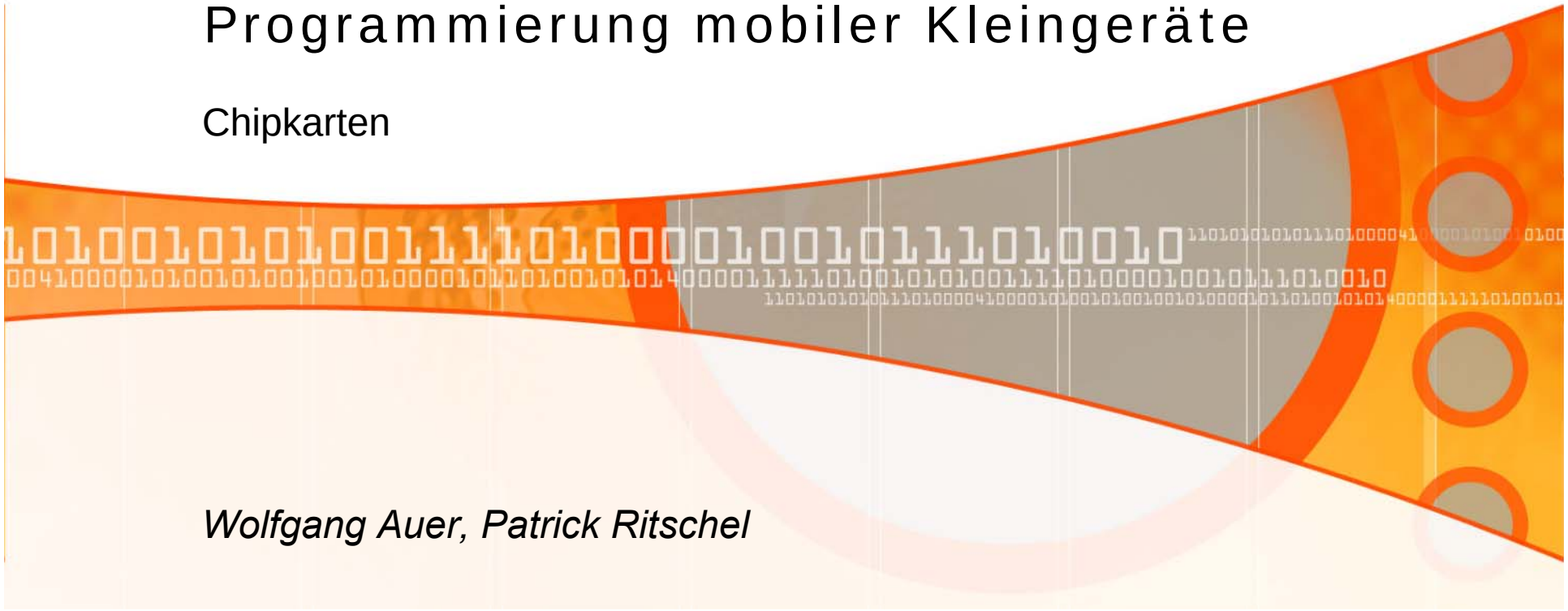
Programmierung mobiler Kleingeräte

Chipkarten



Wolfgang Auer, Patrick Ritschel

Geschichte der Chipkarten

- 1950: Erste Plastikkarte (PVC) von Diners Club
- 1968: Patent von Dethloff/Grötrupp in D
- 1970: Patent von Arimura in Japan
- 1974: Patent von Moreno in F
- 1985: Telefonkarten mit Chip in F und D
- 1984: Erste Bankkarten in F
- 1988: Prozessorkarte im C-Netz in D
- 1996: Weltweit erste flächendeckende Bankkarte mit el. Geldbörse *Quick* in Österreich
- 1999: Verabschiedung der EU-Direktive für el. Unterschriften
- 2005: Testbetrieb der e-Card (el. Krankenschein), Bankomatkarten beinhalten Signaturfunktion

Chipkarten
Normung

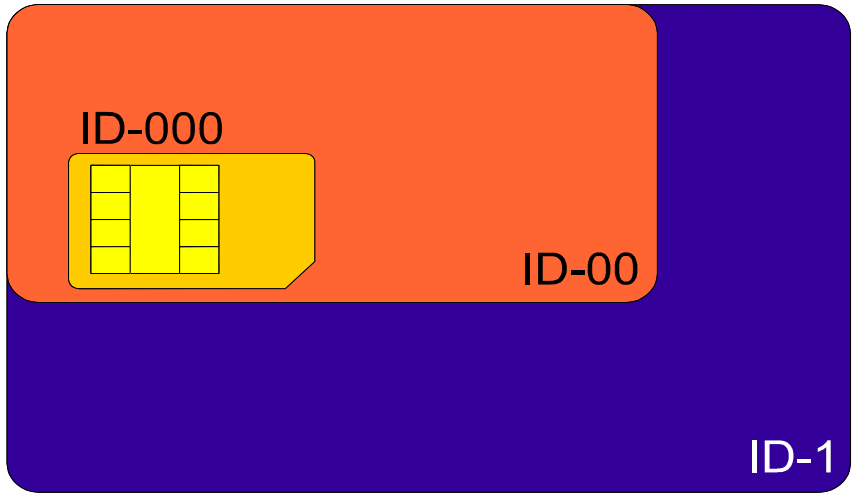
- Normen und Standards sind besonders wichtig
  - Gewährleistung von Kompatibilität
  - Trotzdem auch genügend Freiheit für konkrete Implementierungen einzelner Firmen
  - Oft schwierig, einen Kompromiss zu finden
  - Hängt nicht mit Patenten zusammen (IPRs, *Intellectual Property Right*, zu den Dokumenten können angefordert werden)
- Normierung
  - ISO/IEC: *International Organisation of Standardisation, International Electrotechnical Commission*
  - CEN: *Comitée Européen de Normalisation* für Europa
  - ÖNORM diskutiert und übernimmt die Standards
  - ETSI, 3GPP für Telekommunikationsanwendungen
  - EMV: *Europay, Mastercard, Visa* für Bankanwendungen

Anwendungsgebiete

- Chipkarten sind in einer großen Zahl von Anwendungen einsetzbar
  - Bankapplikationen
  - Telekommunikation
  - Krankenkassenkarten
  - Ausweiskarten
  - Besitzkarten (z.B. Kraftfahrzeuge)
  - Öffentlicher Personennahverkehr (ÖPNV)
  - Eintrittskarten, …
- Einsatz also hauptsächlich als *Secure Token* zur Identifikation

Chipkarten
Arten von Karten

- Abmessungen, Flexibilität und Temperaturbeständigkeit sind in ISO 7810 standardisiert
  - ID-1 (85,6 mm x 54 mm x 0,76 mm)
  - ID-00 (66 mm x 33 mm x 0,76 mm)
  - ID-000 (*Plug-In*, 25 mm x 15 mm x 0,76 mm)
  - Weiterer Formfaktor, der nur noch das Modul wiedergibt, ist in Arbeit
  - USB-Stecker



