Architektur von SmartCards und Embedded Systems

Informationstechnische Grundlagen II

- Authentisierung
- digitale Signatur
- Zertifikate

Seminarvortrag von Heiko Abraham

– Authentisierung –

Begriff allgemein:

- *authentisieren* = die Echtheit bezeugen, beglaubigen

ABER: *Authentisierung ≠ Identifizierung*

Bei SmartCards:

- Authentizität der Kommunikationspartners feststellen, *i.allg. Vertrauenswürdigkeit*
- Sicherheit auf Protokollebene

Vorgehen:

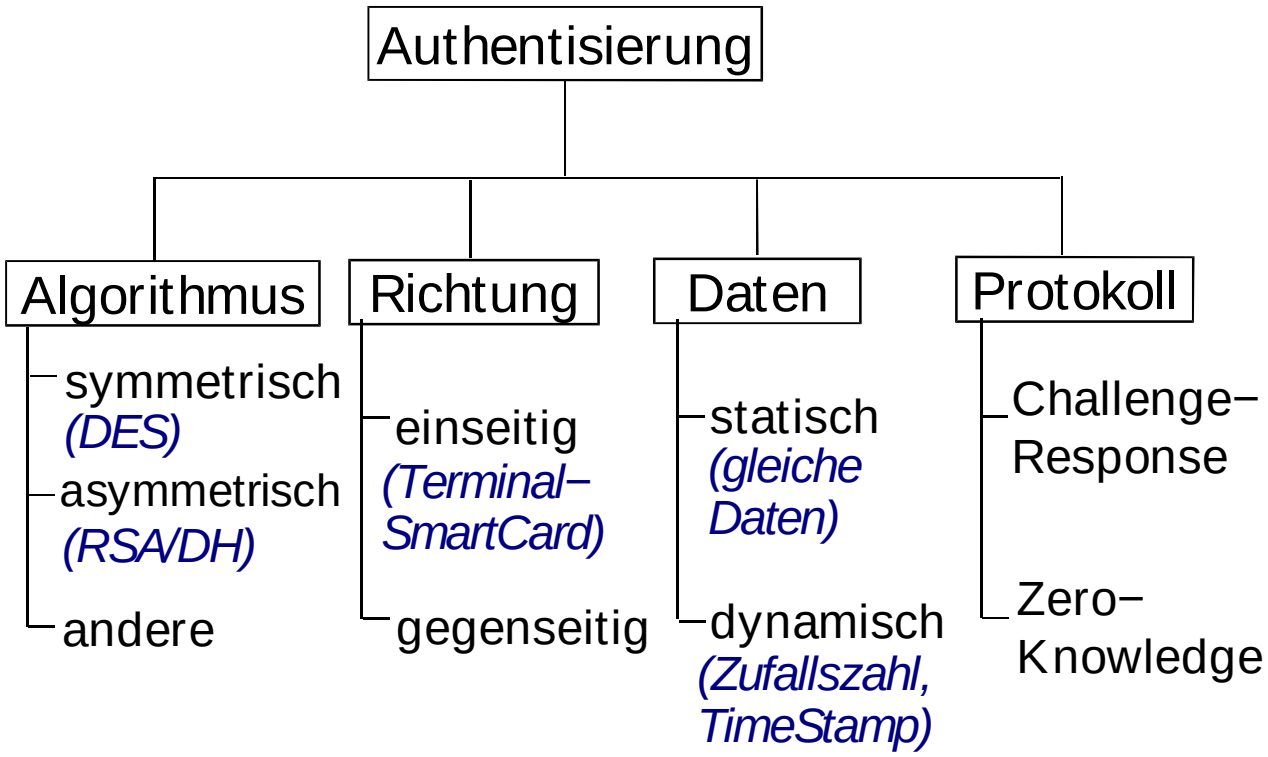
- Überprüfung eines gemeinsamen Geheimnisses (= Schlüssel)
- Geheimnis selbst aber ***nicht*** übertragen

➔ damit sicherer wie z.B. PIN

Voraussetzung:

- perfekte Verschlüsselungsalgorithmen
- Partner haben Geheimnis (Schlüssel)

Einteilung nach Charakteristik:



Challenge–Response:

- ein Partner stellt zufällige Frage *(challenge)*
- anderer Partner (berechnet) Antwort *(response)*

