11. Datenschutz und Zugriffskontrolle in DBS

- **Beobachtung**
  - **Immer mehr Daten werden gespeichert,**
    von Programmen analysiert und zwischen ihnen **ausgetauscht**.
  - Neue Dimensionen beim Sammeln von Daten und dem daraus resultierenden Gefährdungspotential:
    E-*, Data Warehousing, Data Mining, Semantic Web, . . .
  - Wesentliche Schwachpunkte existierender Schutzkonzepte:
    mangelnde Differenzierbarkeit und Einheitlichkeit
  - Die Anzahl der Angreifer (Schnüffler, Hacker, Viren, . . .) nimmt zu!
    Deshalb sind Verlässliche Systeme gefragt!
    (Verfügbarkeit, Sicherheit, Schutz vor Attacken, Vertrauenswürdigkeit, ...)
- **BDSG – Übersicht**
  Legislative Maßnahmen zum Datenschutz
- **Wer sind die Angreifer?**
- **Technische Maßnahmen des Datenschutzes**
  - Zutritts- und Zugangskontrolle, Authentisierung
  - Weitergabekontrolle, Verfügbarkeitskontrolle, Zugriffskontrolle
- **Autorisierungsmodell mit expliziten Zugriffsrechten**
- **Zugriffskontrolle in SQL**
  - Vergabe und Kontrolle von Zugriffsrechten
  - Probleme des Rechteentzugs
- **Verfeinerung des Autorisierungsmodells**
  - Implizite Autorisierung bei Hierarchien
  - Rollenkonzept in SQL
- **Sicherheitsprobleme in statistischen DBs**
  - Inferenzkontrolle
  - Individuelle und allgemeine Tracker

Datenschutz

- **Legislative Maßnahmen (Datenschutzgesetze)[1]**
  - **Festlegung, welche Daten in welchem Umfang schutzbedürftig sind**
  - **Vorschriften, die Missbrauch der Daten entgegenwirken**

  (Festlegung, welche Daten von wem gespeichert werden dürfen,
  welcher Zugriff auf Daten erlaubt ist,
  welche Weitergabe der Daten zulässig ist usw.)

- **BDSG will schutzwürdige Belange der Betroffenen schützen**

a) Allgemeines Verbot der Verarbeitung personenbezogener Daten mit Erlaubnis gewisser Ausnahmen
   ➥ ***Verbotsprinzip mit Erlaubnisvorbehalt***

b) Gewährung spezieller Rechte für die Betroffenen
   (Auskunft, Berichtigung, Sperrung, Löschung)

c) Einführung besonderer Maßnahmen technischer und organisatorischer Art

- **Technische Maßnahmen**

1. **Zutritts- und Zugangskontrolle, Authentisierung**
2. **Weitergabekontrolle in Rechnernetzen**
3. **Zugriffs- und Verfügbarkeitsskontrolle:**
   Isolation der Benutzer und Betriebsmittel, Schutz der Geräte
4. **Zugriffskontrolle: Autorisierung des Zugriffs auf gemeinsame Daten**
5. **Datenflusskontrolle beim Datentransport**
6. **Inferenzkontrolle bei statistischen DB**