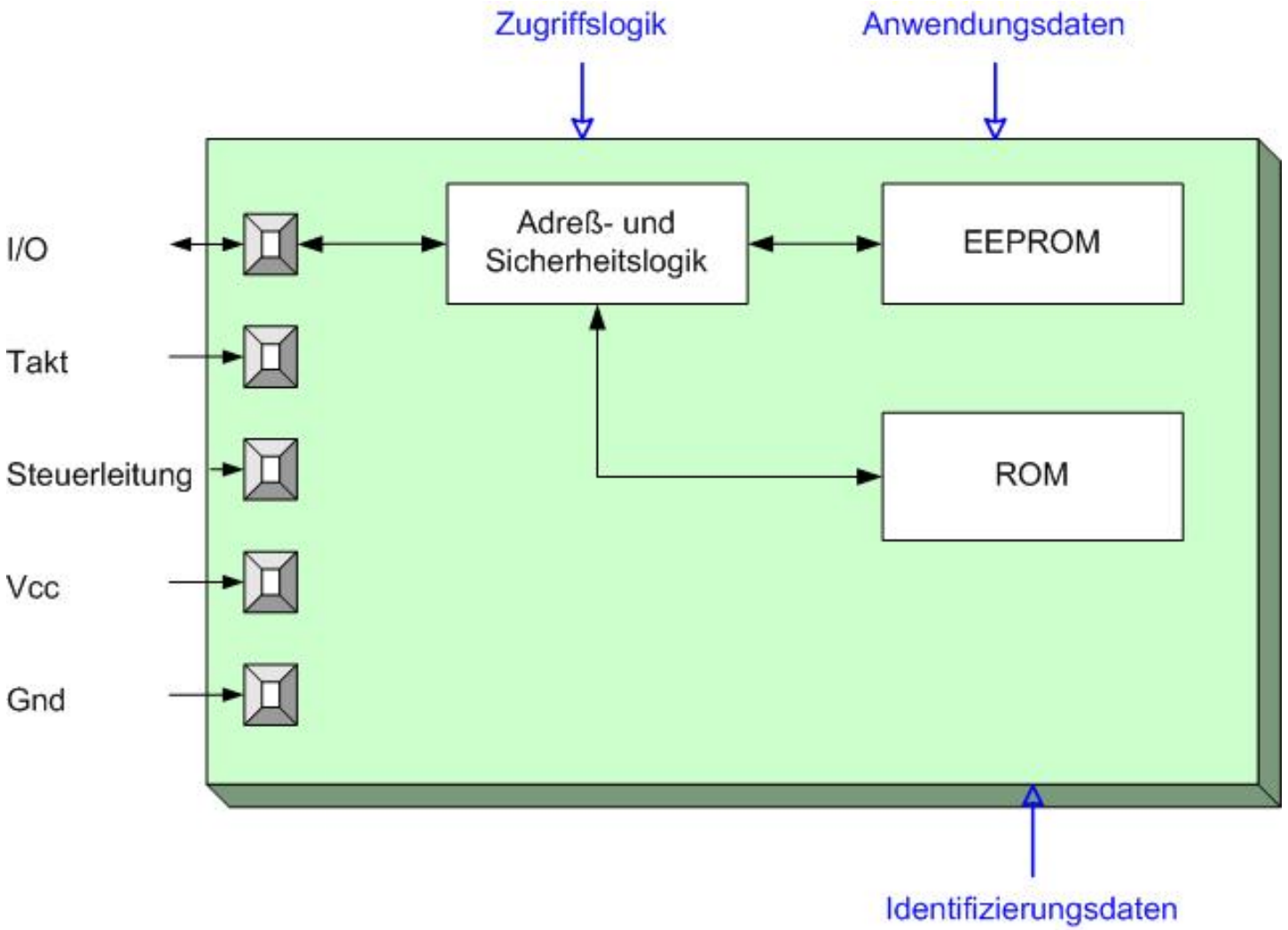
Chipkarten
Arten von Karten

- Hochgeprägte Karten
  - Einfachste Art der „maschinenlesbaren“ Form: *Ritsch-Ratsch-Maschine (Imprinter)*
- Magnetstreifenkarten
  - Ein Streifen hat bis zu drei Spuren
    - Spur 1: 79 Zeichen (6 Bit alphanumerisch), Schreiben nicht erlaubt
    - Spur 2: 40 Zeichen (4 Bit BCD), Schreiben nicht erlaubt
    - Spur 3: 107 Zeichen (4 Bit BCD), Schreiben erlaubt
- ISO 7811 „Identification Cards – Recording Technics“ definiert
  - Lage und Schriftarten der Aufdrucke und eingeprägten Zeichen
  - Lage und Datenspeicherung auf den Magnetstreifen

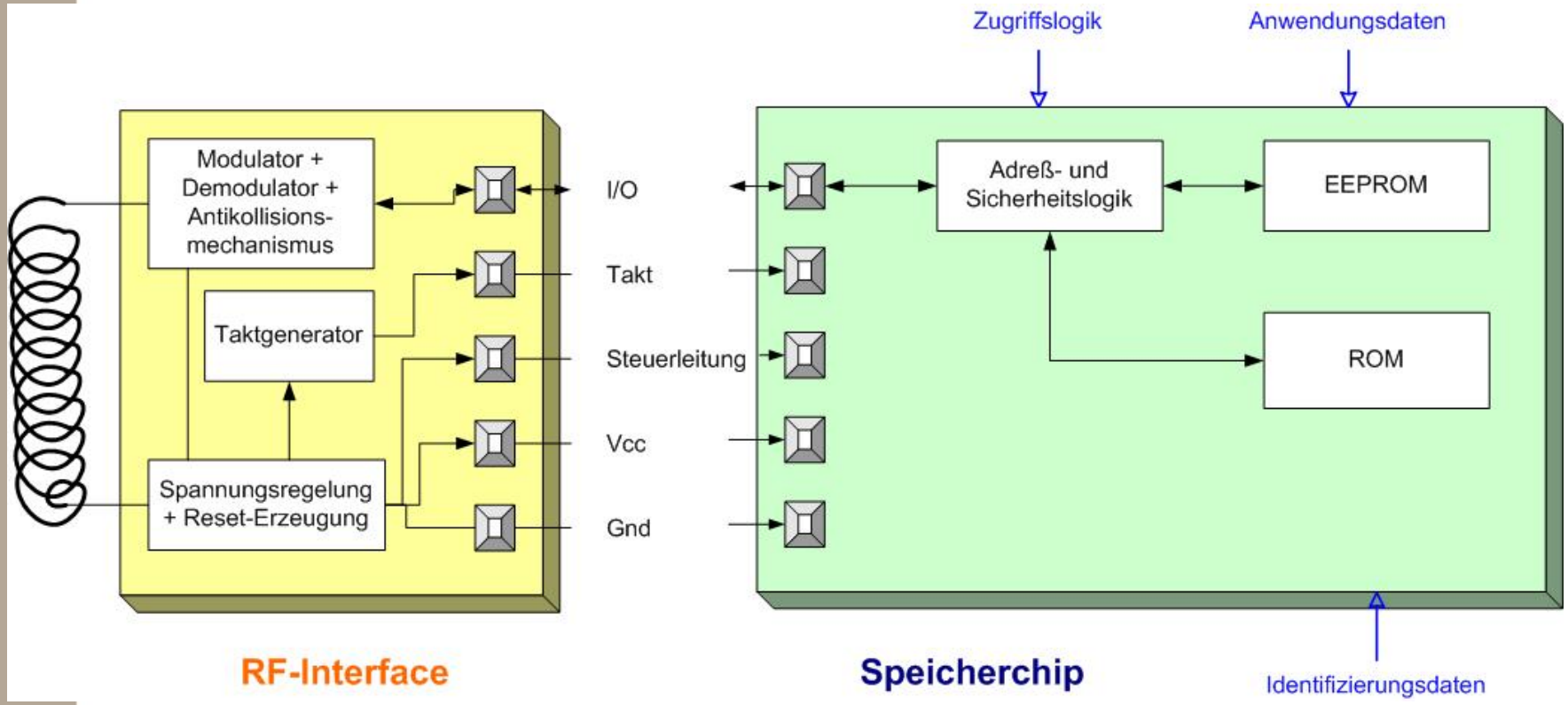
Chipkarten
Arten von Karten

Datenübertragung	Chiptyp				
		Speicherchip		Mikrocontrollerchip	
		Ohne	Mit	Ohne	Mit
		Sicherheitslogic		Coprozessor	
	Kontakt-behaftet				
	Kontaktlos				
	Dual-Interface				

