Gesundheitstelematik

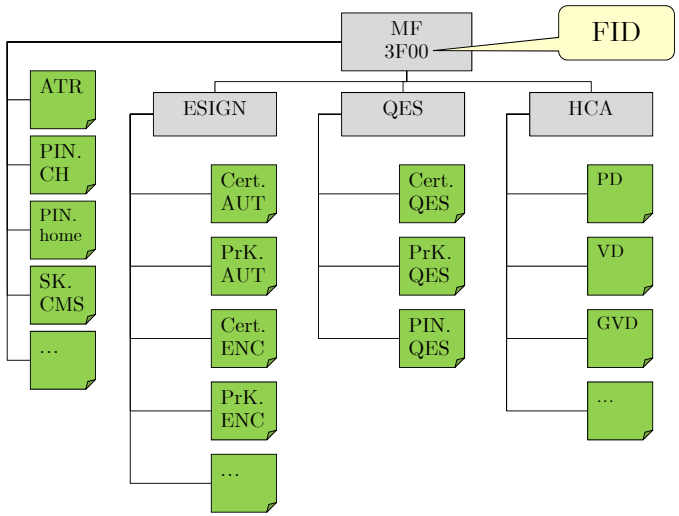
15. Vorlesung am 14.05.2012

m.gumbel@hs-mannheim.de

Dateisystem

- EF – Elementary File
  - entspricht einer Datei in anderen Dateisystemen
  - beinhaltet Daten
- DF – Dedicated File (Eselsbrücke: Directory File)
  - entspricht einem Verzeichnis in anderen Dateisystemen
  - beinhaltet EFs oder DFs
  - Master File (MF): Root-Verzeichnis
- Jedes Objekt (EF, DF) hat einen File-Identifier (FID)

Auszug aus dem Dateisystem der eGK



Legende:

- ATR : Answer to Reset
- CH: Card Holder
- SK.CMS: symm. Schlüssel Kartenherausgeber
- Cert.: Certificate
- AUT: Authentifation
- PrK.: Private Key
- ENC: Encryption
- QES: Qualified Signature
- HCA: Health Care Appl.
- PD: Persönliche Daten
- VD: Versichertend.
- GVD: geschützte Versichertendaten

Zugriffsschutz

- entspricht Login in eine Smart-Card
  1. PIN-Eingabe
  2. Authentifizierung mittels Challenge-Response über Zertifikate
     - Card to Card (C2C)
     - Card-Verifiable Certificates (CV-Zertifikate)
  - auch Kombinationen aus PIN und C2C möglich
- C2C ist Beispiel für Zugriff nur durch Besitz
- Zugriff auf Objekt (Algorithmus/Datei) abhängig vom Login-Status und Regeln hierzu
- Zustand bleibt bis
  - Karte Strom verliert (d.h. gezogen wird)
  - durch Befehl zurückgesetzt wird

Herstellung und Personalisierung

- OS und Dateisystemlayout werden final geschrieben
  - Ausnahme: Java Cards
- In einer sichereren Umgebung werden
  - personenbezogene Zertifikate und deren
  - private Schlüssel aufgebracht
  - PIN und ggf. PUKs initialisiert
- Hersteller hat Master-Schlüssel im ROM hinterlegt
- Kartenkörper und Chip werden zusammengebracht

Schnittstellen und Programmierung

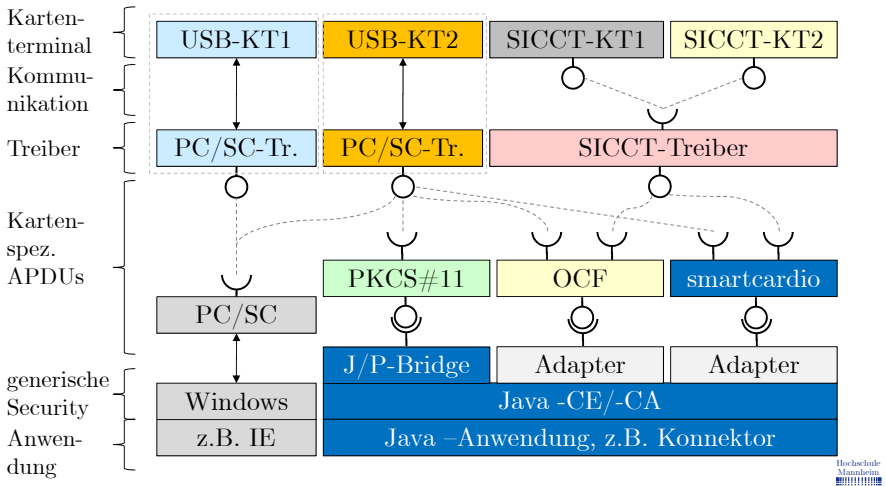
- Kartenterminal
  - Stromversorgung für Smart-Card
  - Low-Level-Kommunikation mit Karte
  - Schnittstelle zu anderen Systemen (z.B. via USB)
  - Sichere PIN-Eingabe und ggf. Display
- Insgesamt sehr viele Standards (siehe nächste Seite)
- Programmierung abhängig von
  - Eigenschaften der Karte (z.B. Dateisystemlayout)
  - Treibersoftware (incl. Betriebssystem)

