Gymnasium Brede
Bredenweg
33034 Brakel

Facharbeit
im Grundkurs Informatik

Thema: electronic cash und Chipkarten – Funktionsweise, Datenstruktur, Sicherheit

Verfasser: Dennis Groppe
Fachlehrerin: Frau Pavlu
Abgabetermin: 15.02.2001

Inhaltsverzeichnis

1. Einleitung

Meine Facharbeit widmet sich dem Thema Chipkarten und Electronic-Cash und soll Informationen über dessen Funktionsweise und Technik bereitstellen. Im Informatikunterricht haben wir das Thema „Kryptographie“, also die Verschlüsselung von Zeichen durchgenommen und verschiedene Methoden des Verschlüssels kennengelernt. Auch Chipkarten, wie man sie zum Beispiel von den sogenannten „Telefonkarten“ her kennt, und das Prinzip des Electronic Cash, kurz „E-cash“, bedienen sich, vor allem zur Bereitstellung von Datensicherheitsmaßnahmen, verschiedener krytographischen Prinzipien, die ich näher erläutern möchte.

2. Chipkarten

2.1. Was sind Chipkarten?

Chipkarten, auch „Smartcards“ genannt, sind im Allgemeinen Plastikkarten mit einer Art Microcomputer, wie man sie, zum Beispiel, als Bankkarte von seinem Bankinstitut bekommt. Chipkarten unterscheiden sich jedoch von den herkömmlichen Magnetstreifenkarten, die Daten auf einem magnetischem Streifen speichern in einem Disketten- oder Musikkassetten-ähnlichem System und von einem Kartenleser gelesen werden können, dadurch, dass sich Daten auf einem genormten Halbleiterchip, der in die Plastikkarte eingelassen und integriert wird, speichern lassen. Chipkarten werden „mit einem Chipterminal ausgelesen“. „Ein Verändern der Information ist“ jederzeit durch das Terminal „möglich“[1]. Der Halbleiterchip kann mit „Eigenintelligenz bis hin zum kompletten Microcomputer ausgestattet sein“[2]. Als Untergruppe der Chipkarten sind noch zusätzlich die sogenannten Hybridkarten im Umlauf, die Sparkassenkarte ist ein Beispiel für dieses System: Als Übergangslösung vom Magnetstreifen zur „reinen“ Chipkarte sind beide Speichermethoden auf einer Plastikkarte vereint, um die Kompatibilität zu älteren Dienstleistungseinrichtungen der Bank sicherzustellen.

2.2. Unterschiedliche Typen von Chipkarten

Chipkarten werden zu verschiedenen Zwecken benutzt:

[1] Quelle 6, „Abgrenzung zu anderen Plastikkarten“
[2] Quelle 6, „Abgrenzung zu anderen Plastikkarten“

- als Telefonkarten, auf dessen Chip das aktuelle, verbleibende Guthaben, was noch ‚ver-telefoniert' werden kann, gespeichert wird und dann vom Leseschlitz der Telefonanlage in Telefonzellen ausgelesen und dann nach dem Gespräch aktualisiert wird
- als Krankenkarten, die von den Krankenkassen an ihre Versicherten vergeben werden und auf dessen Speicherchips die Kundendaten, Gültigkeit und Versichertenstatus festgehalten werden. Dadurch können zum Beispiel in Sekundenschnelle die Patientendaten in die Datenverarbeitung des behandelnden Arztes übertragen werden. Diese Karten wurden im Jahre 1993 als Ersatz für den Krankenschein[3] eingeführt.
- als Identifikationskarten („ID-Cards") in Sicherheitssystemen. Nur wer im Besitz der Chipkarte ist, kann vom Computer identifiziert werden
- im Bereich des E-cash, zum Beispiel bei Geldkarten. Hier wird auf dem Chip der aktuell geladene Geldbetrag und die letzten 10 Buchungen mitsamt Buchungsdatum gespeichert (dazu mehr in Kapitel 3.4.).

2.3. Aufbau der Hardware, Abmessungen und Normen

Chipkarten sind auf die Maße 85.6 mm × 54 mm × 0.76 mm normiert, wobei die „Chipfläche auf 25 mm² beschränkt ist"[4]. Der Chip ist ein sogenannter „Single Chip Computer", der bis zu 2 Millionen Instruktionen pro Sekunde (kurz: 2 MIPS) mit seiner 8-bit-Central Processor Unit (CPU) berechnen kann. Ferner verfügt das System zur Datenspeicherung über Arbeitsspeicher (RAM), Programmspeicher (ROM), in dem das Programm mit den Aufgaben des jeweiligen Chips gespeichert ist, ein Miniatur-Betriebssysteme mit Speichergrössen im kByte-Bereich (6-32kByte)[5], und über einen immer wieder beschreibbaren Datenspeicher (EEPROM), auf dem die Ergebnisse und Daten aus den CPU – Berechnungen gespeichert sind. Weiter sind auch die Anordnung der Kontakte, die Datenstrukturen (siehe 2.4.) und die Kommunikationsprotokolle genormt. Das „weltweit am weitesten" verbreitete Übertragungsprotokol heißt „T=0" und wird vor allem für die GSM – Mobiltelefone , die ja auch Chipkarten (SIMs) benötigen, verwendet[6].