Arten von Karten
Speicherkarten

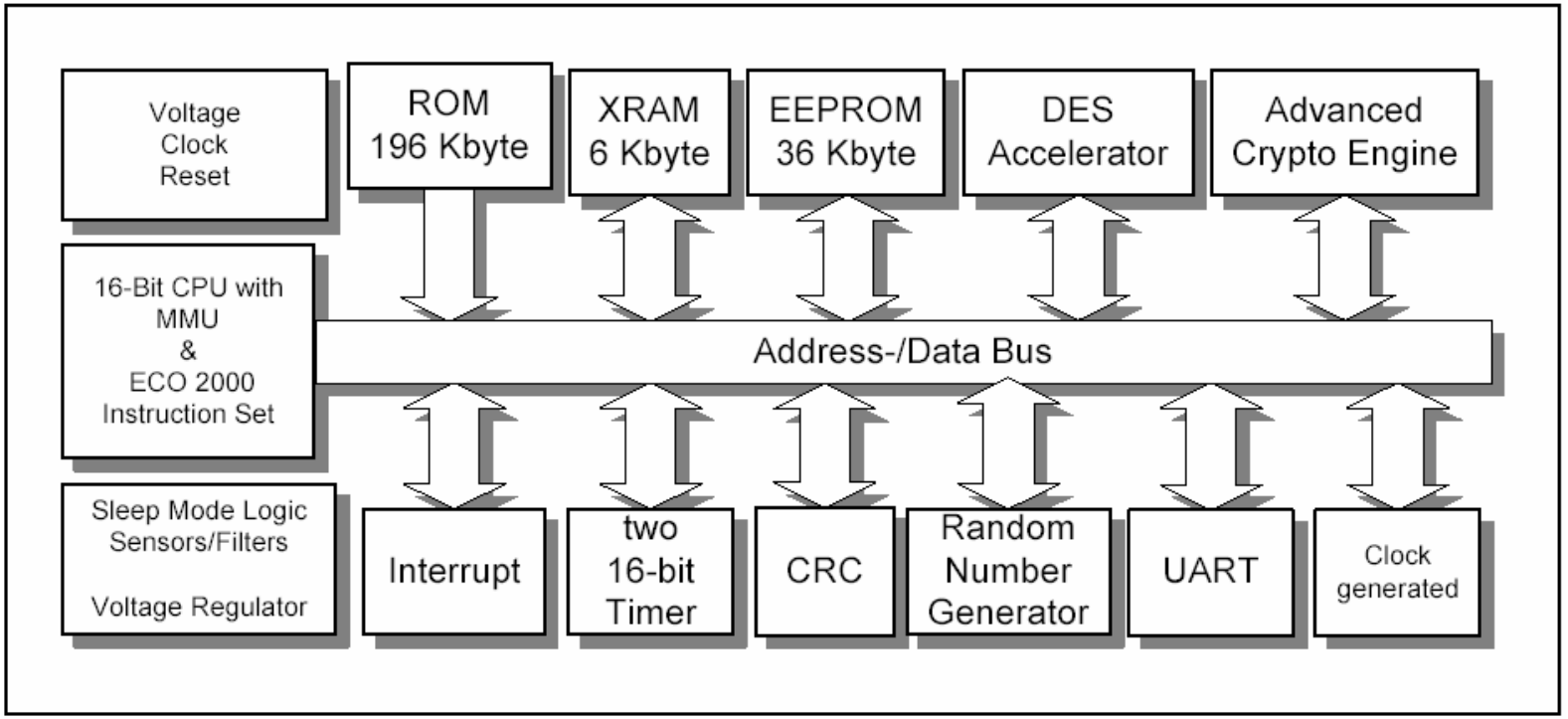


Arten von Karten
Kontaktlose Speicherkarten



Arten von Karten
Mikroprozessorkarten

- Karten mit µP werden auch *SmartCards* genannt
- Beispielsweise Infineon SLE66CX360PE
  - mit Koprozessor, speziell für Signaturkarten



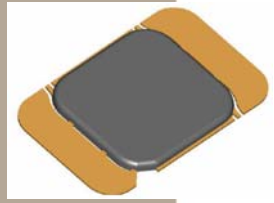
Arten von Karten
Mikroprozessorkarten

- Ein Mikroprozessorchip kann folgende Komponenten beinhalten
  - Mikroprozessor
    - 8 Bit: Derivative von Intel 8051, Motorola 6805
    - 32 Bit: ARM RISC, MIPS
  - Speicher
    - ≤ 8 KB RAM
    - > 350 KB ROM oder um 64 KB FLASH
    - ≤ 128 KB EEPROM mit 10 Jahren Datenbeständigkeit und ungefähr 100.000-500.000 möglichen Schreibzyklen

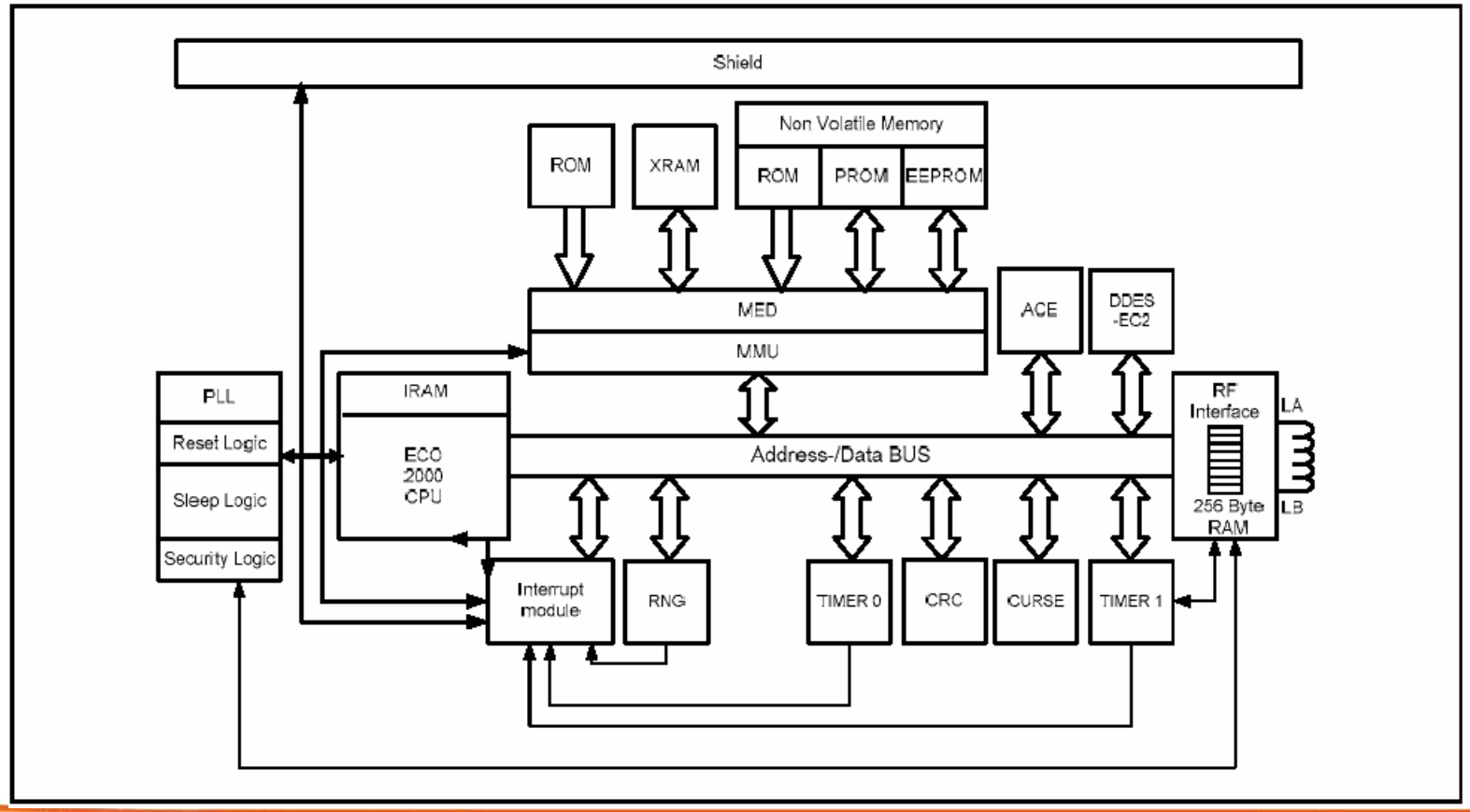
Arten von Karten
Mikroprozessorkarten

- Spezialisierte Module, wie
  - Kryptographische und arithmetische Koprozessoren, speziell für DES, RSA und ECC
  - Module zur Berechnung von Checksummen
  - Timer und Counter
  - True random number generators (TRNG)
  - Beschleuniger für interpretierte Sprachen (z.B. JavaCard)
  - Sensoren für Licht, Temperatur, Spannung, …

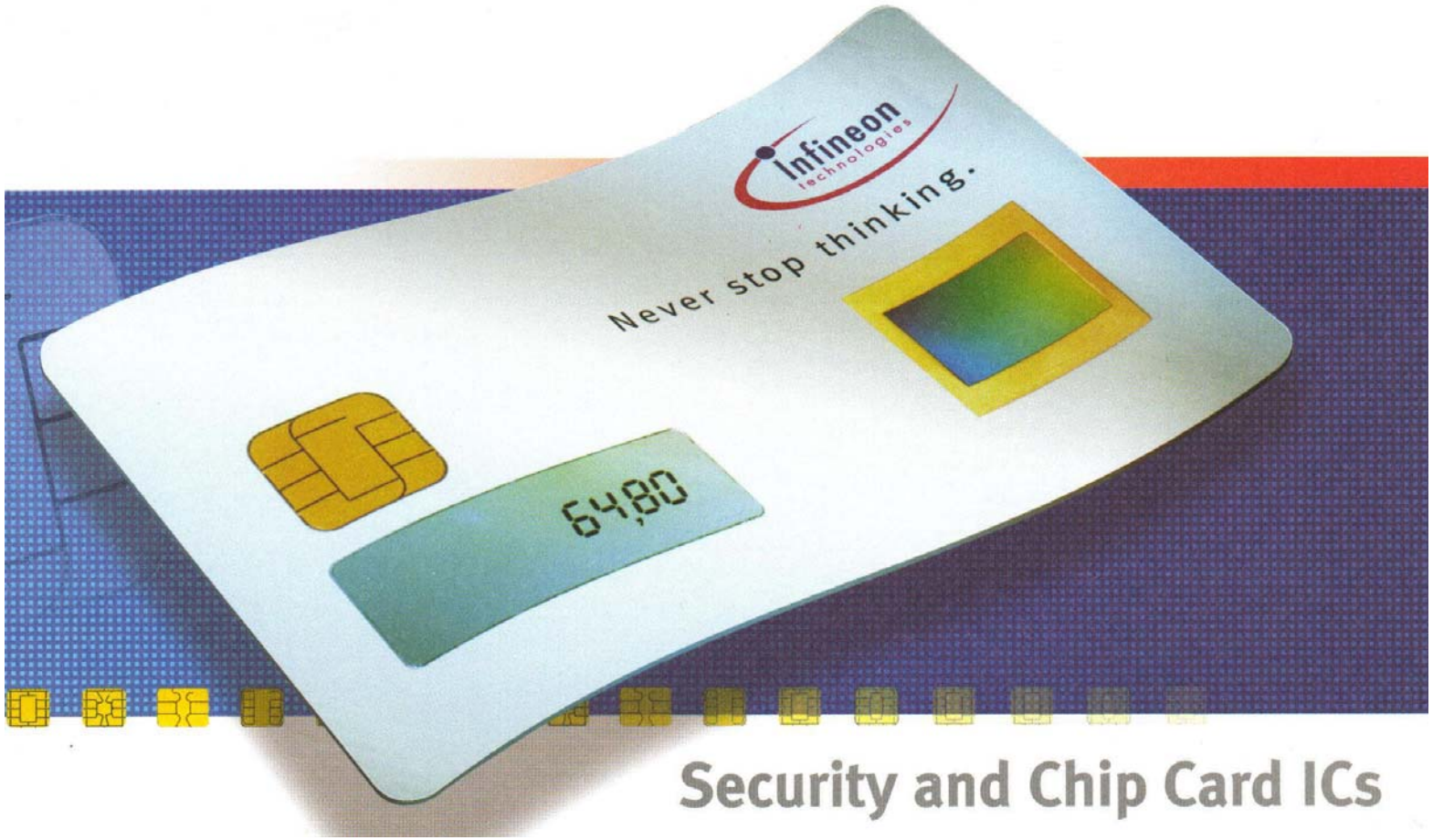
Arten von Karten
Kontaktlose Prozessorkarten



