Was ist eine Chipkarte?

Alexander Pretschner, Perlen der Weisheit, 22.1.2002

Liebe junge Kolleginnen und Kollegen,

... [Smartkarten] werden uns Bürger auch in unserem Verhalten beeinflussen.

Ich sage bewußt nicht „uns Menschen“ ...

Deshalb, meine jungen Freunde/innen: ... Fragen Sie sich, wohin IHR Weg führen wird.

Lassen Sie uns den richtigen Weg wählen, die „Straße der Unabhängigkeit“

[Dethloff im Vorwort von Rankl/Effing]

Überblick

- Vortrag über Chipkarten im wesentlichen „Was ist ein Computer?“

- Anwendungen
- Physik
- Betriebssysteme
  - Dateisystem
  - Kryptographische Routinen; PKI
- WAP, CEPS
- Modellierung

- Kohäsionsförderung

Anwendungen

- Identifizierung/Authentisierung (PINs, biometrisch)
- Speicherkarten: Telefon, KV, (Patientendaten)
- GSM-SIM-Plugin-Karten in Handies, Iridium
- Bankkarten, Geldkarten
- Browser in Chipkarten, OTASS
- Mensa, Zugangskontrolle, Maut, Paß, CEPS, etc.
- PKI
- PalME

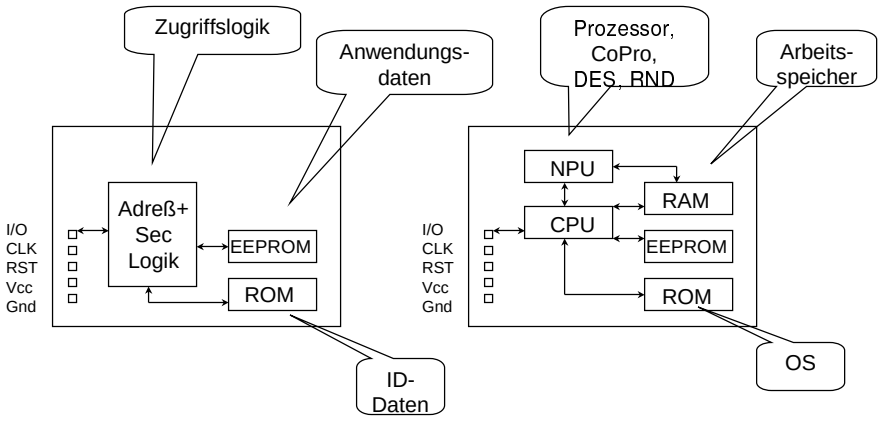
- 2000: 628 Mio Prozessorchipkarten, 1100 Mio Speicherchipkarten (64%)

- Gemplus, Schlumberger (Bull), Giesecke&Devrient

Geschichte

- 1968 Dethloff, Grötrupp: Chipkarte zur ID
- 1970 Moreno
- 1984 PTT: Telefonkarten
- 1984-94 Chips in EC-Karten
- 1985 Thomson Semiconducteurs: Prozessorkarten
- 1992 E-Purse: Danmönt
- 1994 EMV-Spezifikation
- seit 1994 Krankenkassenkarten

Architektur Speicher- und Prozessorkarte



plus
Antenne, (De-)Modulator, Taktgenerator, Kollisionskontrolle, Spannungsregelung
für kontaktlose Karten; ggf. auch Kombikarten

Physik

- Kartenkörper aus PVC, ABS, Polycarbonat (CDs), PET
- Verbindung mit Kontakten gelötet oder gedruckt (7816-2)
- Versorgungsspannung 5V, 3V
- Induktiv: 125kHz (100-1000 Wndg) oder 13, 6MHz (3-10)
- Externer und interner Takt (~5MHz, intern: EEPROM!)
- Prozessoren üblicherweise CISC à la 6805, 8051;
  8 bis 32 Bit
- RAM, EEPROM, ROM auf einem Chip (Platzfaktor 4)
  256 Bytes RAM, 4K-1MB (2003) ROM

- Kommunikation synchron/asynchron seriell; 9600 Baud;
  incl. Protokolloverhead 1.5 ms/Byte

Aufbau – Beispiel GSM OS+Appl.