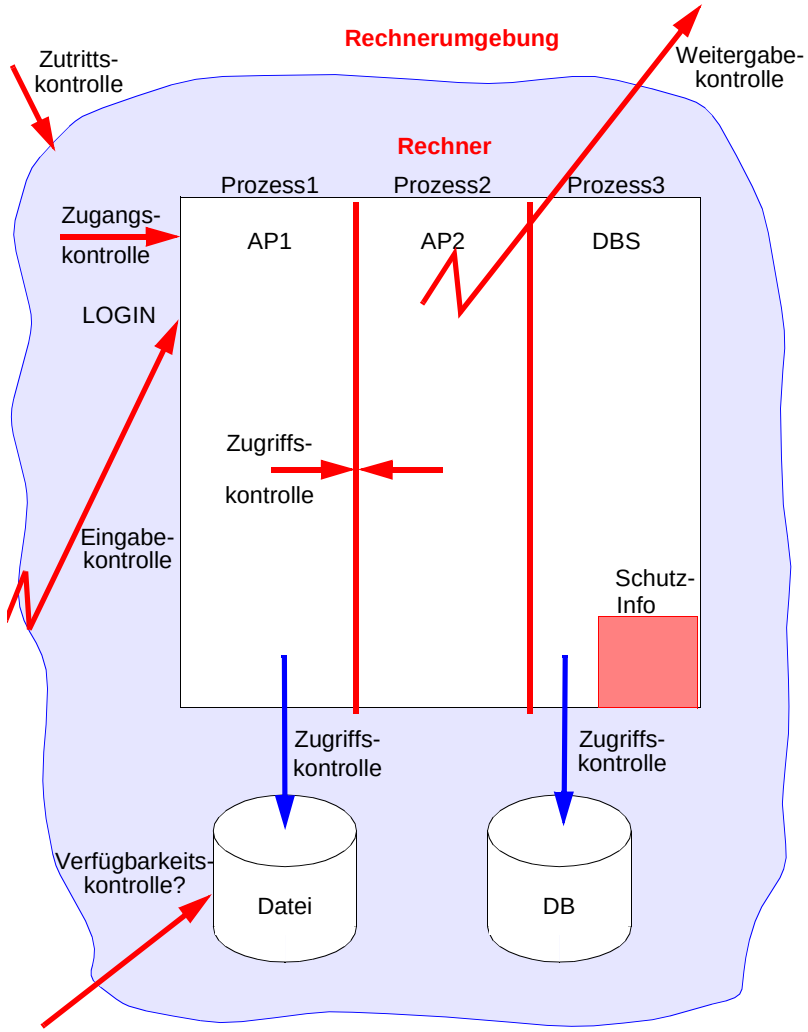
1. http://www.datenschutz-berlin.de/recht/de/bdsg/bdsg03.htm

BDSG-Anlage (zu § 9 Satz 1, vom 14. Januar 2003)
Technische Maßnahmen

Werden personenbezogene Daten automatisiert verarbeitet oder genutzt, ist die innerbehördliche oder innerbetriebliche Organisation so zu gestalten, dass sie den besonderen Anforderungen des Datenschutzes gerecht wird. Dabei sind insbesondere Maßnahmen zu treffen, die je nach der Art der zu schützenden personenbezogenen Daten oder Datenkategorien geeignet sind,

1. Unbefugten den Zutritt zu Datenverarbeitungsanlagen, mit denen personenbezogene Daten verarbeitet oder genutzt werden, zu verwehren (**Zutrittskontrolle**),
2. zu verhindern, dass Datenverarbeitungssysteme von Unbefugten genutzt werden können (**Zugangskontrolle**),
3. zu gewährleisten, dass die zur Benutzung eines Datenverarbeitungssystems Berechtigten ausschließlich auf die ihrer Zugriffsberechtigung unterliegenden Daten zugreifen können, und dass personenbezogene Daten bei der Verarbeitung, Nutzung und nach der Speicherung nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können (**Zugriffskontrolle**),
4. zu gewährleisten, dass personenbezogene Daten bei der elektronischen Übertragung oder während ihres Transports oder ihrer Speicherung auf Datenträger nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können, und dass überprüft und festgestellt werden kann, an welche Stellen eine Übermittlung personenbezogener Daten durch Einrichtungen zur Datenübertragung vorgesehen ist (**Weitergabekontrolle**),
5. zu gewährleisten, dass nachträglich überprüft und festgestellt werden kann, ob und von wem personenbezogene Daten in Datenverarbeitungssysteme eingegeben, verändert oder entfernt worden sind (**Eingabekontrolle**),
6. zu gewährleisten, dass personenbezogene Daten, die im Auftrag verarbeitet werden, nur entsprechend den Weisungen des Auftraggebers verarbeitet werden können (**Auftragskontrolle**),
7. zu gewährleisten, dass personenbezogene Daten gegen zufällige Zerstörung oder Verlust geschützt sind (**Verfügbarkeitskontrolle**),
8. zu gewährleisten, dass zu unterschiedlichen Zwecken erhobene Daten getrennt verarbeitet werden können.