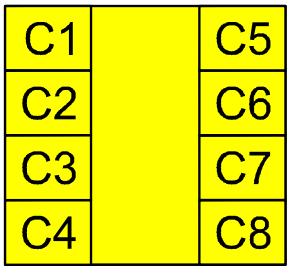
- Beispielsweise Infineon SLE66CLX321P
  - mit Koprozessor, speziell für Signaturkarten



Kartenelemente
Ein Sommernachtstraum



Chipmodule
elektrische Beschaltung



Kontakt	Bezeichnung	Funktion
C1	Vcc	Versorgungsspannung
C2	RST	Reset
C3	CLK	Takt
C4	RFU	
C5	GND	Masse
C6	Vpp	Programmierspannung
C7	I/O	Serielle Schnittstelle
C8	RFU	

Kommunikation

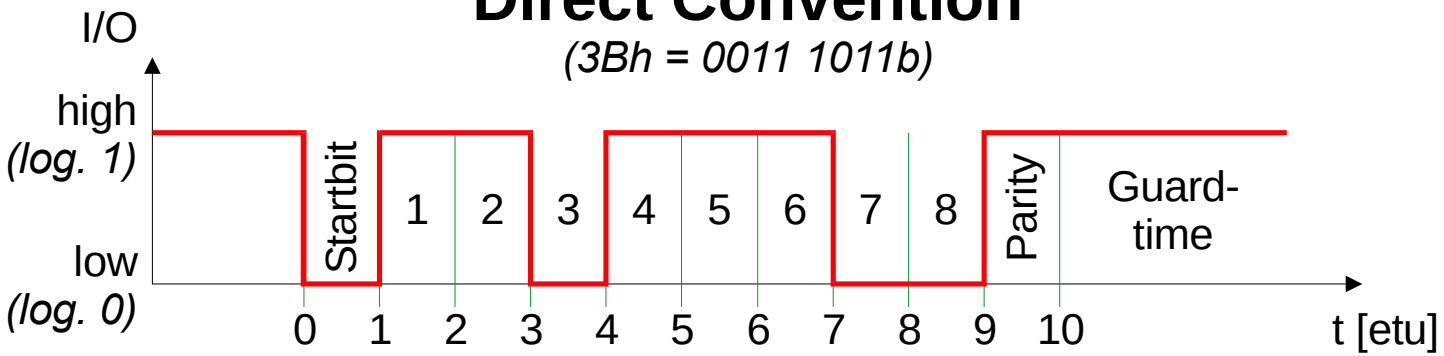
- Standard Seriell
  - Halb-Duplex-Kommunikation über eine I/O-Leitung
  - Protokolle T=0, T=1
  - Protokoll T=2 zur Vollduplexkommunikation mit einer zweiten I/O-Leitung ist in Arbeit
  - Bei 5 MHz ungefähr 150 kBit/s
- USB
  - Version 1.1, 1.5 MBit/s – 12 MBit/s
  - Version 2.0, 480 MBit/s; in Planung, doch derzeit ist es für Chipkarten nicht möglich, die Daten schnell genug bereitzustellen

Kommunikation
Protokollschichten

OSI-Schicht 7 *Anwendungsschicht*	ISO/IEC 7816-4 (APDU) ISO/IEC 7816-7 (SCQL) prEN 1546-3 (elektr. Geldbörse) GSM 11.11 (SIM)
OSI-Schicht 2 *Leitungsschicht*	ISO/IEC 7816-3 (T=0) ISO/IEC 7816-3 (T=1) ISO/IEC 10 536-4 (T=2)
OSI-Schicht 1 *Physikalische Schicht*	ISO/IEC 7816-3 (kontaktbeh. Karten) ISO/IEC 10536-3 (kontaktlose Karten)

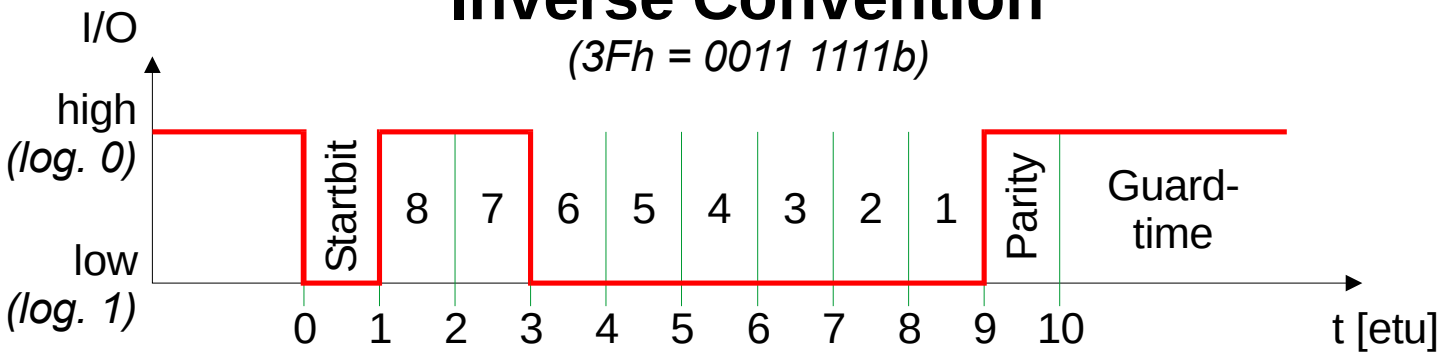
Direct Convention

(3Bh = 0011 1011b)



Inverse Convention

(3Fh = 0011 1111b)

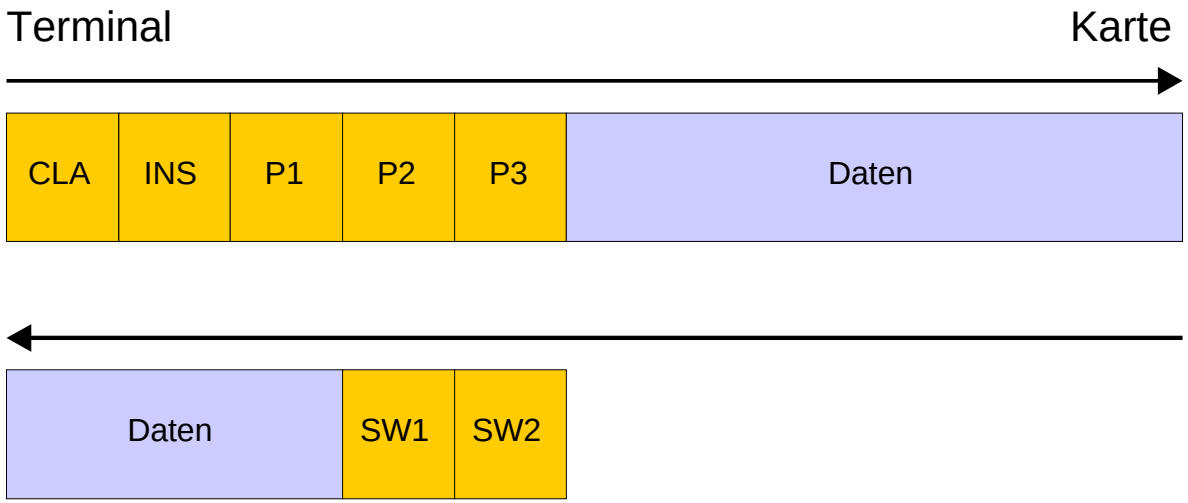


Kommunikation
ATR

- Der ATR, *Answer To Reset*
  - wird sofort nach dem Anlegen von Spannung gesendet (zwischen 400 und 40.000 Taktzyklen)
  - gibt Auskunft über die Chipkarte
    - genormte Kommunikationseigenschaften
    - zusätzliche Informationen des Betriebssystemherstellers
- Gesendet wird
  - Kommunikationsart: Direct / Inverse Convention
  - Kommunikationsgeschwindigkeit: Teiler *F*;*D*
  - Programmierspannung
  - Schutzzeit (*Guard Time* zwischen den Bytes)
  - Wartezeiten (Zwischen Blöcken und Befehlen)
  - I/O Pufferlänge
  - Zusätzliche Zeichen: *Historicals* (Bei *Quick*-Karten z.B. der Bargeldbetrag in der Börse)