2.4. Das Dateisystem der Chipkarten

Die Dateien des Chipkarten-Betriebssystems „sind hierarchisch organisiert"[7]. Den Ursprung des Dateisystems bildet das „Master File" (MF). In diesem MF können alle Daten vorhanden

[3] Quelle 4, „Anwendung"
[4] Quelle 4, „Normierung"
[5] Quelle 5, Kapitel 4.1.
[6] Quelle 5, Kapitel 4.2.
[7] Quelle 5, Kapitel 4.1., „Filesystem"

sein, die alle Anwendungen und Funktionen des Chips nutzen, z.B. „Daten über den Karteninhaber, Seriennummer, Schlüssel“[8] Diese Daten sind wiederum in Elementary Files (EF) im MF abgelegt. Neben dem Master File gibt es die dem MF untergeordneten „Dedicated Files“ (DF), „die wiederum mit ihren EFs, die in den DFs wie in einem „herkömmlichen Verzeichnis“ gespeichert sind[9] und mit ihren Funktionen die Anwendungen in einer Karte repräsentieren“[10]. Dabei können für jedes DF „seperate Sicherheitsfunktionen definiert werden“[11]. EFs kommen also sowohl im MF als auch in den DFs vor, dabei werden Internal Elementary Files (IEF) und Working Elementary Files (WEF) unterschieden. „Die Daten eines Internal Elementary Files unterliegen der Zugriffskontrolle des Betriebssystems. Ein direkter Zugriff mittels des Kartenterminals ist nicht möglich“[12].
„Ein Internal Elementary Files enthält z.B. anwendungsbezogene Paßwörter und Schlüssel. Die Working Elementary Files enthalten die Nutzdaten einer Anwendung. Die Daten können nach obligatorischer Authentifizierung unter Verwendung eines Internal Elementary Files unter Berücksichtigung von Sicherheitsattributen gelesen und/oder verändert werden.“[13]

2.5. Sicherheitsmaßnahmen am Beispiel der Bankkarte

Bei vielen Arten der Chipkarte müssen vertrauliche Daten oder auch geheime Schlüssel auf dem Chip gespeichert werden. Daher bietet die Chipkarte einige Schutzmaßnahmen für diese Daten an, die vor allem vor den drei Grundbedrohungen zu schützen, und zwar vom Zeitpunkt der Herstellung bis zum Ablauf der Gültigkeit der Karte:

1. Verlust der Vertraulichkeit durch Ausspähen bzw. unbefugte Preisgabe von Programmen und Daten, etwa Schlüssel, PIN(Personal Identifikation Number) und Anwenderdaten
2. Verlust der Integrität durch Manipulation bzw. die nicht autorisierte Veränderung von Informationen wie Identifikationsnummer und Zählerstand
3. Verlust der Verfügbarkeit durch unberechtigte Vorenthaltung von Informationen bzw. Beeinträchtigung der Funktionalität eines Systems. [14]

2.5.1. Grundschutzmaßnahmen

[8] Quelle 7, Kapitel III.2.1.
[9] Quelle 5, Kapitel 4.1. „Filesystem“
[10] Quelle 7, Kapitel III.2.1.
[11] Quelle 7, Kapitel III.2.1.
[12] Quelle 5, Kapitel 4.1. „Filesystem“
[13] Quelle 5, Kapitel 4.1. „Filesystem“
[14] Quelle 5, Kapitel 4.1.

- Ausstatten der Karte mit „Authentifizierungsmerkmalen“[15] wie Unterschrift, Foto, Hologramme
- Absicherung der Verbindung zwischen Kartenterminal und Karte gegen Abhören und Auslesen der ausgetauschten Daten
- Kontrollmöglichkeiten zur Echtheit und Gültigkeit der Karte
- Software und kryptographische Verfahren wie im Folgenden beschrieben

2.5.2. Betriebssystem

Bei Chipkarten werden sogenannte „Mikrocontroller“ und „Sicherheitsbetriebssysteme“[16], die im Gegensatz zu PC-Betriebssystemen nicht offen einsehbar gestaltet sind, verwendet. Diese Sicherheitsbetriebssysteme weisen zunächst schon einmal Sicherheitshardware auf Basis von Singlechip-Microcontrollern auf und verfügen über Algorithmen zum sicheren Speichern von Daten und ferner über kryptographische Verfahren zur Ent- und Verschlüsselung von vertraulichen Daten und Schlüsseln, zum Beispiel der PIN (Personal Identification Number = persönliche Indentifikationsnummer), die man vor allem für den Zugriff auf Geldautomaten von Banken benötigt. Da des Chipkartenbetriebssystem im ROM-Speicherbereich (siehe 2.2.) untergebracht ist, der keine Änderungen zuläßt (ROM = read only memory, d. h.: Daten können nur gelesen, nicht geschrieben werden) ist es zusätzlich gegen Angriffe auf die internen Daten des Chips geschützt. Zusätzlich kann im Betriebssystem eine Zugriffslogik integriert sein, die für die einzelnen Datenfelder bestimmen kann, „welche Applikation (oder welche Stelle) die Daten lesen, ändern oder neu schreiben darf (Speicherschutz)“[17]

2.5.3. physikalische Sicherheit des Chips

Als zusätzlicher Sicherheitsmechanismus macht es der mechanische Aufbau der Chipkarte unmöglich, den Chip mit einem Mikroskop zu analysieren oder gar den Chip aus der Karte zu entfernen, ohne dass dieser beim Herauslösen zerstört wird und damit alle Daten auf ihm vernichtet werden.

