. Dabei wird zwischen ISDN-, Modem-, DSL-Zugängen und Firmennetzwerken unterschieden. Danach kann der Kunde mittels des Plug-Ins Gutscheine bei dem PurePay-Server erwerben, was man mit dem Vorgang des Geldabhebens vergleichen kann. Damit PurePay den Gegenwert der ausgegebenen Gutscheine erhält, baut das Plug-In eine kostenpflichtige Telefon- oder Faxverbindung zum PurePay-Server auf. Somit bezahlt der Nutzer die Gutscheine, welche auf der Festplatte gespeichert werden, einfach über seine Telefonrechnung (Installation und Funktionsweise aus [31]).
Stösst der Kunde beim Surfen im Internet auf einen kostenpflichtigen PurePay-Inhalt, so wird neben dem Produkt ein Button angezeigt, welcher die Preisinformationen ähnlich einem Preisschild enthält. Ist der Nutzer interessiert, so klickt er auf diesen Button, worauf das Plug-In gestartet wird. In diesem wird der zu

zahlende Betrag, der aktuelle Gutscheinstand des Nutzer, ein möglicher Ladebetrag und der neue Betrag nach der Transaktion angezeigt. Falls die Gutscheine nicht ausreichen, können direkt im Bezahlschritt noch weitere erworben werden. Nachdem der Kunde die Zahlung bestätigt hat, wird der Gutscheinwert zum Händler übertragen. Die Händlersoftwäre lässt in Echtzeit die Gültigkeit der Gutscheine bei einem PurePay-Server überprüfen. Wurde diese bestätigt, kann der Händler sofort den kostenpflichtigen Inhalt dem Nutzer zur Verfügung stellen. Sollte bei der Auslieferung die Verbindung zusammenbrechen, so kann innerhalb einer bestimmten Zeitspanne der Download wiederholt werden.
PurePay hat für den Kunden den Vorteil, dass es ein anonymes Zahlungssystem ist. PurePay weiss nicht, wofür welcher Kunde die Gutscheine verwendet, da nur die Gültigkeit geprüft wird. Weil die Gutscheine aber nicht von Person zu Person wandern können, wäre eine prinzipielle Zuordnung von Gutschein zu Person theoretisch möglich. Der Händler hingegen weiss nicht, wer der Kunde ist, da er nur anonyme Gutscheine von ihm erhält. Der Nutzer muss damit auch keine persönlichen Daten preisgeben. Der Einstieg erfolgt einfach, und es ist keine Anmeldung des Kunden notwendig. Damit ist die Möglichkeit gegeben, dass Nutzer den kostenpflichtigen Dienst spontan nutzen.
Ein Händler hat die Möglichkeit, zwischen 3 verschiedenen PurePay-Angeboten (private, business, e-commerce) zu wählen und zu wechseln. Abhängig von der Höhe der Transaktionen und dem gewählten Angebot werden die prozentualen Transaktionskosten, welche zwischen 15 und 67 % des Produktpreises betragen können, berechnet. Fixkosten entstehen keine.

Eignung für eVerlage: PurePay hat mit seinem Gutscheinsystem einen interessanten Ansatz für die Gewährleistung der Anonymität gewählt. Die Nutzung kann recht spontan erfolgen und auch Micropayment ist möglich. Ein Problem ist allerdings die Vorgabe der 0.25 Euro-Schritte.
Desweiteren scheint es Probleme bei der Anbindung des Zahlungssystems an das eVerlage-System zu geben, welche nicht ohne hohen Mehraufwand gelösst werden können. Somit wird dieses Zahlungsverfahren wohl nicht in nächster Zeit von eVerlage angeboten werden.

Anforderung	erfüllt	Bemerkung
Anonymität hoch	Ja	Komplette Anonymität gegenüber Händler (und Zahlungssystemanbieter)
Micropayment möglich	Ja	Ab 0.25 Euro
Echtzeit-Clearing	Ja	Echtzeitüberprüfung der Gutscheine
Sicherheit hoch	Mittel	Jeder, der an den Rechner gelangt, kann bezahlen. Daher Login empfehlenswert.
Mobilität hoch	Nein	Gutscheine und Plug-In auf der lokalen Festplatte
Kosten Händler gering	Mittel	Keine Fixkosten, aber evtl. hohe Transaktionsgebühren
Kosten Kunde gering	Ja	
Einstieg einfach	Ja	Kann spontan erfolgen, nur Plug-In Installation
Bedienung einfach	Ja	Keine Passwörter, Nutzerkennungen und keine Anmeldung
Zahlungsbereich gross	Nein	Nur Micropayment bis 10.00 Euro in 0.25 Euro-Schritten.
Akzeptanz hoch	-	Unbekannt
Tranparenz hoch	Ja	Die letzten Transaktionen können eingesehen werden

Tabelle B.4: Bewertung PurePay

Weitere Anbieter:
Ausser Net900, eops-Call/eops-Connector und PurePay exisitieren noch weitere Anbieter mit ähnlichen Verfahren, wie z.B. „infin MicroPayment“ von der Ingenieurgesellschaft für Informationstechnologien (http://www.infin.de), „pay.privision“ (http://www.privision.de), oder „tPay“ von Rate One (http://www.rateone.de) und dmts sowie „X-Presspay“ (http://www.x-presspay.com).
Für eVerlage wird es sinnvoll sein, das System mit den grössten Chancen auf Akzeptanz zu implementieren. Zur Zeit scheint dies noch Net900 classic zu sein.