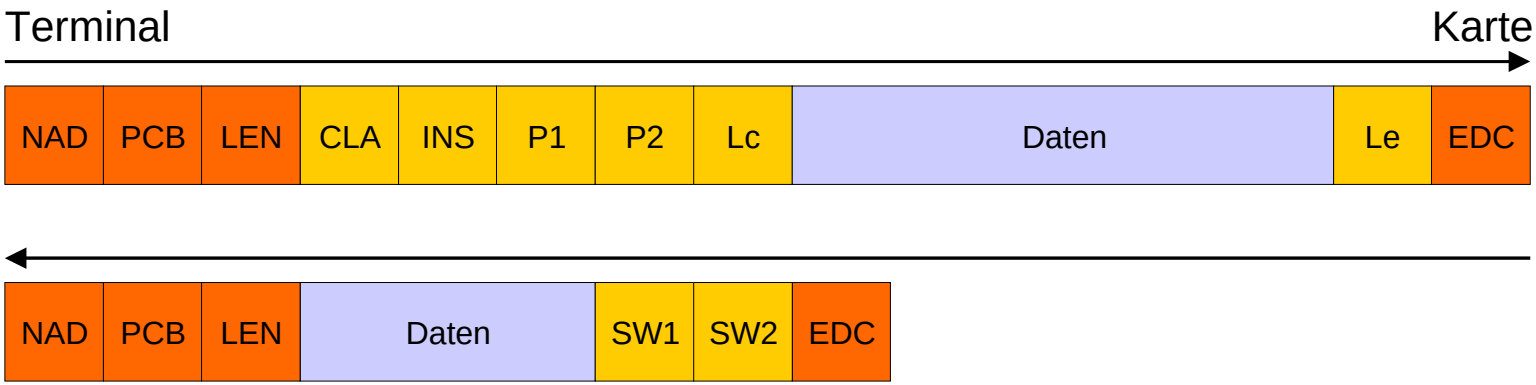
Kommunikation
Übertragungsprotokoll T=0

- Stammt aus Frankreich
- Maximale Einfachheit
- Verwendung z.B. bei GSM-Karten (SIM)



Kommunikation
Chipkartenprotokoll T=1

- Stammt aus Deutschland
- Blockprotokoll
- Strenge Schichtentrennung
- Verwendung z.B. bei Bankkarten

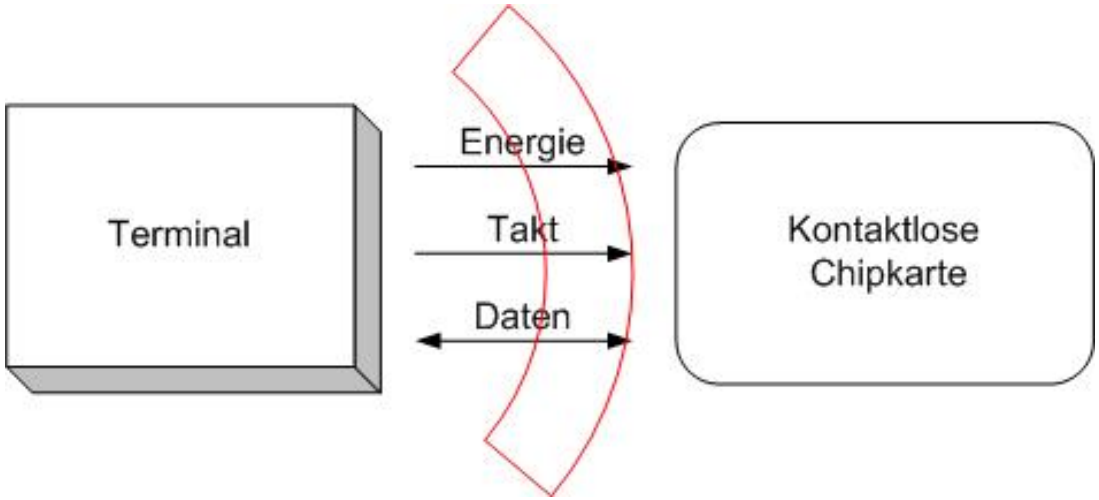


Kommunikation
Chipkartenprotokolle

Kriterium	**T=0**	**T=1**	**T=2**
Ursprung	Frankreich	Deutschland	Deutschland
Datenübertragung	Asynchron Halbduplex Byteorientiert	Asynchron Halbduplex Blockorientiert	Asynchron Vollduplex Blockorientiert
Norm	ISO/IEC 7816-3 GSM 11.11 EMV	ISO/IEC 7816-3 EMV	ISO/IEC 10 536-4
Fehlererkennung	Paritätsbits	Paritätsbits, EDC am Blockende	Paritätsbits, EDC am Blockende

Kommunikation
Kontaktlose Karten

- Seit vielen Jahren in der *RFID (Radio Frequency Identification)* eingesetzte Technik
  - In passiven Transpondern (Tierimplantate, Wegfahrsperren)
- Normiert sind die kapazitive und induktive Kopplung
  - andere Verfahren wie Funk- und Mikrowellen bzw. optische Übertragung werden wegen des Energieaufwandes nicht verwendet



Kommunikation
Induktive Kopplung

- Funktioniert wie ein Transformator mit sehr loser Kopplung
- Stromstärke ist stark entfernungsabhängig
- bei 125kHz werden ca. 100-1000 Spulenwindungen, bei 13,56MHz 3 bis 10 Windungen in der Karte (Inlay) benötigt
- Datenübertragung
  - zur Karte über amplitudenmoduliertes Feld
  - von der Karte durch Laständerung, Detektion durch Innenwiderstandsänderung im Terminal
    - Da die Änderung nur sehr klein ist, wird das Signal in zwei Hilfsträgerfrequenzen zurückmoduliert → ein breiteres Frequenzspektrum ist notwendig
- Antikollision muss implementiert werden