2.5.4. verschiedene Kryptoverfahren

Durch das Anwenden vor allem des Ersten der im folgenden beschriebenen kryptographischen Verfahren sind viele zusätzliche Sicherungensmaßnahmen auf der

---

[15] Quelle 5, Kapitel 4.3., „Mindestanforderungen“
[16] Quelle 5, Kapitel 4.1.
[17] Quelle 6, Interner Aufbau und interne Sicherheit

Chipkarte möglich geworden, da durch diese erst Schlüsseldaten und verschlüsselte Informationen ohne Risiko auf dem Chip gespeichert werden können.

a) Symmetrische Kryptoverfahren mit Hilfe eines sicheren Übertragungskanals
Diese Verfahren benutzt sowohl zum Ver- als auch zum Entschlüsseln nur einen Schlüssel, dadurch entsteht der Nachteil, das jeder, der entschlüsseln kann, auch verschlüsseln kann. Also muss gewährleistet sein, dass die beiden beteiligten Personen, die mit Hilfe eines symmetrischen Kryptoverfahren Daten austauschen wollen, zum einen den gemeinsamen Schlüssel nicht an Dritte weitergeben, und zum anderen zum Schlüsselaustausch über einen „zweiten, sicheren Übertragungskanal“[18] laufen lassen, z.B. über ein DES-System (Data Encryption Standard, siehe hierzu Anhang A). Das symmetrische Kryptoverfahren ist durch seine einfache Implementierung und seine Schnelligkeit („Authentifizierung in einigen Millisekunden“[19]) das einzige, welches auf heutigen Chips realisierbar ist.

b) asymmetrische Kryptoverfahren
Bei asymmetrischen Kryptoverfahren bedient man sich zweier Schlüssel, E zum Verschlüsseln und D zum Entschlüsseln, die unabhängig voneinander sind, d.h. man kann von E nicht auf D schließen und andersherum, daher kann der Schlüsselaustausch auch auf ungesicherten Wegen ablaufen. Man muss nur darauf achten, dass nicht beide Personen ihre Schlüssel preisgeben. Bekanntestes Beispiel für asymmetrischen Verschlüsselung ist das RSA – Verfahren (mehr dazu in Anhang B). Im Gegensatz zum symmetrischen Verfahren ist dieses jedoch bis jetzt nur sehr langsam realisiert worden (wenige Kilobit pro Sekunde) und ist „relativ schwer zu implementieren“[20], d.h. die Umsetzung für einen Chip ist zu umständlich und wenig nutzbar.

c) Zero – Knowledge – Verfahren
Zur Erklärung dieses Verfahrens gibt es eine verbreitete Zusammenfassung, die ich in zwei meiner Quellen exakt gleich vorgefunden habe:

> Zero-Knowledge-Verfahren eignen sich dazu, einen Kommunikationspartner davon zu überzeugen, daß man ein Geheimnis G kennt, ohne irgendetwas über G mitzuteilen. Dazu bedient man sich zweier Mengen von geheimen und öffentlichen Schlüsseln. Mit einem mehrfach wiederholten Challenge-Response-Protokoll wird authentifiziert.
> Man kann sich die Funktion von Zero-Knowledge-Verfahren mit folgendem Versuchsaufbau vergegenwärtigen. Er besteht aus einem Raum, der durch eine Wand in

[18] Quelle 6, „Typen von Kryptoverfahren“
[19] Quelle 5, Kapitel 4.3., „Kryptoverfahren“
[20] Quelle 6, „Typen von Kryptoverfahren“

zwei Teile getrennt ist. Die Trennwand enthält eine magische Tür, die nur mit Kenntnis eines Geheimnisses (oder Schlüssel) geöffnet werden kann.

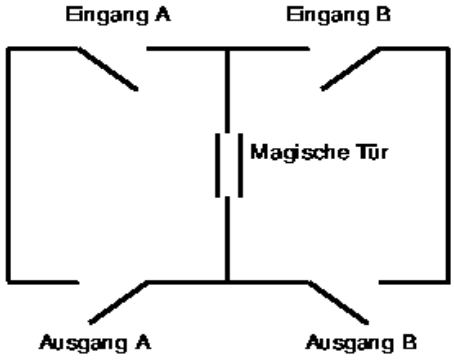


Abb.1

Angenommen Anton will Berta davon überzeugen, daß er die magische Tür öffnen kann, ohne daß er ihr sein Geheimnis (die Kenntnis, wie die Tür geöffnet werden kann) mitteilt. Dazu stellt Berta sich vor die Ausgangstüren. Anton betritt einen der beiden Teilräume. Nun ruft Berta ihm zu, zu welcher der beiden Ausgangstüren er herauskommen soll (A oder B).
Kommt Anton zur "falschen" Tür heraus, kennt er offensichtlich das Geheimnis der magischen Tür nicht. Kommt er dagegen zur richtigen Tür heraus, gibt es zwei Möglichkeiten: Entweder er war von vornherein im richtigen Teilraum oder er kennt das Geheimnis. Berta kann in diesem Fall mit einer Wahrscheinlichkeit von 50% annehmen, daß Anton das Geheimnis kennt. Durch mehrfaches Wiederholen des Versuchs kann diese Wahrscheinlichkeit auf einen sicheren Wert erhöht werden.
Zero-Knowledge-Verfahren sind sehr leicht zu implementieren und relativ schnell. Sie eignen sich ebenfalls für die Generierung elektronischer Unterschriften. Allerdings sind für einen Authentifizierungsvorgang relativ viele Kommunikationsschritte notwendig. Zudem eignen sich Zero-Knowledge-Verfahren nicht zum Verschlüsseln von Daten.[21]

2.6. Steht meine PIN auf der Bankkarte ?

Ein klares „Nein“, denn das wäre viel zu gefährlich für die Datensicherheit, da man die „Personal Identification Number“ einfach so vom Chip oder Magnetstreifen mit entsprechenden Lesegeräten ablesen könnte, zum Geldautomaten gehen und auf dem Konto des tatsächlichen Besitzers der Karte Geld abheben kann, weil auch er sich wie jener mit der PIN als zugriffsberechtigte Person für das Konto identifizieren kann. Auch hier kommt wieder die Kryptographie zu Hilfe: Auf dem Magnetstreifen der Geldkarte sind folgende Daten gespeichert: Bankleitzahl, Kontonummer, Verfallsdatum der Karte und ein Fehlbedienungszähler[22]. Hiervon sind das Verfallsdatum und der Fehlbedienungszähler dazu da, „die Gültigkeit der Karte zu überprüfen“. Die Karte wird vom Automat als ungültig erkannt, „wenn das Verfallsdatum abgelaufen ist“ und/oder „der Fehlbedienungszähler auf 0 steht“[23], und das tut dieser immer dann, wenn dreimal hintereinander die falsche PIN am Geldautomaten eingegeben wurde. Am Anfang einer jeder ‚Geldautomaten – Sitzung‘ steht