Technische Maßnahmen des Datenschutzes



➥ Das „schwächste Glied“ in einer Kette von Maßnahmen bestimmt die Sicherheit des Gesamtsystems!

Wer sind die Angreifer?

- **Klassifikation:**
  - Mitarbeiter (allergrößte Gruppe der Angreifer)
  - Wannabees („Ich will Hacker werden“; sie finden es „cool“)
  - Script-Kidies
  - Cracker (Zerstören Systeme!)
  - Hacker- Gemeinde ist nicht monolithisch[2]
    - „black hats“: kriminell motivierte Cracker
    - „grey hats“: schwanken zwischen guter und böser Absicht
    - **„white hats“**: haben lautere Motive, etwa um durch Aufdecken von Sicherheitslücken IT-Systeme zu verbessern (Datenklau mit Rückgabegarantie!). Viele dieser „ethischen Hacker“ arbeiten irgendwann als IT-Profis in Unternehmen oder machen sich als Berater selbstständig.
  - Gurus & Wizards (kriegen fast jede Kiste auf!)
  - Professionelle Industriespione, Geheimdienste

Der Chaos Computer Club[3]

- **Ziele:**
  - Freiheit der Information, Recht auf Kommunikation, Informationelle Selbstbestimmung
  - Forum für kreative Techniknutzer („Hacker“)
  - kritische Analyse und Aufzeigen der Gefahren der Informationsgesellschaft
  - Cyber-Rights, Lobbyarbeit, Verbreitung der Hacker-Ethik

2. www.elfqrin.com/docs/BeingHacker.html, www.plethora.net/~seebs/faqs/hacker.html
3. www.ccc.de

Die Hackerethik

- Der Zugang zu Computern und allem, was einem zeigen kann, wie diese Welt funktioniert, sollte unbegrenzt und vollständig sein!
- Alle Informationen müssen frei sein!
- Misstraue Autoritäten – fördere Dezentralisierung!
- Beurteile einen Hacker nach dem, was er tut und nicht nach üblichen Kriterien wie Aussehen, Alter, Rasse, Geschlecht oder gesellschaftlicher Stellung!
- Man kann mit einem Computer Kunst und Schönheit schaffen!
- Computer können dein Leben zum Besseren verändern!
- *Mülle nicht in den Daten anderer Leute!*
- *Öffentliche Daten nützen, private Daten schützen!*

Information Warfare

- **Und International?[4]**
  - Gemeinsame Erklärung von CCC, 2600, L0pht, Phrack, Cult of the Dead Cow, Pulhas, !Hispahack u. a.
  - Mehr als 120 Hackergruppen haben unterzeichnet
  - „We - the undersigned - strongly oppose any attempt to use the power of hacking **to threaten or destroy the information infrastructure of a country**, for any reason. Declaring „war“ against a country is the most irresponsible thing a hacker group could do. This has nothing to do with hacktivism or hacker ethics and is nothing a hacker could be proud of.“
  - „The signatories to this statement are asking hackers to reject all actions that seek to damage the information infrastructure of any country. **DO NOT support** any acts of „**Cyberwar**“. Keep the networks of communication alive. They are the nervous system for human progress“.

4. 2600: The Hacker Quaterly (www.2600.com)

Zutritts- und Zugangskontrolle

- **Kernfrage 1:**
  Wie erkennt ein Rechensystem einen berechtigten Benutzer?

  ➥ **Frage nach der Identifikation/Authentisierung**

- **Organisatorische Maßnahmen**

  (Zutrittskontrolle, bauliche Maßnahmen, . . .)