Beispiel für APDU unter javax.smartcardio

```
TerminalFactory factory = TerminalFactory.getDefault();
List<CardTerminal> terminals = factory.terminals().list();
CardTerminal terminal = terminals.get(0);
System.out.println("KT #1: " + terminal);
System.out.println("Karte gesteckt? " + terminal.isCardPresent());
Card card = terminal.connect("T=1");
ATR atr = card.getATR();
for (byte b : atr.getBytes()) {
       System.out.print(Integer.toHexString(0x000000FF & (int) b) + " ");
}
System.out.println(); System.out.println(atr);
System.out.println("card: " + card);
CardChannel channel = card.getBasicChannel();
byte[] indata = { (byte) 0xd0, (byte) 0x01 };
CommandAPDU cin = new CommandAPDU(0, 0xa4, 0x01, 0x0c, indata);
ResponseAPDU r = channel.transmit(cin);
System.out.println("response: " + r);
byte[] outdata = r.getData();
```

```
KT #1: PC/SC terminal HP USB Smart Card Keyboard 0
Karte gesteckt? true
3b df 96 ff 81 b1 fe 45 1f 3 0 64 4 5 6 0 31 be 73 9e 21 53 0 90 0 c2
ATR: 26 bytes
card: PC/SC card in HP USB Smart Card Keyboard 0, protocol T=1, state OK
response: ResponseAPDU: 2 bytes, SW=6a82
```


Beispiel für Smart-Card-API-Universum



Legende zur vorigen Folie

- SICCT: Secure Interoperable ChipCard Terminal
- PC/SC: Interoperability Specification for ICCs and Personal Computer Systems
  - kurz: Personal Computer/ Smart Card
- OCF: Open Card Framework (Java)
- PKCS#11: Public Key Cryptography Standards, Standard Nr. 11
- J/P-Bridge: Java-PKCS#11-Bridge
- JCE/JCA: Java Cryptography Extension/Architecture

Das war‘s

Gesundheitstelematik

Kapitel: Sicherheit

Abschnitt: Verteilte und verschlüsselte Dateisysteme

Einführung

- Smart-Card-Dateisystem sicher, aber zu klein
- → Auslagerung in eine serverseitige Datenbank
- Anforderungen hierfür
  - unlesbar; selbst für Systemadministratoren
  - keine Profilbildung oder Statistiken usw.
- Mögliche Lösungen
  - Verschlüsseltes Dateisystem
  - Verschlüsselte Datenbank (z.B. Oracle)
  - Nachteil: Entschlüsselung durch Systemadministrator

End-to-End Security

- Besser: Daten werden „so nah wie möglich“ am Benutzer (End-2-End) ver- und entschlüsselt
- Hier: im Konnektor
- Vorteil: Egal wo Daten gespeichert sind, sie sind unlesbar; sogar für Systemadministratoren
- Aber: wie können andere Personen für diese Daten autorisiert werden?
  - d.h. Delegation von Rechten
  - wäre mit technischem User in einer DB möglich

heise online

05.05.2009 12:01

Cracker fordern 10 Millionen US-Dollar für Patientendatenbank

Kriminelle sollen **US-Medienberichten[1]** zufolge die Daten von 8 Millionen amerikanischen Schmerzpatienten vom Server des Virginia Prescription Monitoring Program gestohlen haben und nun 10 Millionen US-Dollar Lösegeld fordern. Der Server dient Ärzten zur Überwachung der Herausgabe von Rezepten für Schmerzmittel wie Opiate und soll den Drogenmissbrauch verhindern.

Nach Angaben von Brian Krebs von der Washington Post drangen die Kriminellen auf unbekanntem Weg in den Server ein, verschlüsselten die Daten, löschten die Originale anschließend auf dem Server und hinterließen den Erpresserbrief. Dieser wurde auch auf **Wikileaks[2]** veröffentlicht:

"I have your shit! In *my* possession, right now, are 8,257,378 patient records and a total of 35,548,087 prescriptions. Also, I made an encrypted backup and deleted the original. Unfortunately for Virginia, their backups seem to have gone missing, too. Uhoh :(For $10 million, I will gladly send along the password."