[21] Quelle 5, Kapitel 4.3., „Kryptoverfahren“ UND Quelle 6, „Typen von Kryptoverfahren“
[22] vgl. Quelle 2, Seite 46
[23] Quelle 2, Seite 46

der Zähler also auf 3 und wenn der Betreffende die PIN nicht weiß, weil es entweder nicht seine Karte ist (Diebstahl) oder sie einfach vergessen hat, hat er oder sie „höchstens drei sukzessive Fehlversuche“[24] bis die Karte als ungültig deklariert wird. Wenn die PIN bei der zum Beispiel zweiten Eingabe dann richtig ist, wird der Zähler wieder auf 3 gesetzt und man erhält Zugang zu seinem Konto.

Der Automat überprüft die Richtigkeit der Geheimzahl nun auf folgende Weise: Da es viel zu unsicher wäre, auf dem Bankrechner einfach alle Kontonummern mit ihren PINs / Geheimnummern zu speichern, da sich ja dann die Bankangestellten, die ja auch nicht an mein persönliches Konto heran dürfen sollen, Zugriff auf alle diese Nummern haben. Außerdem würde dadurch eine riesige Datentabelle entstehen, die täglich aktualisiert werden müsste, immer wenn eine Karte ungültig oder neu ausgestellt wird. Im Idealfall sind also die Geheimzahlen unlesbar oder gar nicht auf den Bankrechnern gespeichert. Und das geschieht folgendermaßen: „Der Automat berechnet die Geheimzahl aus den Daten des Magnetstreifens (Bankleitzahl, Kontonummer, Verfallsdatum)“[25]. Unter Zuhilfenahme des DES (siehe 2.5.3. und Anhang A) und einem geheimen Schlüssel des Bankinstitutes wird dann aus den „‚offenen Daten‘ des Magnetstreifens die zugehörige Geheimzahl berechnet“[26].

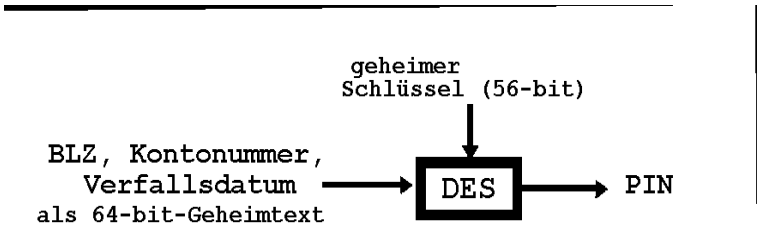


Abb.2

Diese Berechnung wird vor Abgabe der Karte an den Kunden das erste Mal durchgeführt, der erhält seine PIN dann in einem Schreiben und muss sie sich ab jetzt merken und dafür sorgen, dass niemand anderes an die Nummer herankommt. Und ab diesem Zeitpunkt wird die „Berechnung bei jeder Verwendung der Karte am Geldautomaten durchgeführt“[27]. Der Geldautomat vergleicht dann die mit dem DES errechnete Zahl mit der eingegebenen und entscheidet dann über die Zugangsberechtigung zum Konto des Karteninhabers. Daraus folgt, dass bei dem angewandten Verfahren die PIN wie gefordert gar nicht auf den Bankrechnern vermerkt sein muss, sondern vielmehr nur der geheime Schlüssel, da sich diese Zahl aus der individuellen Bankkarte und den Berechnungen des DES – Algorithmus ergibt. Weitere Sicherheit bietet der Geldautomat übrigens auch bei der Eingabe der PIN und der dann

[24] Quelle 2, Seite 46
[25] Quelle 2, Seite 47
[26] Quelle 2, Seite 47
[27] Quelle 2, Seite 47

folgenden Übertragung der Klartext – PIN dadurch, dass die Daten „nur in verschlüsselter Form übertragen werden“[28]. Zusammenfassend kann man sagen, dass durch die Kryptographie auch das Geldautomatensystem ein sehr sicheres ist, vorausgesetzt, der Bankkunde vergißt seine PIN nicht und der DES wird nicht geknackt, was bisher noch nicht möglich ist (siehe Anhang A).

3. electronic cash

3.1. Was ist E-cash?

„E-cash“ ist die Kurzform für „Electronic Cash, zu deutsch „elektronisches Bargeld“, „in elektronische Form umgewandeltes Bargeld“[29]. Hierzu gehören die Geldkarte, eine spezielle Form der Chipkarte, bargeldlose Überweisungen und das Bezahlen per Bank- oder Kreditkarte (vor allem beim Online-Shopping). Beim E-cash werden also Geldbeträge verschoben, zum Beispiel vom Kunde über die Kreditkarte zum Händler, ohne dass dabei physisch vorhandenes Bargeld (Münzen und Geldscheine) benötigt und bewegt wird.

3.2. Die Schwächen physischen Geldes – Vorteile von E-cash

Durch die ständige Vergrößerung der Wirtschaft der Industriestaaten werden heutzutage immer größere Mengen Geld bewegt. Dadurch stößt das physische, herkömmliche Geld an seine Grenzen:

- Transportproblem: Es ist nicht möglich, Millionenbeträge immer hin und her zu bringen, zum Beispiel bei Börsenhandlungen. dadurch entstehen nämlich immense Transferkosten und Lagerprobleme.
- Aufwand zur Bereitstellung: physisches Geld muss erst gedruckt werden, was sich bei bargeldlosen Zahlungsverkehr erübrigt.
- Gefahr des Raubes und der Fälschung: „Physisch vorhandenes Geld muss an sicheren Orten aufbewahrt werden“[30] und zusätzlich vor Fälschungen und Kopien geschützt werden.