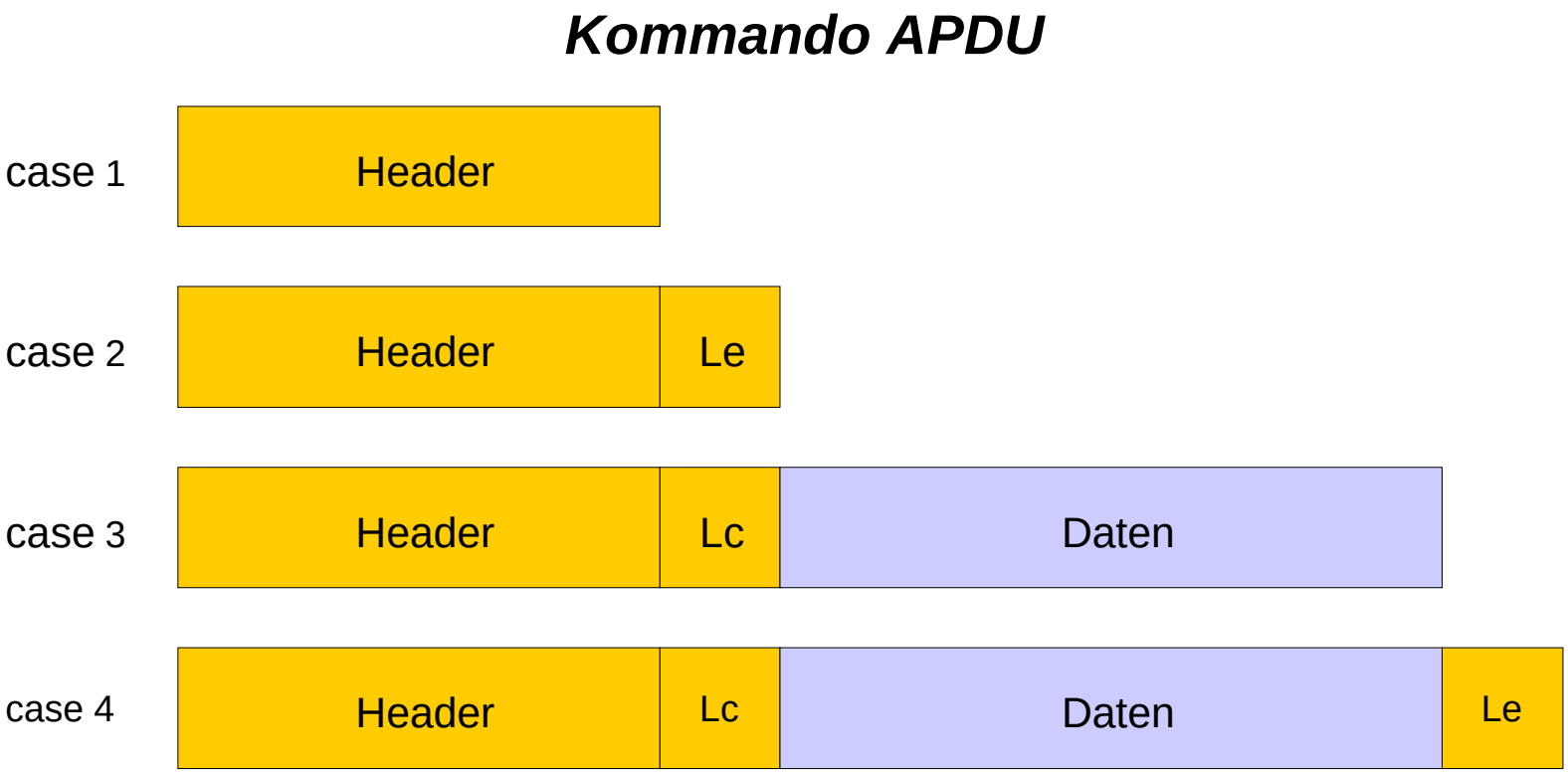
Kommunikation
Kontaktlose Karten

- Verschiedene Bereiche für die Kommunikation kontaktloser Karten sind definiert
  - Close Coupling Cards, ISO/IEC 10536
    - Auflegen oder Einführen der Karte auf/in ein Terminal
    - Trägerfrequenz 3 – 5 MHz
    - Leistungsaufnahme der Karte < 150 mW
  - Proximity IC Cards, ISO/IEC 14443 A/B
    - Reichweite ca. 10 cm
    - 13,56 MHz
  - Vicinity IC Cards, ISO/IEC 15693
    - Reichweite bis zu 1 m
    - Hat sich noch kaum durchgesetzt; Anwendungen??

Kommunikation
APDU, Definition

- Kommandos werden immer an die Chipkarte gesendet, diese reagiert nur
- In GSM ist es auch möglich, daß die Chipkarte in einem komplizierten Protokoll Kommandos an das Terminal sendet (*proaktive Kommandos*)
- Einzelne Kommandos werden über APDUs gesendet
  - *Application Protocol Data Unit*
- Antworten werden mit Response APDUs von der Karte zum Terminal gesendet

Kommunikation
APDU, Aufbau



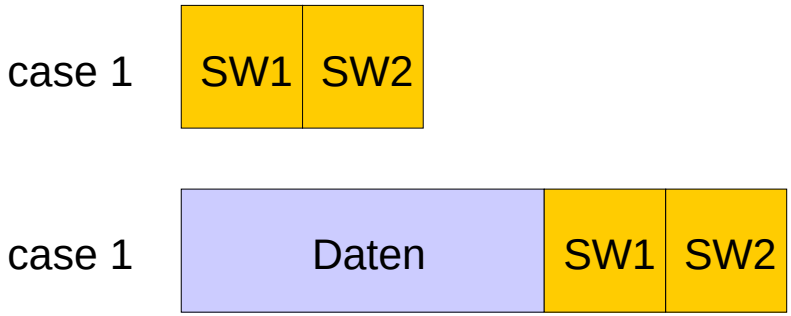
- Der Header setzt sich zusammen aus
  - *CLA:* Byte, das die Anwendungsklasse des Kommandos darstellt
    - *0x00* für ISO-kompatibles Normkommando
    - *0x80* für Applikationskommando wie GSM
  - *INS:* Byte, das das tatsächliche Kommando angibt
    - *0xA4*: SELECT FILE
  - *P1, P2:* Zwei Bytes, die spezielle Parameter zu den Kommandos darstellen
    - *File ID* des zu selektierenden Files
- *Le:* Anzahl der zurückzugebenden Zeichen
- *Lc:* Anzahl der mit dem Kommando gesendeten Zeichen

Kommunikation
APDU, Aufbau

•Das *Status Word* gibt zurück, ob ein Kommando funktioniert hat oder nicht

- *0x9000:* OK
- *0x9FXX:* OK, XX Daten sind noch abzuholen
- andere Werte kodieren meist Fehler

Antwort APDU



Betriebssysteme
Allgemeine Programmierbarkeit

- Java Card
  - Untermenge von Java, noch eingeschränkter als J2ME
    - Kein Löschen von angelegten Objekten
    - Kein Garbage Collector
    - Keine Fließkommazahlen
    - int ist 16 Bit lang
  - Sehr viele Anwendungen im Feld
  - Teurere Chips, da die Java VM höhere Anforderung an die Hardware stellt
- MULTOS
  - Sehr sicher für Bankapplikationen
  - Applikationen müssen kompliziert zertifiziert werden

Betriebssysteme
Allgemeine Programmierbarkeit

- Basic Card, Firma Zeitcontrol
  - Sehr langsame Karten
  - Meist für Zutrittskontrolle eingesetzt
- Windows for Smartcards
  - Microsofts Betriebssystem für Smart Cards
  - Mittlerweile nicht mehr unterstützt und weiterentwickelt (seit 2003)
- Linux
  - Noch nicht verfügbar implementiert
  - Derzeit zu aufwendig für gängige Controller

Kommandos von Chipkarten

- `SELECT FILE`
  - selektiert eine Datei
- `READ BINARY, UPDATE BINARY`
  - liest bzw. schreibt eine unstrukturierte Datei
- `READ RECORD, UPDATE RECORD`
  - liest bzw. schreibt eine Record-Datei
- `VERIFY, UNBLOCK CHV, UPDATE CHV`
  - Prüft, entsperrt oder ändert einen *Card Holder Value* (PIN, Personal Identification Number)
- `INTERNAL AUTHENTICATE, EXTERNAL AUTHENTICATE`
  - Führt eine Authentifizierung (extern oder intern) durch
- `GET CHALLENGE`
  - Holt eine Zufallszahl für eine Authentifizierung ab