- **Identitätskontrolle (Authentisierung)**

  Nachweis der Identität des Benutzers gegenüber Transaktionssystem bzw. gegenüber BS und DBS

  ➥ Verfahrensklassen bei Authentisierung

- **Standard: Passwortmethoden**

  Benutzer — System — Passwort-Datei

  Id =

  PW =

Id	PW

- **Was man ist, was man hat, was man weiß**
  - Benutzercharakteristika werden überprüft (Stimme, Handgeometrie, Fingerabdruck, Unterschrift, . . ., „der Körper als Ausweis“).
  - Ausgehändigte Gegenstände ermöglichen Zugriff (Schlüssel für Terminal, maschinell lesbare Ausweise).
  - Authentisierung mittels Wissen (Frage-Antwort-Methoden, Challenge-Response-Verfahren)

Authentisierung

- **Kernfrage 2:**
  Wie kann ich mich gegenüber einen anderen zweifelsfrei ausweisen?
  Wie kann ich sicher sein, dass eine Nachricht wirklich von dem anderen Sender stammt?

  ➥ **Frage nach der Authentisierung von Systemen/Dokumenten**

- **Authentisierung**[5]
  - Nachweis der Identität des Benutzers (Netzknoten, Dokument)
  - Authentisierung bezieht sich auf die Quelle der Information (Sender-Authentisierung) und auf ihren Inhalt (Datenintegrität).

- **Beidseitige Authentisierung für Rechner-Rechner-Kommunikation**
  - Challenge-Response-Verfahren
  - Einigung auf kryptographisches Verfahren, Schlüsselaustausch
  - Authorization-Encryption: Chiffrierschlüssel mit Gültigkeit für die gesamte Sitzung (*Session*) dient der Autorisierung (ohne ständig die beidseitige Authentisierung wiederholen zu müssen).

- **Nachrichtenauthentisierung**
  - Unterschriften, Echtheitsmerkmale
  - Der Ersteller besitzt etwas, mit dessen Hilfe er das Dokument authentisch macht.

- **Aber manchmal ist auch Anonymität erwünscht!**
  - Wie kann das mit Rechnern simuliert werden?
  - Anonymität und Verlässlichkeit: Prisoner’s Dilemma[6]

5. authentisieren = für glaubwürdig, rechtsgültig machen; oft auch: authentifizieren = die Echtheit bezeugen, beglaubigen
6. http://serendip.brynmawr.edu/playground/pd.html

Weitergabekontrolle

- **Kernfrage 3:**
  Wie kann ich mit jemand vertraulich kommunizieren?

  ➥ **Frage nach der Geheimhaltung der Information**

Neben organisatorischen und baulichen Maßnahmen hier vor allem

- **Kryptographische Maßnahmen**
  - Symmetrische Verfahren
    - Schlüssel K wird zum Ver- und Entschlüsseln verwendet (wenig Aufwand)
    - Ersetzungs- und Versetzungsverfahren (DES: *Data Encryption Standard*)
  - Asymmetrische Verfahren
    - Sie beruhen auf dem Einsatz von zwei einander zugeordneten Schlüsseln S (secret) und P (public)[7]
    - RSA-Verfahren ist am bekanntesten (R. Rivest, A. Shamir, L. Adleman)

➥ Sie heißen auch Public-Key-Verfahren
(typischerweise Faktor 1000 langsamer als symmetrische Verfahren)

- aber auch: Steganographie

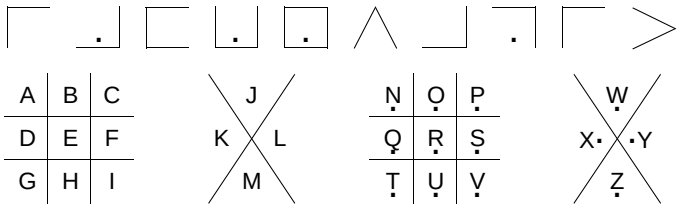
➥ Wer das „Geheimnis“ kennt, kommt auch an die Information!