Diese Nachteile des herkömmlichen Bargeld liegen bei der Bezahlung mit E-cash nicht vor. Besonders die Einsparung einer großen Menge Verarbeitungskosten pro Transaktion sind ein

[28] Quelle 1, Seite 313
[29] Quelle 3, II B „Wichtige Begriffe“,1)
[30] Quelle 3, II A

Vorteil für das E-cash System, den Unterschied zur Barzahlung und anderen Zahlungsarten macht Abb.3 deutlich:

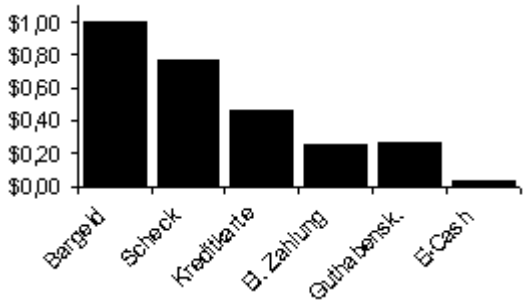


Abb.3

3.3. Sicherheit bei E-cash – Zahlungen

Wie auch bei (Geld-)Chipkarten, die, wie bereits erwähnt, ein Teil des E-cash – Prinzipes sind, werden für alle Bereiche des bargeldlosen Zahlungsverkehr Sicherheitsmechanismen und Authentifizierungsmaßnahmen benötigt, um vor allem dem Problem des sogenannten ‚double spending', „die größte technische Hürde für elektronisches Geld"[31], entgegenzugehen: Da das elektronische Geld „nur als Bit-Folge vorhanden" ist, wäre es möglich, identische Kopien dieser Zeichenfolgen anlegen und „sowohl mit dem original als auch mit der Kopie bezahlen" zu können[32]. Das wird mit folgenden Methoden versucht, unmöglich zu machen:

- bei Chipkarten ist auf dem „manipulationssicheren Kartenchip" „ein Mechanismus realisiert"[33], der das double spending unmöglich macht.
- bei Online-Zahlungen beim Händler fragt das Kartenlesegerät bei der Bank an, ob das vorliegende E-cash, genauer gesagt die Bit-Folge die das elektronische Geld darstellt, des Kunden zuvor schon einmal eingereicht worden ist. Um zu verhindern, dass dadurch die Bank erfahren kann, wer bei welchem Händler nun gerade einkauft und um die „Untraceability" (=Nicht-Zurückverfolgbarkeit) und Anonymität des Kunden zu bewahren, verwendet man zusätzlich das Konzept der ‚blind signature' (siehe Kapitel 3.3.2.).
- bei Offline-Systemen, also solchen ohne eine permanente Verbindung zur Bank-Datenbank, wird, wird auch eine Liste mit schon eingereichten E-cash-Banknoten geführt, wobei diese bei jedem Zahlungsvorgang eine weitere Signatur erhalten und sobald double spending versucht wird, die Banknote durch die Signaturen zurückverfolgt werden kann.

[31] Quelle 3, II B 3)
[32] Quelle 3, II B 3)
[33] Quelle 3, II B 3)

3.3.1. Verschlüsselung

Die genannten Lösungen zur Verhinderung von double spending „erfordern, dass jedes E-cash – Geldstück eine fälschungssichere und einmalige Seriennummer besitzt, die eindeutig einen Zahler und einem Empfänger zugeordnet werden kann“[34]. Deshalb werden auch hier Verschlüsselungsverfahren benutzt, wie sie im Zusammenhang mit Chipkarten weiter oben schon einmal genannt wurden (siehe 2.5.4.).

Man unterscheidet zwei Arten der Verschlüsselung:

1. die symmetrische Verschlüsselung mit ‚Private Key‘

Hier wird genau wie oben beschrieben derselbe Schlüssel (der sogenannte ‚private key‘) zum Ver- und Entschlüsseln verwendet, und es muss gewährleistet sein, dass der Schlüssel nicht an Unbeteiligte des Austausches weitergegeben wird und dazu benötigt man ein sicheren Schlüssel – Übermittlungsweg

2. die Verschlüsselung mit asymmetrischen ‚public key‘

Wie bereits bekannt, werden hierbei zwei unabhängige Schlüssel benutzt, einen zum Verschlüsseln (der öffentlichen Schlüssel ‚public key‘) und einen zweiten zum Entschlüsseln (der geheime Schlüssel ‚private key‘, der wie bei der symmetrischen Verschlüsselung nicht an Unbeteiligte weitergegeben werden darf). Die geheime Nachricht (in diesem Fall die sicherheitsrelevante Seriennummer des elektronischen Geldes) wird also „vom Sender mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und können nur von diesem mit seinem privaten Schlüssel entschlüsselt werden“[35].

3.3.2. Signierung

Als weitere Sicherheitsmaßnahme beim E-cash ist es möglich, sowohl verschlüsselte als auch unverschlüsselte Daten mit einem eigenen geheimen Schlüssel zu signieren, sodass die Nachricht durch den Schlüssel nicht verschlüsselt wird, jeder kann sie weiterhin entschlüsseln oder einsehen, jedoch durch den individuellen Signaturschlüssel eindeutig dem Teilnehmer des Zahlungsverkehrs zugeordnet werden kann. Signatur ist also ein Mittel zur Authentifikation des Senders im Zahlungsverkehr und dient dazu, dass die Sendung weder verfälscht werden kann noch dass der Sender seine Sendung später abstreiten kann („Unwiederrufbarkeit“[36]).