ROM – GSM-Betriebssystemroutinen

I/O, E2PROM Schreibroutinen
- Krypto
- BIT
- Herstellerkommandos
- GSM-Kommandos (Terminal Response)
- SIM Appl. Toolkit Interface (DisplayText)

Integrierte Applikationen

- OTASS
- WIM

E2PROM

Systembereich

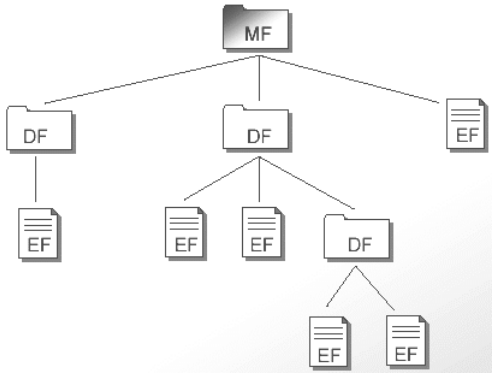
Dateisystem

Applikationen (Banking, Browser)

Betriebssysteme

- Aufgaben: Datenübertragung, Kommandoablaufsteuerung, Dateiverwaltung, Krypto
- Funktionalität für Dateisysteme:
  SelectFile, Read/Update Binary/Record, Verify
- Ggf. Applikationen:
  Authentisierung (Profil Q), Digitale Signatur, WAP (Push), ...
- CardOS, MFC, Multos, P/M/MP-COS, Starcos
- Kommandos von außen oder auf Karte
  (Maschinencode, VMs)
- ISO 7816-4
- ~ 16kB im ROM

Filesystem (7816-4)



- Master File, Dedicated Files, Elementary Files
- ggf. OO-Struktur mit Zugriffsberechtigung (PKCS#15)
- Dateien für Schlüssel (PINs, PUKs), Applikationen (single/multi), Applikationsdaten, Programmcode

Kryptographie

- Ziele:
  Vertraulichkeit, Authentizität, Verbindlichkeit
- symmetrisch (DES,IDEA)
- asymmetrisch (RSA,DSA,EC)
- Z.B. im WIM des WAP
  - Als Applikation auf der Karte
  - Kommandos: Dateiverwaltung, Security Environments, encipher/decipher, Zufallszahlen
- Im folgenden: DES, RSA z.B. für WAP-Bankapplikationen: Auth. per RSA, Komm. per DES

Asymmetrische Krypto: RSA

- Prinzip
  Verschlüsseln $enc(m)=m^e \% n$ mit öff. Schlüssel e
  Entschlüsseln $m=(enc(m))^d \% n$ mit pr. Schlüssel d
  mit $m = dec(d,enc(e,m)) = enc(e,dec(d,m))$
- Schlüsselgenerierung
  - Primzahlen p=3, q=11; öffentlicher Modulus n=p*q=33
  - Hilfsvariable z=(p-1)*(q-1)
  - *öff. Schlüssel* e mit e<z, ggT(z,e)=1; wähle einen: e=7
  - Wähle einen *pr. Schlüssel* d mit (d*e)%z=1: d=3
- Ver-/Entschlüsselung
  - Klartext x=4 mit x<n
  - Verschlüsselung: $enc(4)=4^7 \% 33=16$
  - Entschlüsselung: $dec(16)=16^3 \% 33=4$

Symmetrische Krypto: DES (FIPS PUB 46-2)

- RSA aufwendig, deshalb nur für Authentisierung, CCs
  - CoPro mit 1kB Assembler für 1024 Bit-Schlüssel: 200msec
- Nachrichtenverschlüsselung dann mit DEA/DES
  - Schlüsselerzeugung aus Zufallszahlen

- Idee: inverse Ver- und Entschlüsselungsfunktionen
  - z.B. selbstinverses $f=\lambda x.x+_{\%2}K$: $f(f(x)) = x$
- 64-Bit Blöcke B mit Schlüssel S kombinieren:
  $B' = enc(B) = B+_{\%2}S$; $dec(B') = B'+_{\%2}S = B$
- Varianten von ECB
  - Cipher Block Chaining mit Key Schedule
  - Triple-DES mit drei Schlüsseln

- 64 Byte verschlüsseln im μsec-Bereich

Digitale Signatur, Zertifikate, MAC

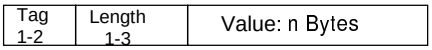
- $m=dec(K_{pr},enc(K_{pub},m))=enc(K_{pub},dec(K_{pr},m))$
- $dec(K_{pr},hash(m))$ ist digitale Signatur:
  $m \circ dec(K_{pr},hash(m))$ ist signierte Nachricht