7. Sie werden aus Primzahlen mit oft ~120 Stellen abgeleitet. Der Primzahl-Rekord liegt bei 9,15 Mio. Stellen: die in 2006 größte bekannte Primzahl lautet $2^{30\,402\,457} - 1$

Kryptographische Verfahren

- **Kryptographie**
  - befasst sich mit dem Ver- und Entschlüsseln von Nachrichten
  - Sicherheit ergibt sich aus der Qualität des eingesetzten Algorithmus (nicht aus der Verschleierung des Verfahrens ➥ Steganographie).

- **Beispiele aus der Geschichte**
  - Caesar-Chiffre:
    Jeder Buchstabe des Alphabets
    wird durch seinen Nachfolger ersetzt.
  - Freimaurer (16. Jhd.):
    Ersetzung der Buchstaben durch geometrische Figuren



  - Spanische Geheimschrift (16. Jhd.):
    Ersetzung von Buchstabenpaaren durch spezielle Zeichen:
    vermutlich $25^2 = 625$ Zeichen

- **Verschlüsselungsbeispiel:**

Nachricht M:	Das ist Klartext
Schlüssel K:	azxbazxbazxba
Chiffre C:	xywwusrfeqzphj

- **Schlüsselaustausch**
  - Sender und Empfänger müssen das „Geheimnis“ kennen.
  - Schlüssel K (oder S/P) erlaubt die Entschlüsselung.

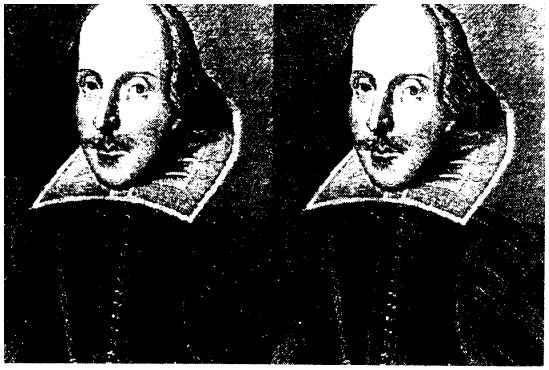
Steganographie

- **Ziel: Verschlüsselte Informationen so zu speichern, dass**
  - niemand diese Informationen findet und dass
  - niemand beweisen kann, dass verschlüsselte Informationen da sind.

  ➥ liefert Argumente für die Gegner staatlich kontrollierter Chiffrierverfahren!

- **Mögliche Anwendung**

  Nutzung der niederwertigsten Bits in Daten vom Typ Bild, Ton, . . .



Das linke Bild ist das Original, im rechten Bild ist der Text *Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the „enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other „harmless" messages in a way that does not allow any „enemy" to even detect that there is a second secret message present [Markus Kuhn 1995-07-03] versteckt.*

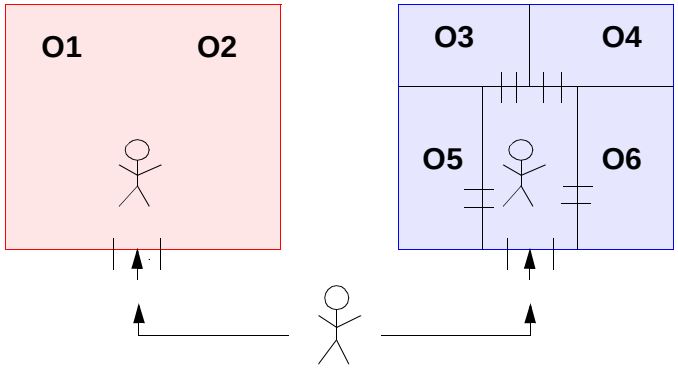
Verarbeitungs- und Zugriffskontrolle

- **Kernfrage 4:**

  Wie kann ich erreichen, dass die unbefugte Benutzung von Systemressourcen unterbleibt?