Eine Weiterentwicklung sind ‚blind signatures‘ (verdeckte Unterschriften), die eine Möglichkeit der Anonymisierung schaffen: Der Kunde kann sein elektronisches Geld mit

[34] Quelle 3, II B 4)
[35] Quelle 3, II B 4)
[36] Quelle 3, II B 4)

einem frei wählbaren Wert verschlüsseln, die Bank erfährt diese Werte aber nicht und gibt es dem Kunden, nach einer Belastung dessen Kontos mit den Nominalwerten der Schlüsselnummern, unverändert, mit Möglichkeit des Ausgebens, zurück. Wenn der Kunde nun seinen Schlüssel bekanntgibt, mit dem das E-cash immer noch markiert ist, könnte er alle Banknoten von ihm von der Bank identifizieren und notfalls sperren lassen“[37]

3.4. Das Dateisystem der Geldkarte simuliert mit Turbo Pascal

Da wir im Unterricht auch viel mit der Turbo-Pascal – Programmiersprache gearbeitet haben, habe ich ein Programm in Pascal geschrieben, welches das Dateisystem und die Datenspeicherung auf der Geldkarte nachstellt. Über ein Menü kann man, wie an einem entsprechenden Terminal, elektronisches Geld aufladen und abbuchen, die Werte der letzten 18 (bedingt durch die Größe des DOS-Fensters)Transaktionen anzeigen lassen (werden auch bei der ‚echten‘ Geldkarte gespeichert, jedoch nur bis zum 10. Eintrag) und den aktuellen Ladungszustand ausgeben lassen.

```
PROGRAM geldkarte;
USES crt, reaqueue;
VAR q:queue;
VAR e:CHAR;
   g:REAL;

PROCEDURE speicherausgabe (q:queue) ;
 VAR i:INTEGER; w:REAL;
 BEGIN
   REPEAT
   q_front(q,w);
   writeln(w:3:2);
   q_out(q);
   UNTIL q_empty (q) = TRUE
 END;


PROCEDURE menue (VAR  eingabe:CHAR;VAR g : REAL);
VAR  b:REAL; eintrag:REAL;
 BEGIN

 writeln ('OPTIONSMENUE');
 writeln (' (A) um die Geldkarte aufzuladen');
 writeln (' (E) um die Geldkarte zu entladen');
 writeln (' (T) um die letzten gespeicherten Transaktionen anzuzeigen');
 writeln (' (K) um den aktuellen Kontostand anzuzeigen');
 writeln (' (X) um das Programm zu beenden');
 readln (eingabe);

 CASE eingabe OF
  'A','a': BEGIN
         writeln (' Wieviel DEM sollen aufgeladen werden? ');
         readln (b);
```

[37] Quelle 3, II B 4)

```
          g:=g+b;
          eintrag:=0+b;
          q_in (q,eintrag);
          writeln ('Auf der Geldkarte sind nun ',g:3:2,' DEM');
         END;
 'E','e': BEGIN
          writeln ('Wieviel DEM sollen entladen werden?');
          readln (b);
          g:=g-b;
          IF g < 0 {Schutz vor Entladung unter 0.00 DEM}
           THEN
            BEGIN
             writeln ('Soviel Geld ist zur Zeit nicht auf der Geldkarte geladen!');
             writeln ('bitte wählen Sie einen kleineren Betrag!');
             g:=g+b;
             writeln ('Auf der Geldkarte sind aktuell ',g:3:2,' DEM');
            END
          ELSE
           BEGIN
            eintrag:=0-b;
            q_in (q,eintrag);
            writeln ('Auf der Geldkarte sind nun ',g:3:2,' DEM');
           END;
         END;
 't','T': BEGIN
          speicherausgabe (q);
         END;
 'k','K': BEGIN
          writeln ('Aktuell sind ',g:3:2,' DEM auf der Geldkarte');
         END;
 'x','X': BEGIN
         END;
 END {of CASE} ;
END;


BEGIN {Hauptprogramm}
 q_create (q);
 g:= 0; {Kontostand wird zu Beginn auf 0 gesetzt}
REPEAT
 menue (e,g);
UNTIL e in ['x', 'X'];
END.
```

Dazu wird noch die Unit (TPU) ‚Reaqueue' benötigt, die Befehle zur Queue q enthält:

```
UNIT reaqueue;
INTERFACE
TYPE elementtyp    = REAL;
    zeiger       = ^queueeintrag;
    queueeintrag  = RECORD
                     inhalt : elementtyp;
                     next   : zeiger
                    END;
    queue         = RECORD
                     anfang, ende : zeiger
                    END;

PROCEDURE q_create (VAR q : queue);
PROCEDURE q_front (q : queue; VAR front : elementtyp);
```

```
PROCEDURE q_in     (VAR q : queue; e : elementtyp);
PROCEDURE q_out    (VAR q : queue);
FUNCTION q_empty   (q:queue) : BOOLEAN;
FUNCTION q_full (q:queue) : BOOLEAN;

IMPLEMENTATION

PROCEDURE q_create (VAR q : queue) ;
BEGIN
  q.anfang := NIL;
  q.ende := NIL
END;

PROCEDURE q_front (q : queue; VAR front : elementtyp) ;
BEGIN
  front := q.anfang^.inhalt
END;

PROCEDURE q_in (VAR q : queue; e : elementtyp) ;
VAR hilf : zeiger;
BEGIN
  new (hilf);
  hilf^.inhalt := e;
  hilf^.next := NIL;
  IF q.anfang = NIL
    THEN q.anfang := hilf
    ELSE q.ende^.next := hilf;
  q.ende := hilf
END;

PROCEDURE q_out (VAR q : queue) ;
BEGIN
  q.anfang:= q.anfang^.next
END;

FUNCTION q_empty  (q:queue) : BOOLEAN ;
BEGIN
  q_empty := (q.anfang = NIL )
END;

FUNCTION q_full (q :queue) : BOOLEAN  ;
VAR hilf : zeiger;
    zaehler : INTEGER;
BEGIN
  hilf := q.anfang;
  zaehler := 1;
  while not (hilf = q.ende) DO
  BEGIN
    hilf := hilf^.next;
    zaehler := zaehler +1;
  END;
  q_full := (zaehler >= 10);
  writeln(zaehler)
END;


BEGIN
(* Initialisierung *)
END.
```