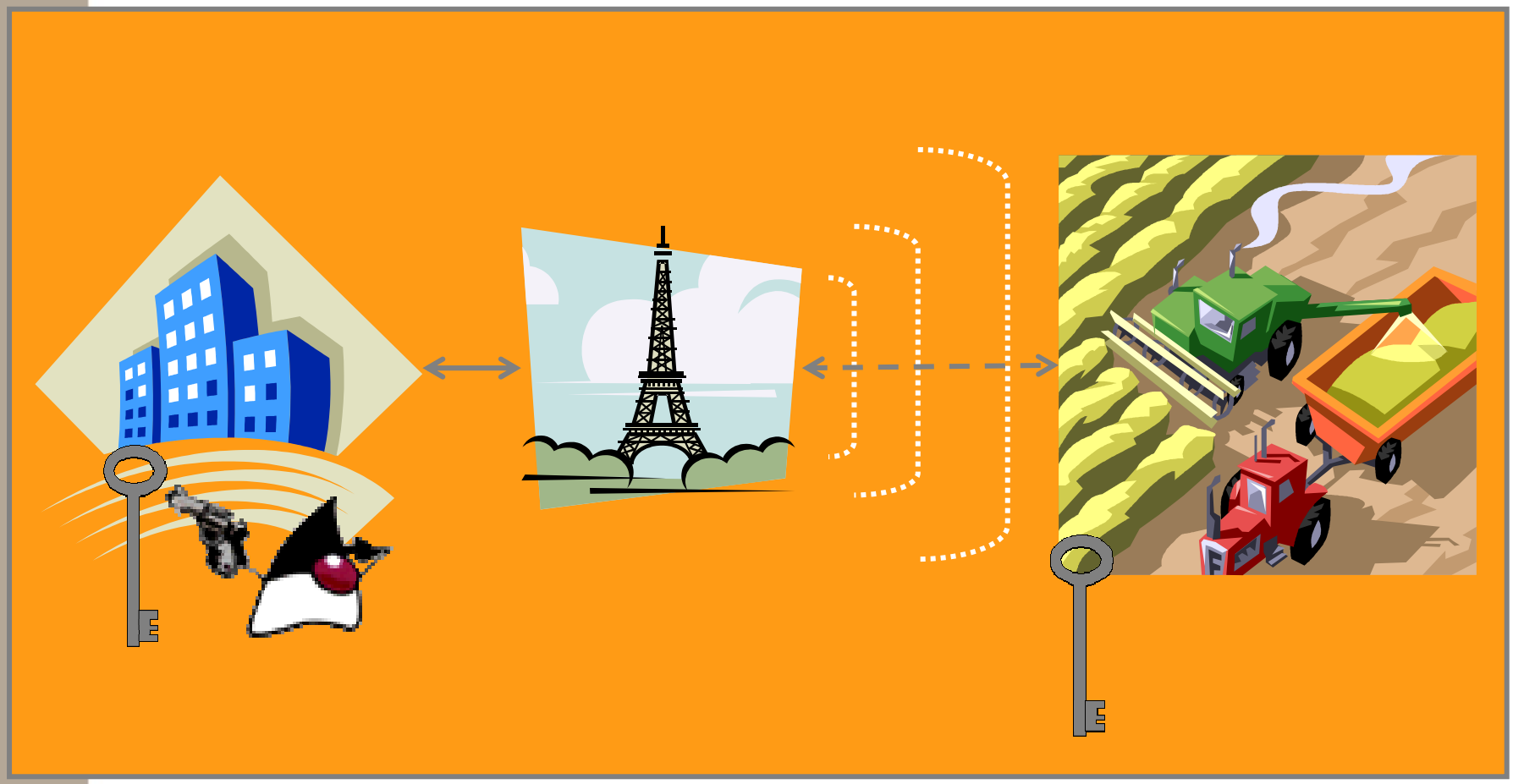
Smart Cards
Abschließendes

- Die Entwicklung von Chipkartenbetriebssystemen und –software ist trotz der Größe (~100KB) sehr aufwendig
  - Eine Testproduktion läuft mit ca. 10.000-20.000 Chips (1-2 Wafer)
  - Eine Produktion läuft ab 100.000, oft Millionen Chips
  - Ein Fehler ist meist theoretisch behebbar (Patches), doch wenn die Karten im Feld sind nicht mehr praktisch durchführbar
- Ein Simulator für ein Chipkartenbetriebssystem ist unter GPL verfügbar:

  http://www.geocities.com/SiliconValley/Foothills/4710/tscs.html

Angriffe

Wo wird angegriffen?



Angriffe
Social Engineering

- Der Anwender (meist ein Mensch) ist direkt angreifbar
- Deshalb werden menschliche Schwächen für Angriffe ausgenutzt
  - Faulheit: Ok Anklicken ohne die Dialoge genau zu lesen
  - Neugier: Bewußtes Eingehen kleiner Risiken
  - Andere soziale Eigenschaften
    - Rachsucht
    - Habgier
    - Freundschaftsdienste
    - Triebe

Angriffe auf die Hardware

- Die Hardware von Systemen kann auf verschiedene Arten untersucht werden
  - Der Speicher eines Computers kann ausgelesen werden
  - Die Hardwarebausteine können mikroskopisch untersucht werden
  - Aus den gewonnenen Untersuchungsergebnissen (Codestücke, Schlüssel) können weitere Angriffe abgeleitet werden
  - Diese Attacken werden Side-Channel-Attacken genannt
- Die Hardware kann aber auch manipuliert werden durch
  - Strom- oder Spannungsimpulse
  - Ändern der anliegenden Taktfrequenz
  - Freilegen des Siliziums und mit einer Lampe/einem Laser anblitzen
  - Erhitzen/Abkühlen des Chips
  - …

Angriffe auf die Hardware
Kompromittierende Abstrahlung

- Ermöglicht das Auslesen von Daten ohne direkten Kontakt mit dem Gerät
  - Über Induktion können Systeme auch gestört werden
  - Elektromagnetische und akustische Abstrahlung
- Am besten eignet sich der Bildschirm
  - bei günstigsten Eigenschaften > 100m, auch vom mobilen Fahrzeug aus
  - bis 20 Bildschirme sind auseinanderhaltbar
  - Verschlüsselung nützt hier nichts
  - Gesetzlich relativ unbedenklich
- Technische Signaldaten
  - Frequenz idealerweise 150MHz – 200MHz
  - Datenrate > 100 kBit/s
  - Kabel sind nur in unmittelbarer Nähe messbar
  - Lichtleiter sind hier eine sehr gute Alternative

Angriffe
auf mobile Geräte

- Mobile Geräte sind speziell gefährdet
  - Mobile Geräte können gestohlen und untersucht werden
- Verbesserung durch Nutzung von Smartcards (*Hardware Security Module*) in den Endgeräten
- Schlüsselsuche nicht nur logisch möglich, sondern auch mit
  - Hardware Attacken (statisch)
  - Elektrischen Attacken (dynamisch)
  - Analytischen Attacken

Angriffe
Hardware Attacken

- Verschiedene Methoden
  - Optische Analyse
  - Speicher direkt auslesen/ verändern
  - Reverse Engineering
- Gegenmaßnahmen von HW-Hersteller
  - Verschlüsselte Ablage von Programm und Daten
  - Verschleiern des Layouts
  - Schutzschichten auf der Oberfläche

Hardware
Elektrische Attacken

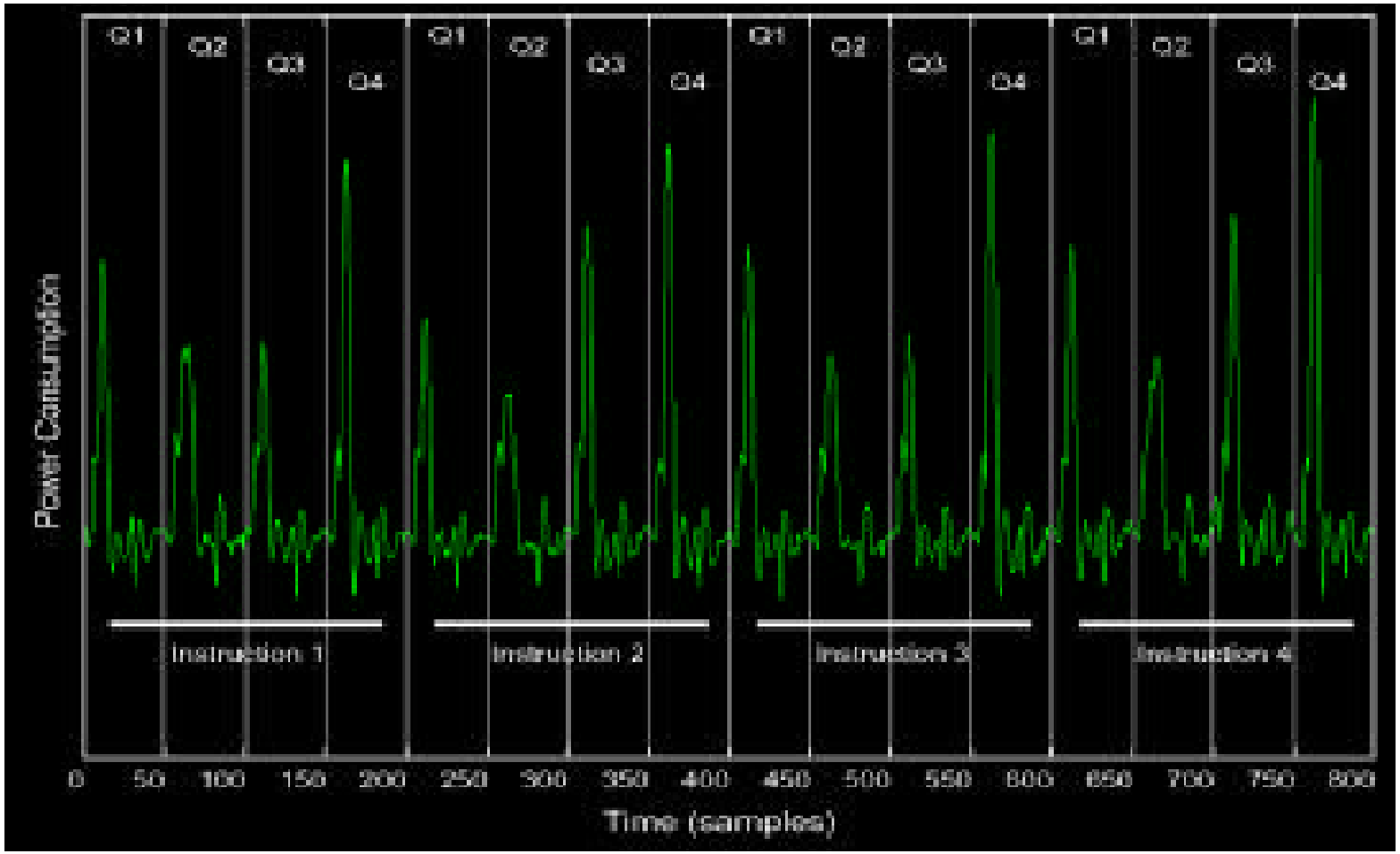
- Probing
  - Busleitungen während des Betriebs abgreifen und Daten visualisieren
  - Abwehr: scrambled bus, Schutzschichten
- Physical Stressing
  - Änderung des Programmablaufs durch Störung (Licht, Stromimpulse, …)
  - Abwehr: Sensoren verschiedenster Art

Attacken
Analytische Attacken

- Messung der Stromaufnahme
  - *SPA, Simple Power Analysis*: Identifizierung einzelner Instruktionen und Operanden anhand eines Profils
  - Abwehr: Rauschen oder Glättung
  - *DPA, Differential Power Analysis*: Statistische Analyse einiger 1000 Messungen; Eliminierung des Rauschens durch Mittelwertbildung
  - Abwehr: Zufällige Zeitverschiebung