  ➥ **Frage nach der Verarbeitungskontrolle (bei Prozessen und Dateien)**

- **Prozesse und Virtuelle Adressräume**
  - Isolation durch Prozess
  - Prozess ist oft Einheit der Adressierung, der Betriebsmittelvergabe sowie des Schutzes
  - **Analogie:** Haus mit Zimmern

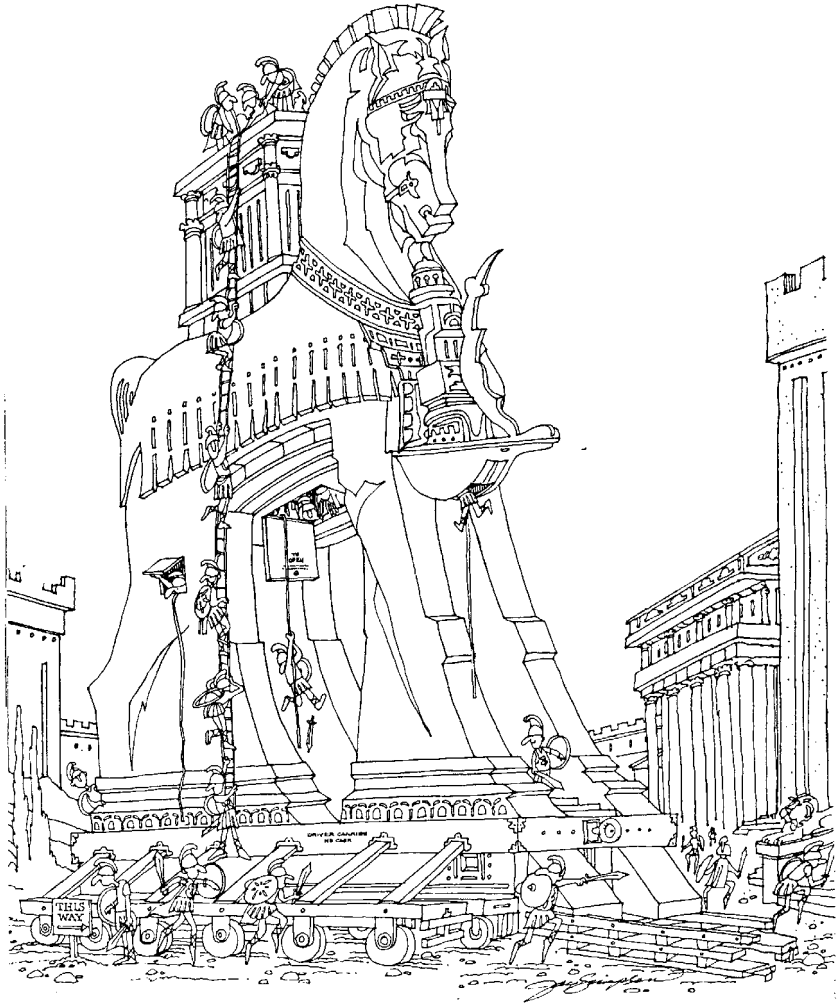


- **Kontrollprobleme bei gemeinsamer Nutzung oder Infiltration**
  - nur Eingangskontrolle
  - Problem des Trojanischen Pferdes
- **Verbesserung**
  - Zugriff auf einzelne Dateien
  - Kontrolle durch Passwort (Schlüssel)
  - Problem der vielen Schlüssel (Gruppenschlüssel)

  ➥ aber als Zugriffsprinzip: „alles oder nichts"!