4. Zusammenfassung und ein Blick in die Zukunft des E-cash

Wie man an der Chipkarte und im besonderen an der Geld- und Bankkarte sehen kann, ist eine Menge Aufwand, die Benutzung verschiedener Kryptoverfahren und Sicherheitsmaßnahmen, nötig, um einen sicheren Ablauf dieser Art finanzieller Aktionen zu gewährleisten. Solange aber die beiden wichtigsten Verschlüsselungsalgorithmen, DES als symmetrisches und RSA als asymmetrisches Verfahren, noch ungeknackt sind (was aller Voraussicht nach auch erst einmal so bleiben wird, siehe Anhang A), ist die Sicherheit für den ‚normalen' Bankkunden gewährleistet. Und trotz einiger ungeklärter Fragen zum E-cash, vor allem bezüglich des „möglichen Steuerschlupfwinkel Electronic Commerce"[38], d.h. es gibt derzeit keine Gesetze zur Versteuerung im Internet – Handel (= ‚Electronic Commerce'), und der Frage, wer elektronische Geld herausgeben darf und wie sich „die Regierungen gegenüber den unkontrollierbaren grenzenlosen Kapitalströmen"[39] verhalten sollen, ist man sicher, dass das Wachstum des E-cash „rasant ist und bleiben wird", und sich ferner „unser gesamter Wirtschaftskreislauf durch elektronisches Geld verändern wird"[40]. Und auch die Chipkarte im einzelnen wird in Zukunft immer häufiger, wie schon jetzt, vor allem als „ein bequemes, schnelles und multifunktionales Zahlungsmittel"[41] genutzt werden, PCs sollen mit eingebauten Smartcard – Lesegeräten ausgerüstet werden und zusätzlich werden Versuche zur „Portierung der" betriebssystemunabhängigen „Programmiersprache Java auf Smartcards" gemacht[42].

Zusammenfassend kann man also sagen, dass der Markt des E-cash und der Chipkarten, ein stark wachsende Wirtschaftssektor ist und auch in Zukunft sein wird, und seine Sicherheit auch in Zukunft vor allem durch den Einsatz von erprobten und zuverlässigen kryptographischen Verfahren gewährleistet ist.

---

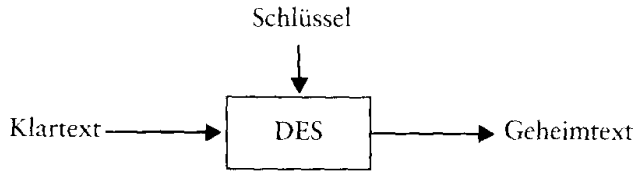
[38] Quelle 3, V 6
[39] Quelle 3, V 6
[40] Quelle 3, V 1
[41] Quelle 3, V 5
[42] Quelle 3, V 5

5. Quellenverzeichnis

1) Kippenhahn, Rudolf: Verschlüsselte Botschaften: Geheimschrift. Enigma und Chipkarte. Hamburg 1997
2) Beutelspacher, Albrecht: Geheimsprachen. Geschichte und Techniken. München 1997
3) Rott, Christian: E-Cash Bestandsaufnahme 1997/98: Zahlungssysteme im Online-Handel, Smart Cards, Micropayment. http://stud1.tuwien.ac.at/~e8525020/preecash.html
4) Informationen über Chipkarten. http://www.uni-mainz.de/~pommeren/DSVorlesung96/Chipkarte.html
5) Deuschle, Jürgen: Funktionsweise und Marktübersicht von Chipkarten. http://rhlx01.rz.fht-esslingen.de/projects/krypto/chip/chip.html
6) Dieterich, Roland: SmartCards - Grundlagen, Technik, Sicherheitsaspekte. http://fit.gmd.de/~cactus/smartcards.html
7) Anforderungen zur informationstechnischen Sicherheit bei Chipkarten, herausgegeben vom Datenschutzbeauftragten der Stadt Hamburg. http://www.hamburg.de/Behoerden/HmbDSB/Material/chip_o.htm
8) Kryptographieprojekt des Jahrgang 12 (99/00) des Gymnasium Brede. http://www.schulnetz.net/brakel/brede/krypto/index.htm

Anhang A – Der Data Encryption Standard DES

Der DES ist der „populärste heutige Algorithmus“, „der kommerziell mit Abstand am häufigsten eingesetzt“ und implementiert wird[43]. DES wurde 1976 veröffentlicht und basiert auf dem „lucifer“-Algorithmus von IBM.