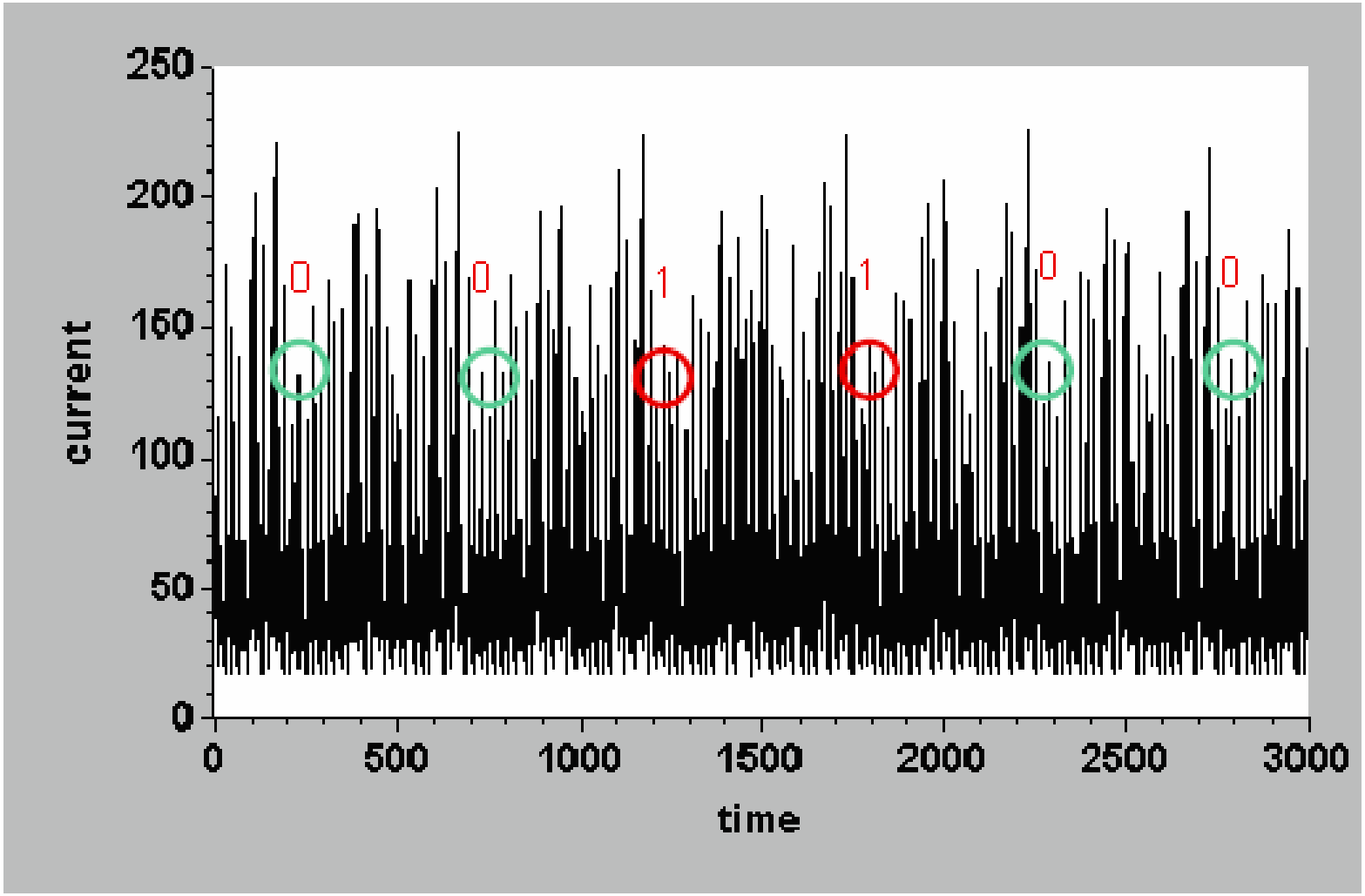
Angriffe: SPA

Grafik eines PIC-Prozessors



Angriffe: DPA

Auslesen eines DES-Schlüssels



Attacken
Analytische Attacken

- Messung des Timings
  - Verarbeitungszeit oft abhängig von Schlüssel und Eingabedaten (nicht *rauschfrei*)
  - Einfaches Beispiel:
    - Zeichenvergleich: Abbruch der Schleife bei erster Unstimmigkeit
  - Abwehr: Gleiche Zeit für alle möglichen Verarbeitungen des Algorithmus
  - Die Gegenmaßnahmen im Code sind nicht trivial
    - Ein `if (i != 0)` in Assembler übersetzt benötigt normalerweise nicht für beide genommenen Zweige gleich lange (`JZ` und `JMP` sind nicht dieselben Befehle, haben also **nicht** unbedingt dieselbe Laufzeit)

```
MOV ACC, @i
JZ not_null
; hier Code für i == 0
JMP naechster_schritt
not_null: ; hier Code für i != 0
naechster_schritt:
```

Attacken
Analytische Attacken

- *DFA, Differential Fault Analysis* bzw. *Bellcore*
- Durch Einstreuung eines Fehlers in Verschlüsselung kann Schlüssel berechnet werden
  - RSA: viele Messungen; RSA-CRT: eine
  - DSA: zwei Messungen; EC DSA: zwei
  - DES: 200 Messungen
- Abwehr: zweifache Ausführung oder Multiplikation mit einer Zufallszahl (Basisblinding bei RSA)

Angriffe: DFA
RSA mit CRT

- *CRT: Chinese Remainder Theorem*
  - beschleunigt die RSA-Berechnung durch Aufteilung in zwei Summenteile
- *RSA:*
  - *N* Produkt von 2 zufälligen Primzahlen *p* und *q*
  - *d* Geheimer Schlüssel, *e* Öffentlicher Schlüssel
  - Verschlüsselung (Verifikation): $s := m^e \bmod N$
  - Entschlüsselung (Signatur): $m := s^d \bmod N$
- Signatur mit RSA-CRT:
  - $S_1 := m^d \bmod p$
  - $S_2 := m^d \bmod q$
  - Signatur: $S := a*S_1 + b*S_2$ für vordefinierte Konstanten *a, b* aus $\mathbf{Z}_N$

Angriffe: DFA
RSA mit CRT

- Wir kennen zwei Signaturen derselben Nachricht *m*
  - *S*, korrekte Signatur
  - *S‘*, fehlerhafte Signatur
- Annahme, daß während Berechnung der Signatur *S‘* mit CRT
  - $S_1' := m^d \bmod p$ falsch (i.e. $S_1 \neq S_1' \bmod p$)
  - $S_2' := m^d \bmod q$ richtig (i.e. $S_2 = S_2' \bmod q$)
- Deshalb
  - $S \neq S' \bmod p$
  - $S = S' \bmod q$, i.e. $S - S' = 0 \bmod q$
- Also
  - ***ggT(S – S‘, N) =: q***
  - *N / q =: p*
  - *d* ableitbar aus *N, p, q, e.*

Softwareangriffe
Pufferüberläufe

- Beim Aufruf einer Funktion werden auf den Stack verschiedene Werte geschrieben (abhängig von System und Compiler sowie dess Optimierung)
  - Rücksprungadresse
  - lokale Variablen und Parameter
  - Adresse des Stackframes für die Funktion
- Geschicktes Überschreiben von Puffern kann das System manipulieren
  - meist im Stack
  - Heap ist als Angriffspunkt auch möglich
- Teilweise Abhilfe schaffen Canary Values (Löckvögel)
  - Schreiben eines Zufallswertes, der bei der Rückkehr aus der Funktion geprüft wird; hiermit kann festgestellt werden, ob der Wert überschrieben wurde

Softwareangriffe
Pufferüberläufe

- Der Angriff funktioniert meist über unsichere Funktionen, die aufgerufen werden
  - `char *strcpy(char *dest, const char *src)`
  - `char *strcat(char *dest, const char *src)`
  - `char *gets(char *buffer)`
  - und viele andere
- Ist die Programmiersprache C der Böse?
  - Einerseits ja: In C/C++ kann beliebiger hardwarenaher Code implementiert werden
  - Andererseits nein: Das Betriebssystem müsste sich vor Angriffen schützen

Softwareangriffe
Format-String-Schwachstellen

- Sehr ähnlich zu Pufferüberläufen, aber
  - es handelt sich mehr um Eingabefunktionen
    - Diese werden dann z.B. als Ausgabefunktion mißbraucht
  - der Angreifer weiß, wo die eingeschleusten Codes/Daten im Speicher stehen werden
- Die printf()-Funktionen sind
  - `printf(), fprintf(), sprintf(), snprintf()`
  - und andere
- Eine Codezeile wie printf(argv[1]); ist sehr gefährlich, da der Angreifer Formatanweisungen einschleusen kann und sich so (nicht für ihn bestimmte) Daten anzeigen lassen kann
  - Dies eignet sich z.B. für einen *Dump* des Speichers, um nach dem privaten Schlüssel zu suchen