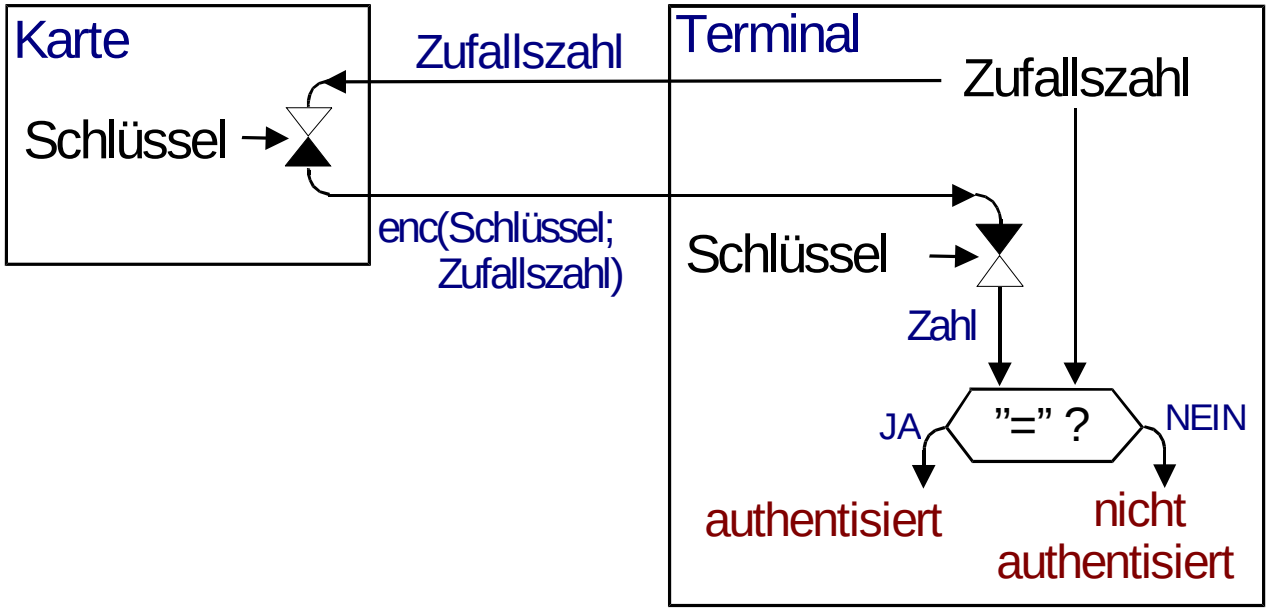
Zero–Knowledge:

- Terminal stellt Chipkarte *n* – viele Fragen
- Chipkarte antwortet entsprechend
- Angreifer könnte Antwort raten (zu 50% richtig)
- nach *n* Fragen ist Chipkarte authentisiert

  ➔ *Feige–Fiat–Shamir Proof of Identity*
  ➔ *Guillou–Quisquater Proof of Identity*

Einseitige sym. Authentisierung

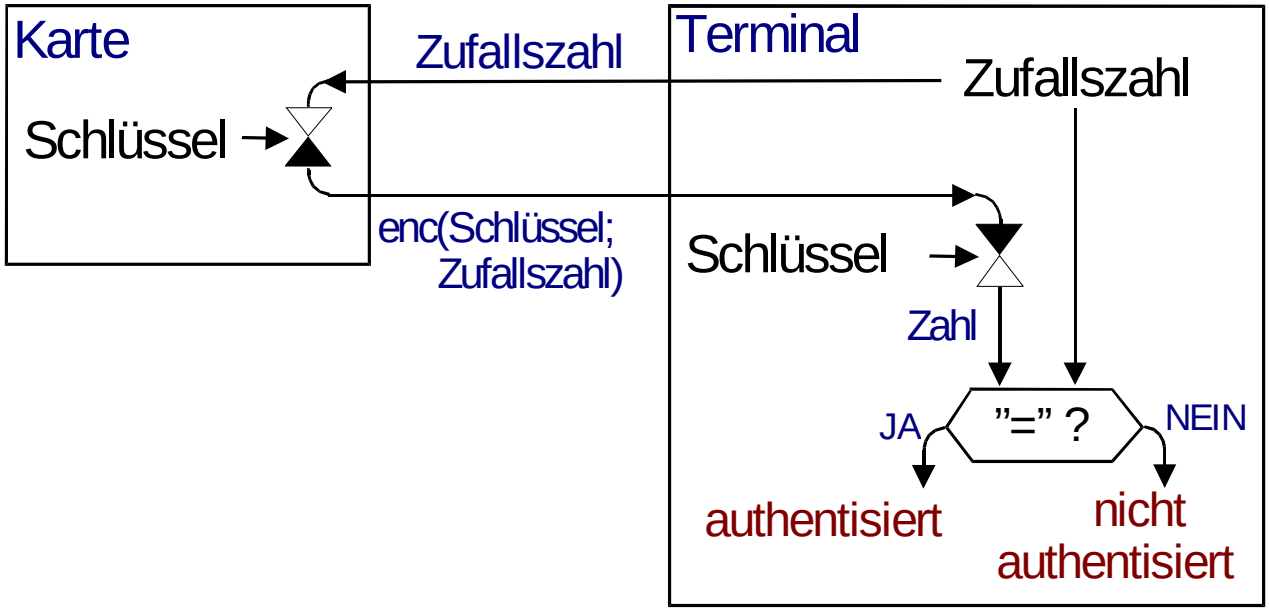
einseitig Chipkarte durch Terminal authentisieren (dynamisch, Challenge–Response)