Der Server ist derzeit nicht mehr erreichbar. Nach Angaben des Virginia's Department of Health Professions sind die Ermittlungsbehörden bereits eingeschaltet.

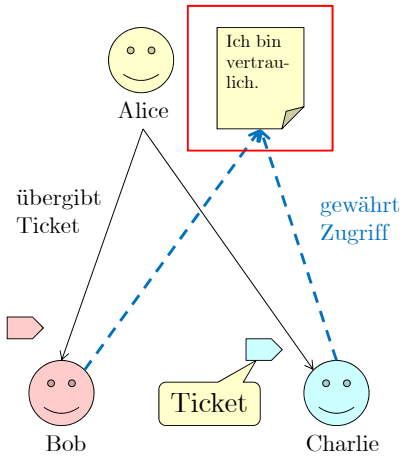
(**dab[3]**/c't)

URL dieses Artikels:
http://www.heise.de/newsticker/meldung/137281

Links in diesem Artikel:
[1] http://voices.washingtonpost.com/securityfix/2009/05/hackers_break_into_virginia_he.html
[2] http://wikileaks.org/wiki/Over_8M_Virginian_patient_records_held_to_ransom,_30_Apr_2009
[3] mailto:dab@ct.heise.de

Ticket-Konzept

- Ticket: „Eintrittskarte“, die Zugriff auf bestimmte Daten ermöglicht
  - Gültigkeitsdauer
  - kann gespeichert werden
- Tickets
  - werden individuell vergeben
  - beziehen sich (zunächst) auf ein Dokument
- Idee aus Kerberos entnommen



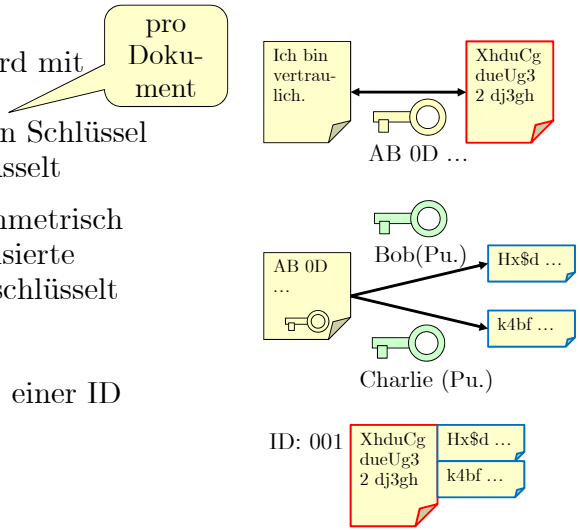
Gruppenarbeit

Aufgabe

- Wie würden Sie ein Ticket-Konzept realisieren?
- Überlegen Sie in 2er Gruppen
- Zeit: 5 min

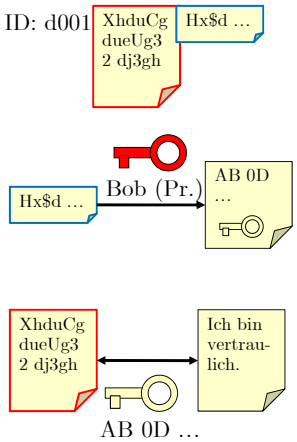
Erstellen eines sicheren Dokuments

1. Dokument wird mit generiertem symmetrischen Schlüssel (SK) verschlüsselt
2. SK wird asymmetrisch für alle autorisierte Personen verschlüsselt
3. Dies wird mit einer ID gespeichert



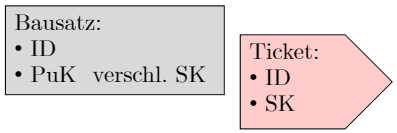
Zugriff auf ein sicheres Dokument

1. Bob erhält verschl. Dokument und verschl. SK
2. Er entschlüsselt SK mit privaten Schlüssel
3. Dokument wird symmetrisch mit SK entschlüsselt



Eigenschaften

- Ticket wird indirekt über Server und nicht vom Inhaber vergeben
  - Bob *sucht* Dokument und
  - erhält Ticket-Bausatz
- Besitzer des privaten Schlüssels kann daraus Ticket erstellen



  - enthält alle Information, um auf Datum zuzugreifen
  - geschieht z.B. Konnektor
- Für Autorisierung eines Benutzers ist dessen Zertifikat (=Public Key) nötig
  - von Smart-Card (z.B. HBA falls Arzt o. Apotheker)
  - von Public-Key-Verzeichnisdienst