Anhang C

Dokumentation der Programmvariablen

An dieser Stelle werden die in dem GK-API-Quelltext verwendeten Variablen dokumentiert.

Globale Variabalen:

- unsigned short ctn=1;
  ctn=CardTerminalNumber - frei wählbare Variable, ähnlich einem filedescriptor. Wird u.a. zum Senden einer APDU mit CT_data benötigt.
- unsigned short port=0;
  COM-Port/ttyx, an dem das Kartenterminal steckt. Init=0, damit das Setzen der port-Variable mittels gk_api_config erkannt wird. Verwendung für ct_init.

- p_gk_init_output init_output;
  Zeiger auf Outputstruktur von gk_api_init.
- p_gk_kt_output kt_output;
  Zeiger auf Outputstruktur für zahlungsrelevante KT-Funktionen.

- p_gk_error_output error_output;
  Zeiger auf Outputstruktur für gk_api_error.

- FILE *gksessionlogfile;
  Filepointer auf Sessionlogfile „gkapi_sessionlog.txt“.

- FILE *gktransactionlogfile;
  Filepointer auf Transaktionslogfile „gkapi_transactionlog.txt“.

- unsigned char data_block[1000];
  Puffer für reguläre Outputdaten. Auf diesen Speicherbereich zeigt *data_block der Outputstruktur (gk_api_kt_output) der zahlungsrelevanten Funktionen. In diesem Datenblock können normale Zahlungsdaten oder gesicherte Buchungsdatensätze untergebracht sein.

- unsigned char error_code[100];
  Puffer für Fehlerantwortdaten (sollten 6 Bytes sein). Auf diesen Speicherbereich zeigt *error_code der Outputstruktur (gk_api_kt_output) der zahlungsrelevanten Funktionen.

- unsigned char sad;
  Quelladresse des mit CT_data versandten Kommandos (2=Host). Kodes siehe CT-API-Spezifikation.

- unsigned char dad;
  Zieladresse des mit CT_data versandten Kommandos (1=Kartenterminal). Kodes siehe CT-API-Spezifikation.

- unsigned char command[300];
  Speicher für generierte ADPUs, welche mit CT_data dem Kartenterminal übergeben werden.

- unsigned char response[300];
  Speicher für durch CT_data zurückgelieferte Antwortdaten des Kartenterminals.

- unsigned short lenr;
  Anzahl der durch CT_data zurückgelieferten Bytes (Grösse der Antwort im response-Feld).

- unsigned short lenc;
  Grösse in Byte der im command-Feld mit CT_data übergebenen APDU

- unsigned int length_data;
  Länge der Rückgabedaten in Byte im Daten-Feld data_block einer Outputstruktur.

- unsigned int length_error;
  Länge der Rückgabedaten in Byte im Fehler-Feld error_code einer Outputstruktur.

- unsigned int ret;
  Rckgabewert der ct-api. Fehlerkodes siehe CT-API-Spezifikation.

- unsigned int i;
  Dummy-Zählervariable

Globale Typen und Variablen für CT-API:
Die folgenden Typen und Variablen dienen dem Zugriff auf die Funktionen der CT-API. Es bestehen dabei Unterschiede zwischen Windows und Solaris/Linux.

Windows:

- HINSTANCE CTAPI_DLL;
  Handle auf die ctapi.dll.

- typedef char (FAR WINAPI *CTAPI_INIT)(unsigned short, unsigned short);
  typedef char (FAR WINAPI *CTAPI_CLOSE)(unsigned short);
  typedef char (FAR WINAPI *CTAPI_DATA)(unsigned short, unsigned char *, unsigned char *, unsigned short, unsigned char *, unsigned short *, unsigned char *);
  Typdefinitionen der CT-API-Funktionen.

- CTAPI_INIT MCT_init;
  CTAPI_CLOSE MCT_close;
  CTAPI_DATA MCT_data;
  Instanzen der oben festgelegten Funktionstypen. Über diese wird auf die CT-API-Funktionen zugegriffen.

Solaris/Linux:

- void *CTAPI_DLL;
  Handle auf die libct_b1.so.

- typedef char CTAPI_INIT(unsigned short,unsigned short);
  typedef char CTAPI_CLOSE(unsigned short);
  typedef char CTAPI_DATA(unsigned short,unsigned char *,unsigned char *,unsigned short,unsigned char *,unsigned short *,unsigned char *);
  Typdefinitionen der CT-API-Funktionen.

- CTAPI_INIT *MCT_init;
  CTAPI_CLOSE *MCT_close;
  CTAPI_DATA *MCT_data;
  Zeiger auf die ebend festgelegten Funktionstypen. Über diese wird auf die CT-API-Funktionen zugegriffen.

Lokale Variablen:
Hier werden die lokalen Variablen aufgeführt.

gk_api_init:

- time_t ltime;
  Systemzeit

- struct tm *today;
  Zeitstruktur für Datum und Uhrzeit in beiden Protokolldateien.

gk_api_error:

- unsigned int responsepos;
  Feldnummer der auszuwertenden Position in der Antwortnachricht response.

fill_output_structure:

- unsigned int datapos;
  Feldnummer der aktuellen Einfügeposititon im data_block. Diese Variable enthält nach Füllen des data_block dessen Grösse in Byte und wird deshalb length_data einer Outputstruktur zugewiesen.
- unsigned int responsepos;
  Feldnummer der auszuwertenden Position in der Antwortnachricht response.