Problem:
Terminal muß (individuell) Kartenschlüssel kennen

Einseitige sym. Authentisierung

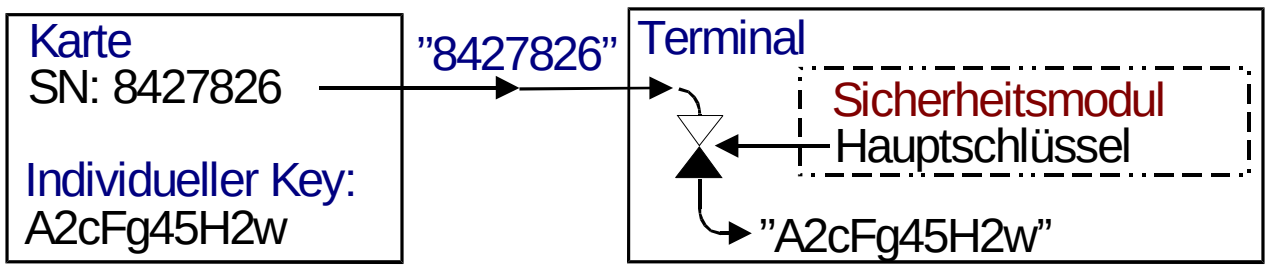
einseitig Chipkarte durch Terminal authentisieren (dynamisch, Challenge–Response)



Problem:

~~Terminal muß (individuell) Kartenschlüssel kennen~~

Lösung: individuellen Kartenschlüssel ableiten



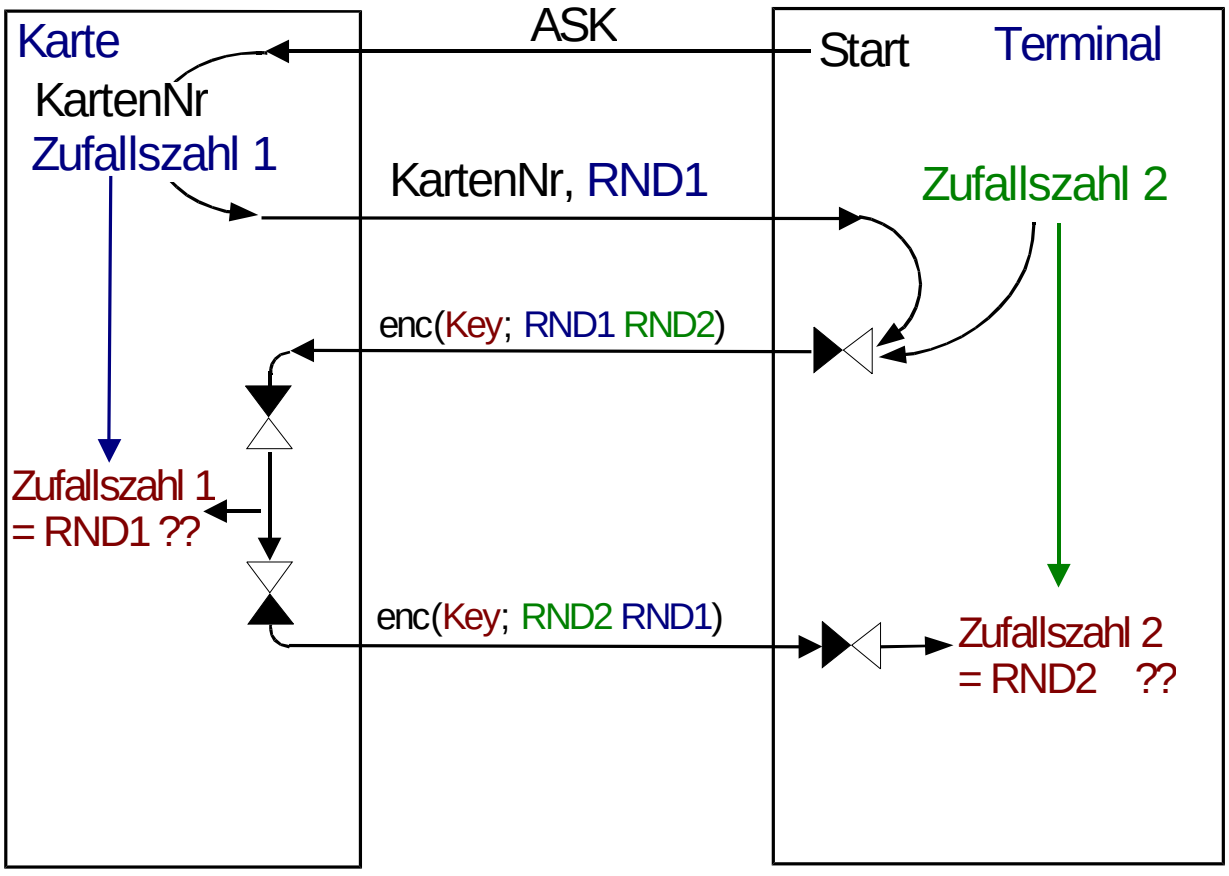
Gegenseitige sym. Authentisierung

Ziel: beide Seiten authentisiert

Voraussetzung: gemeinsamer sym. Schlüssel

Vorgehen:

- Im Prinzip zwei einseitige Authentisierungen
- mit Verflechtung Kommunikation reduzieren

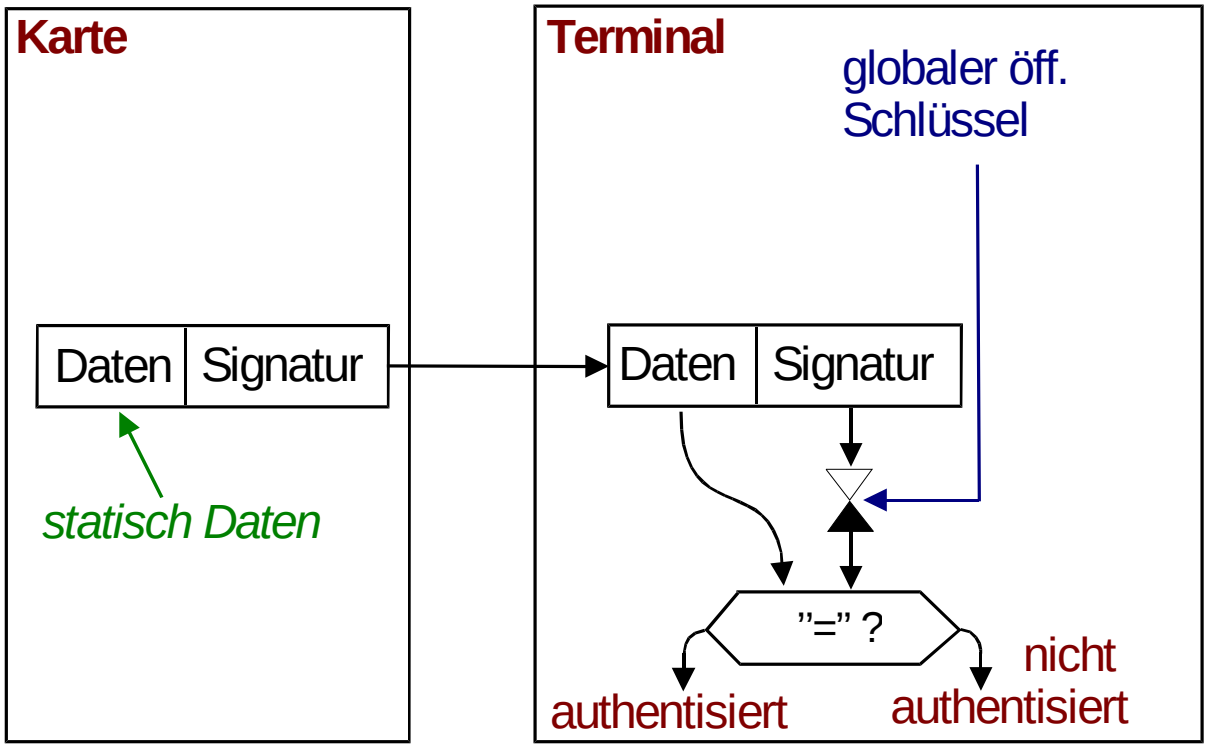


SmartCards und RSA:

- braucht viel Rechenzeit
- wenige Chipkarten haben RSA–Einheit

→ statische asym. Authentisierung

= Ausweg, Rechenzeit von Karte in Terminal
= nur einseitige Authentisierung
= verminderte Sicherheit



Probleme:

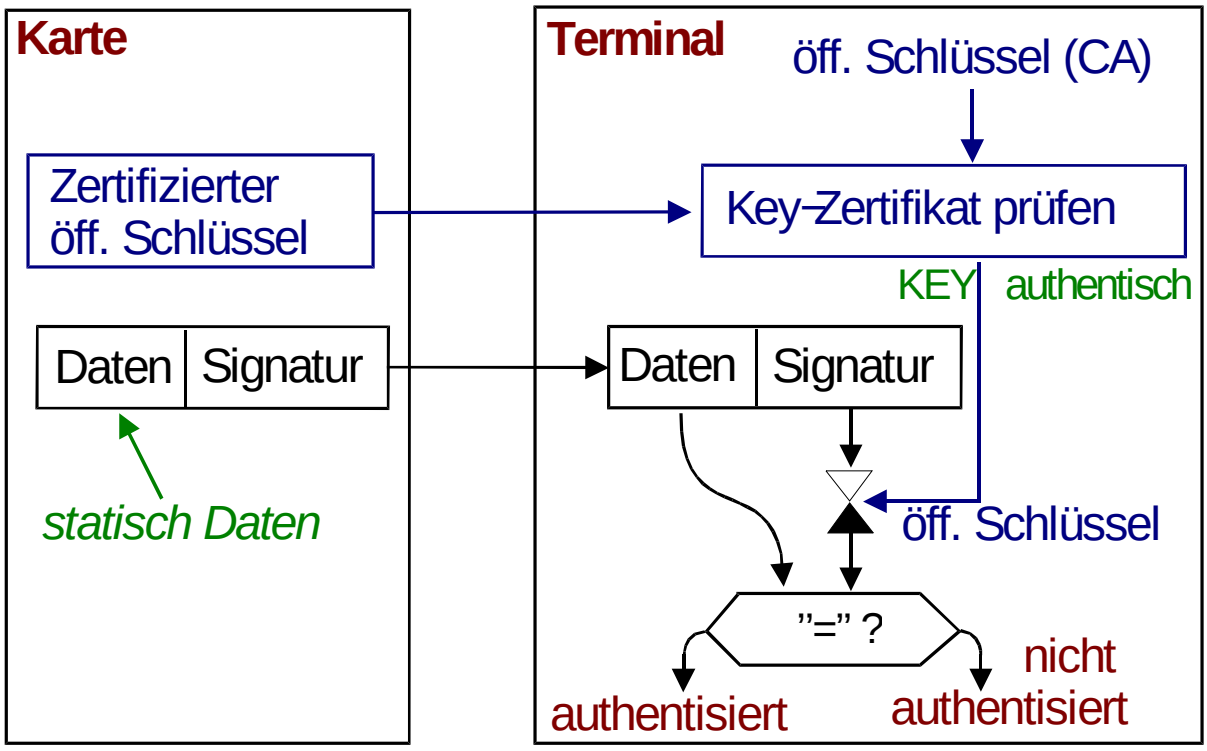
- nur ein öffentlicher Schlüssel

SmartCards und RSA:

- braucht viel Rechenzeit
- wenige Chipkarten haben RSA–Einheit

→ statische asym. Authentisierung

= Ausweg, Rechenzeit von Karte in Terminal
= nur einseitige Authentisierung
= verminderte Sicherheit



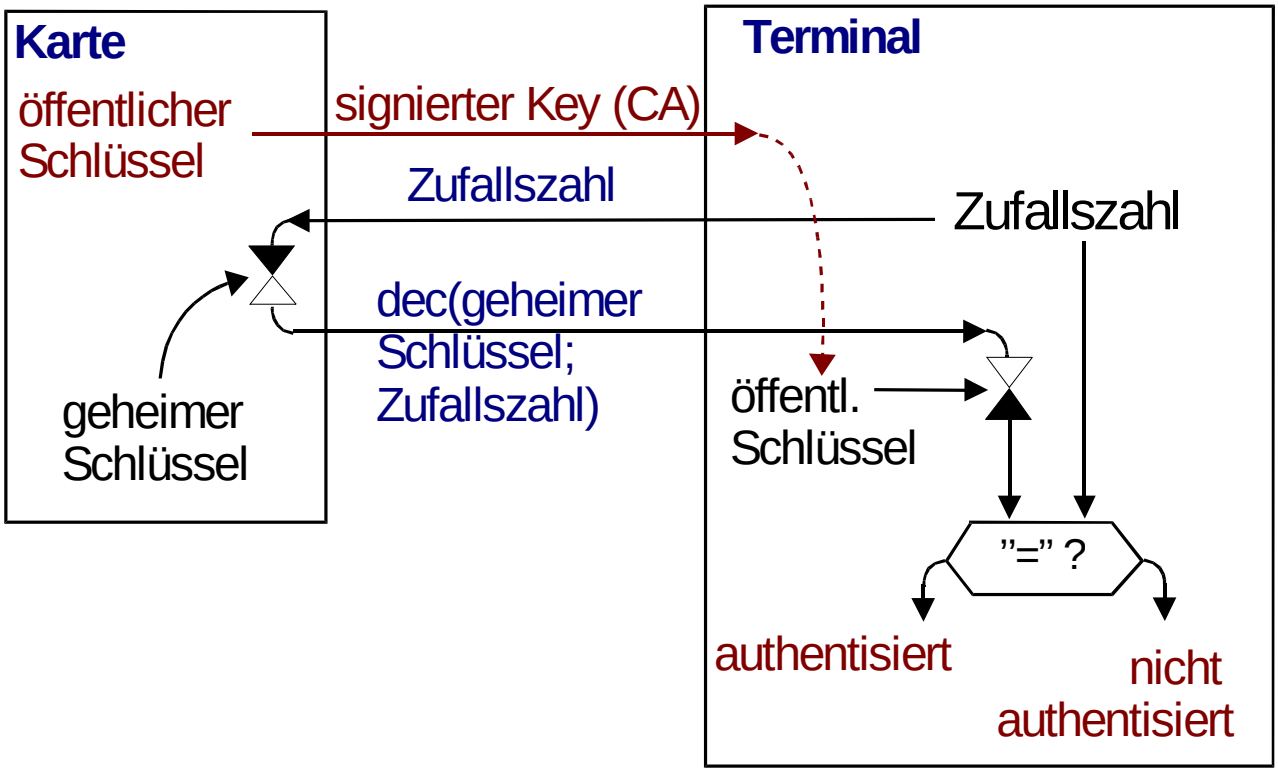
Probleme:

- ~~nur ein öffentlicher Schlüssel~~
- Replay–Angriff möglich

dynamische asym. Authentisierung

Ziel:

- Schutz vor »Replay« durch dynamische Daten



Praxis:

- RSA–Recheneinheit auf Karte
- Speicherbedarf

- bislang selten verwendet
- in Zukunft interessant

Überblick

Vergleich:

	Protokoll Iterations	***Amount of Calculation***	***Memory***	***Message Size***
Zero–Knowledge	many	large	large	large
public–key	one	very large	large	large
symmetric	one	small	small	small

Quelle: Hannu A. Aronsson, »Zero–Knowledge Protokolls and Small Systems«

In der Praxis:

- symmetrische Verfahren
- nur einseitige Authentisierung
- dynamischer Ablauf *(Zufallszahl)*
- Challenge–Response–Verfahren
- aber es exist. Zero–Knowledge–Ansätze

Sym. Verfahren – Probleme

- Hauptschlüssel wird bekannt *?!?*
- Schlüssellänge klein
- known–plaintext–attack

– Digitale Signatur –

Ziel:

- (dig.) Nachbildung der natürlichen Unterschrift
- Authentizität
- Integrität
- nicht wiederverwendbar

verankert im Signaturgesetz (SigG) u. §§415f ZPO

Vorgehen:

- asymmetrischen Algorithmen verwenden
  - ➔ erstellen: mit privaten Schlüssel
  - ➔ prüfen: mit öffentlichen Schlüssel
- aus Effizienzgründen: Hash–Wert verwenden

Begriffe unterscheiden !

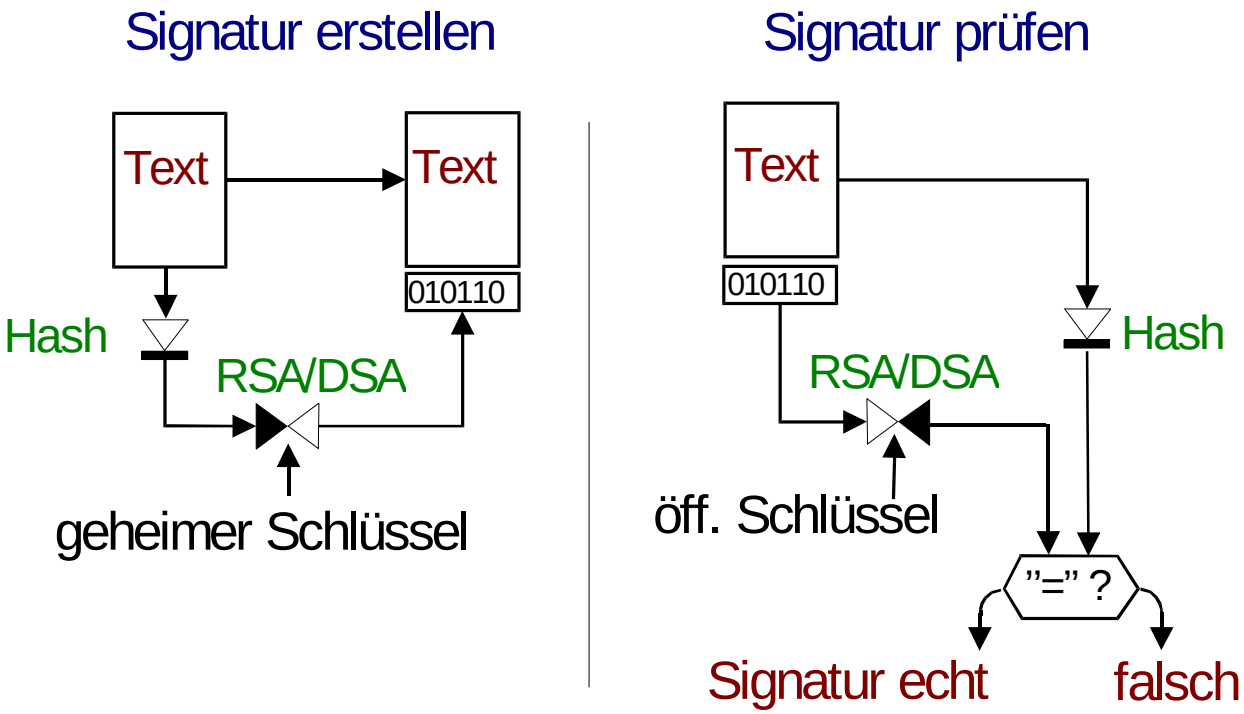
dig. Signatur ≠ Signatur (sym. Algorithmen)