- Zertifikat: CA-signierter öffentlicher Schlüssel
- Öff. Schlüssel S wird mit $dec(K_{pr}^{CA},hash(S))=\Sigma^{CA}$ signiert; Zertifikat $S \circ \Sigma^{CA}$
- Falls $enc(K_{pub}^{CA}, \Sigma^{CA})= * \circ hash(S)$, dann S authentisch, denn $enc(K_{pub}^{CA}, dec(K_{pr}^{CA},hash(S)))=hash(S)$

- Message Authentication Code (MAC) identisch, aber üblicherweise symmetrisch

Programmausführung auf Chipkarte

- ME sendet Kommando ASN.1



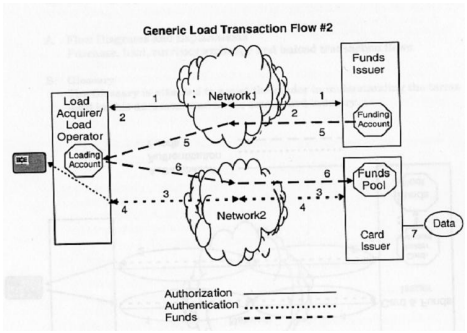
- Karte antwortet mit Statuswort und ggf. Daten
  - Karte passiv
  - „proaktive“ Applikationen z.B. GSM SIM Application Toolkit mit Polling
  - WIM-Kommandos: computeDigSig, performRSAHandshake
  - GSM-SIMAT: displayText, sendSMS, getInkey, setupMMI zusätzlich zu Kanal- und Dateiverwaltung
- Karten mit nachladbarem Code
  - Relozierbarer Maschinencode
  - VM, z.B. JavaCard

Anwendung: WAP (www.wapforum.org)

- Integration „Wireless Data“, Telephonie, Internet
- Plattformübergreifende Protokolle (Internet, Segurity, Proxies)
- WWW
  - Naming model (URLs, RFC 2396)
  - Content typing (RFC 2045&48)
  - Content formats (HTML, JavaScript)
  - Protokolle (HTTP, RFC2616; TCP/IP)
- WAP zusätzlich
  - Push
  - WTA (Telephoniesupport)
  - Proxies ermöglichen u.a. Kommunikation WAP-Protokollstack WSP-WTP-WTLS-WDP nach WWW (HTTP, TCP/IP)
  - WTLS-Segurity mit Chipkarten

CEPS

- Common Electronic Purse Specification – Geldkarten
  - Währungsunabhängig, sichere Transaktionen für Aufladen, Abheben, Anzeige, Cancellation, Sperren
- Dynamische wechselseitige Authentisierung
- $ 2 Mrd. für Einkäufe <10$



Modellierung

- Mikro- und Makrozustandsautomaten modellieren Applikation, z.B. WIM
- rein sequentielles System

- Abstraktionen für Orthogonalisierung der Funktionalität gefunden
- ich will Petrinetze für implizite Interleavings

- Treiberkomponente für Protokolle (DinSig)
  - Permutationen explizit

Chipkarten: Herausforderungen

- Authentifizierung etc. sicherheitskritische Applikationen

- Stark begrenzte Ressourcen
- nach Distribution nicht „reparabel“
- Kleinere Displays und Eingabeeinheiten
- Domänenspezifische Abstraktionen
- 64K-1MB Speicher – Qualitätssicherung?

Standards/Normen

- IEC/ISO 7816
  - (1) Physik, (2) Abmessungen und Lage der Kontakte, (3) Kommunikationsprotokolle, elektrische Eigenschaften der Kontakte, (4) Dateistruktur, Kommandos, (5) Registrierung

- PKCS#15 (RSA): Standardisierung für Applikationen (der Dateiformate, z.B. Zertifikate)

- WAP – wireless application protocol

- CEPS/PKI

Zusammenfassung

- Chipkartenanwendungen: Speichern und authentifizieren, aber auch: Browser
- Im wesentlichen: Dateisystem/-verwaltung
- Nachladbarer Programmcode z.B. mit JavaCard
- Sicherheitskritisch
  - Gatter randomisiert verschieben
  - Einfrieren von Prozessoren
- SIM: Authentisierung (WIM), OTA
- GSM-Gesprächsverschlüsselung nicht über SIM, sondern mit IMSI im ME

Literatur

- Rankl/Effing: Handbuch der Chipkarten
- Internet für Krypto, WAP, PKCS#15, CEPS