Die Grobstruktur des DES

Abb.4

Der DES verschlüsselt „jeweils einen Block von 64 Bits auf einen Schlag“[44] mit Hilfe eines 56-bit-Schlüssels, wie schematisch in Abb.2 dargestellt. Daraus folgt, dass der Klartext als Bit – Folge vorliegen muss und dann in 64-bit-Stücke geteilt wird und dann werden die Stücke nacheinander verschlüsselt. Obwohl der Algorithmus zur DES – Verschlüsselung vollständig veröffentlicht wurde, ist er sehr kompliziert und bisher, seit seiner Veröffentlichung auch ungeknackt. das liegt zum einen daran, dass bisher kein Kryptologe eine Schwachstelle das Algorithmus gefunden hat, zum anderen ist die „Anzahl der“ möglichen „Schlüssel“ „sehr groß“. ‚56-bit-Schlüssel‘ heißt im Klartext, dass es $2^{56} = 72.057.594.037.927.936$ Schlüssel gibt, von denen ein einziger bei einer DES – Anwendung benutzt wird. Und genau diesen unter den über 72 Billiarden möglichen zu finden, ist „noch schwieriger, als die berühmte Nadel im Heuhaufen zu finden“[45].

Anhang B – Kurze Erklärung des RSA – Verfahrens [46]

Man benötigt ein Schloss mit drei Schlüssellöchern. Es gibt einen großen Schlüssel N und zwei kleine Schlüssel E und D. Wenn man das Schloss nun abschließen will, benötigt man den großen Schlüssel N und einen der kleinen Schlüssel, z.B. E. Zum Aufschließen braucht man nun wieder den großen Schlüssel N und den anderen der beiden kleinen Schlüssel, in diesem Fall Schlüssel D. In der Praxis sieht das nun so aus: Man bekommt drei Schlüssel, N, E und D, von denen man den Schlüssel D für sich behält und niemandem anderen gibt oder weitererzählt. Die beiden Schlüssel N und E stellt man der Öffentlichkeit zur Verfügung, so

[43] Quelle 2, Seite 41
[44] Quelle 2, Seite 42
[45] Quelle 2, Seite 43
[46] Quelle 8, „RSA – Verfahren“ – Frame , zitiert mit leichten Veränderungen

dass man von jedem Nachrichten bekommen kann. Damit man nicht alles wieder rückgängig machen kann, hat man das asymmetrisches Verschlüsselungsverfahren RSA (nach seinen Entwicklern Ronald Rivest, Adi Shamir und Leonard Adleman) entwickelt.
Die Schlüssel von denen wir hier sprechen sind in der Praxis Zahlen. Zum Beispiel N=33, E=3 und D=7. Dies sind nicht etwa beliebige Zahlen, sondern ganz bestimmte (warum man gerade diese Zahlen nimmt wird weiter unten unter „Bildung der Schlüssel N, E und D“ noch erläutert). Eine weitere Zahlenkombination wären z.B.: N=49048499, E=61 und D=2409781.Das sind die sogenannten magischen Zahlen.
Vom Klartext zur Klartextzahl: Die Umwandlung vom Klartext zur Klartextzahl ist recht simpel. Man schreibt einfach den Klartext auf und schreibt dann unter jeden Klartextbuchstaben dessen entsprechende Stelle im Alphabet.
Z.B.:

m	o	r	g	e	n	u	m	d	r	e	i
1	1	1	0	0	1	2	1	0	1	0	0
3	5	8	7	5	4	1	3	4	8	5	9

Verschlüsseln mit N und E: Man schreibe die Klartextzahl E-mal hin (z.B.: 3 mal) und setze Malzeichen dazwischen. Beim Ausmultiplizieren ziehe man schon bei den Zwischenergebnissen immer wieder N ab, sooft es nur geht, damit die Zahlen nicht zu groß werden. Am Ende bleibt eine Zahl übrig, die kleiner ist als N. Das ist die Geheimzahl,
Beispiel:
N = 33 ; E = 3; D = 7 ;
Klartextzahl = 5(=e)
5*5*5=125
125-33=92
92-33=59
59-33=26
also: Geheimtextzahl ist die 26 (=z)

Entschlüsseln mit N und D: Man schreibe die Geheimtextzahl D-mal hin und setze Malzeichen dazwischen. Beim Ausmultiplizieren ziehe man schon bei den Zwischenergebnissen immer wieder N ab, sooft es nur geht, damit die Zahlen nicht zu groß werden. Am Ende bleibt eine Zahl übrig, die kleiner ist als N. Das ist die Klartextzahl,
Beispiel:
N = 33 ; E = 3; D = 7;
Geheimtextzahl = 26 (=z)

26*26*26*26*26*26*26=8031810176

8031810176-33-33-33....-33=5

also: Klartextzahl ist die 5(=e)

Praktische Erklärung: Frau Schwarz besitzt die drei Schlüssel N, E und D. Von den Schlüsseln N und E hat sie allen ihren Freunden eine Kopie gegeben. Den Schlüssel D aber behält sie. Außer ihr hat ihn niemand. Wenn Herr Weiß einen geheimen Brief in der Truhe an Frau Schwarz schicken will, verschließt er das Schloss mit den Schlüsseln N und E. Danach kann er es selbst nicht mehr öffnen. Nur Frau Schwarz kann mit Schlüssel D an den Inhalt gelangen.

Bildung der Schlüssel N, E und D:

Man nehme:

Zwei Primzahlen p und q und setze

N:=p*q

Hilfszahl z:=(p-1)*(q-1)

Dann vergisst man p und q ganz schnell.

Man wähle für E eine Primzahl, die kleiner ist als z.

Man wähle D so, dass (E*D) mod z =1 ist.

Dann vergesse man z ganz schnell.

Hiermit versichere ich, dass ich die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Facharbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken entnommen wurden, mit genauer Quellenangabe kenntlich gemacht habe. Verwendete Informationen aus dem Internet sind der Lehrerin vollständig zur Verfügung gestellt worden.

Brakel, den 14.02.2001
Dennis Groppe

Hiermit erkläre ich, dass ich damit einverstanden bin, wenn die von mir verfasste Facharbeit der schulinternen Öffentlichkeit zugänglich gemacht wird.

Brakel, den 14.02.2001
Dennis Groppe