Signatursysteme:

- RSA *(Rivest, Shamir, Adleman 1978)*
- DSA *(Digital Signature Algorithmus)*
- *...*

dig. Signatur: appendix:

Hash–Wert der Nachricht »verschlüsselt« nachstellen



Nachteil:

- verschieden Texte haben gleichen Hash–Wert
  → gute Hash–Funktionen verwenden

- Text ohne Überprüfung lesbar

Problem:

- eindeutige Schlüsselzuordnung → Zertifikate

– Zertifikate –

Ziel:

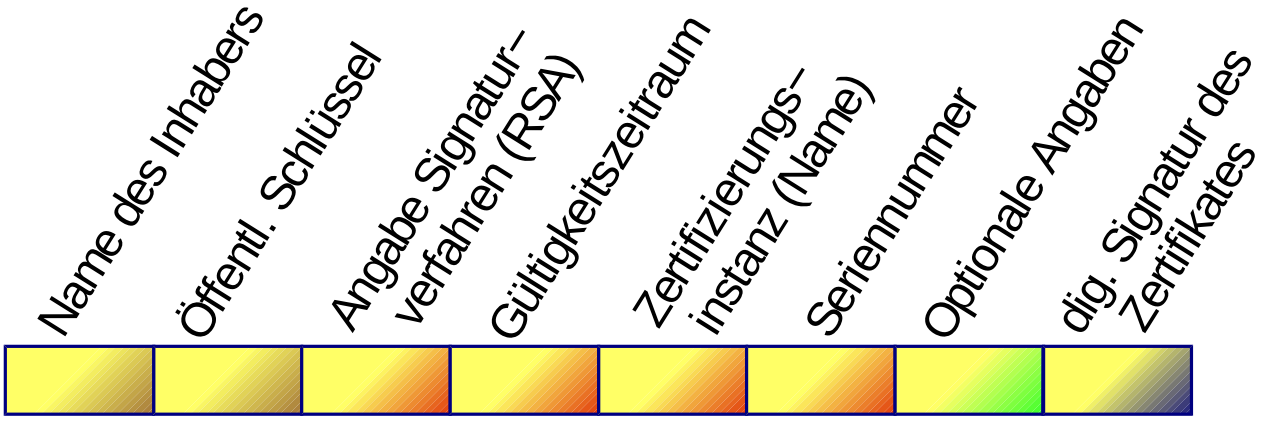
Zuordnung eines öffentl. Schlüssels zu Person

hierzu signiert *vertraueneswürdige Instanz* die Zuordnung des öffentl. Schlüssels = Zertifikat

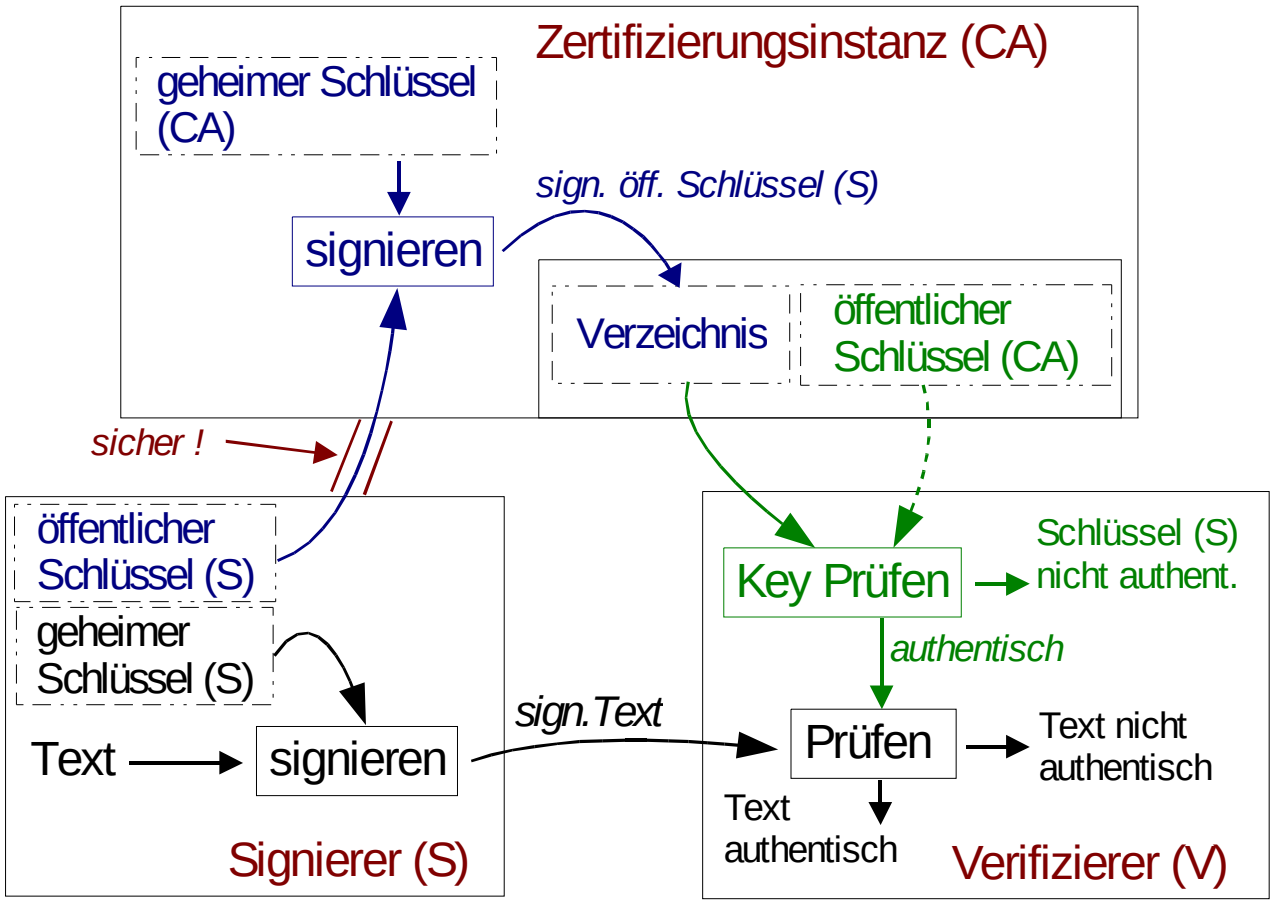
- Zertifikate sind testbar

  - CA (Certification Authority)
    - hierarchisch, zentralistisch, Root=RegTP
  - TrustCenter
    - Service, Schlüsselerzeugung

Zertifikat–Inhalt:



Anwendungsbeispiel:



– Literaturhinweis –

Standardwerk für dieses Seminar
W. Rankl, W. Effing, *Handbuch der Chipkarte*, Hanser–VerlagMünchen, 1999

Authentisierungsverfahren: ISO/IEC 9798 (Teil 1 bis 5)
http://www.x3.org/tc_home/t4htm/DOCS/9798–1.HTM
http://www.x3.org/tc_home/t4htm/DOCS/9798–2.HTM
http://www.x3.org/tc_home/t4htm/DOCS/9798–3.HTM
http://www.x3.org/tc_home/t4htm/DOCS/9798–4.HTM
http://www.x3.org/tc_home/t4htm/DOCS/9798–5.HTM

Überblick über Algorithmen, Angreifermodelle, Techniken...
St. Fisher, A. Steinacker, R. Bertram, R. Steinmetz, *Open Security*, Springer–Verlag, 1998

Paper zum Thema Zero–Knowledge–Protokolle und SmartCards
H. A. Aronsson, *Network Security: Zero Knowledge and Small Systems*,
http://www.tcm.hut.fi/Opinnot/Tik–110.501/1995/zeroknowledge.html

Vorlesungsfolien »Theorie und Praxis der IT–Sicherheit«, WS99/00, respektive Themen wie Signaturgesetz, CA's
http://www.informatik.uni–freiburg.de/~softech/

Artikel aus dreiteiligen Serie über Zertifizierungsstellen, Signaturen, Praxisbezug ...
Dipl.–Inf. R. Gehring, *Digitale Signaturen – Asymmetrisches*, Linux–Magazin 10/98
http://www.linux–magazin.de/ausgabe/1998/10/Signatur/signatur2.html

Paper über Angriffe auf SmartCards
Ross Anderson, Markus Kuhn, *Tamper Resistance – A Coutionary Note*, The Second USENIX Workshop on Electronic Commerce Proceedings, Oakland, California, Nov. 18–21,1996, pp1–11, ISBN 1–880446–83–9
http://www.cl.cam.ac.uk/users/rja14/tamper.html