‘Trojan Horse’-Problem

- **Problem der Zugriffsbeschränkung bei Programmen**
  („Erschleichung“ der Zugriffsrechte eines Benutzers)



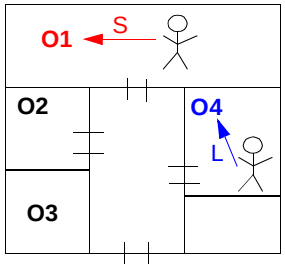
Zugriffskontrolle

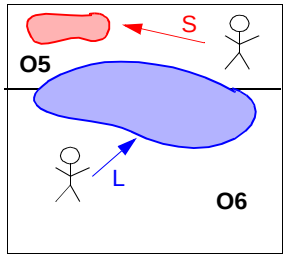
- **Kernfrage 5:**
  Wie kann ich erreichen, dass Benutzer mit unterschiedlichen Rechten gemeinsam auf Daten zugreifen können?

  ➥ **Frage nach der Zugriffskontrolle (bei Daten)**

- **Zugriffskontrolle (Autorisierung)**
  - Vergabe von Zugriffsrechten (Lesen, Schreiben, . . .) auf DB-Objekten, Programmen usw.
  - Ziele
    - Verhinderung von zufälligen oder böswilligen Änderungen
    - möglichst weitgehende Isolation von Programmfehlern
    - Verhinderung von unberechtigtem Lesen/Kopieren

- **Analogie**





minimale Objektgranulate
(z. B. durch Sichten)

- **Kontrollkonzepte:**
  - unbeschränkte Zugriffskontrolle (Schlüssel + Privileg)
  - Teilordnung der Nutzungsprivilegien
  - Prinzip des kleinstmöglichen Privilegs
    - Need-to-know-Prinzip
    - **Least Privilege Principle**
  - kooperative Autorisierung (n Schlüssel)

Autorisierungsmodell

- **Explizite Autorisierung[8]:**
  - Der Zugriff auf ein Objekt o kann **nur** erfolgen, wenn für den Benutzer (Subjekt s) ein Zugriffsrecht (Privileg p) vorliegt
  - Autorisierungsregel (o, s, p) legt eine **explizite starke** Autorisierung mit **positivem** Recht fest
- **Schutzinformation als Zugriffsmatrix**

  Subjekte: Benutzer, Programme, Terminals

  Objekte: Programme (Anwendungs-, Dienstprogramme), DB-Objekte (Relationen, Sichten, Attribute)

  Zugriffsrechte: Lesen, Ändern, Ausführen, Erzeugen, Weitergabe von Zugriffsrechten usw., ggf. abhängig von Terminal, Uhrzeit usw.

Subjekte, Benutzer	Objekte				
	O1	O2	O3	...	On
B1	P1, P2		P3		Pi
B2		P1	P2, P3		P1
B3		P2, P3	P2		
• • •					
Bm	P1, P2	Pi	P1		Pi, Pk

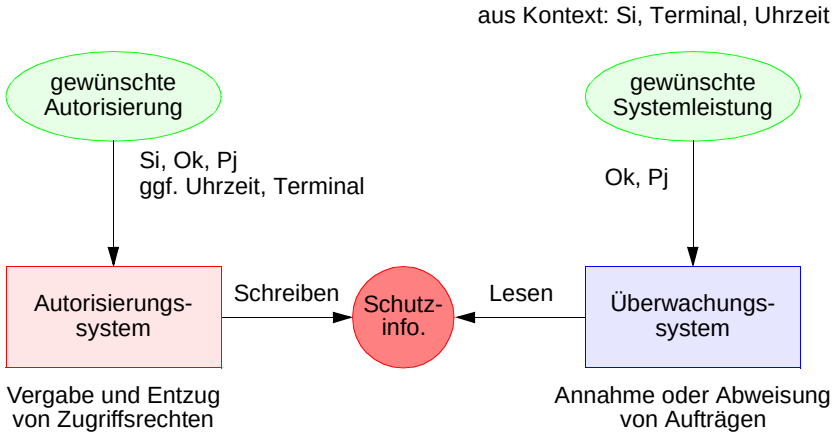
- Zugriffsmatrix ist typischerweise **sehr groß und dünn besetzt**

➥ Welche Realisierungstechniken bieten sich an?

8. Dieses Modell wird im Englischen als Discretionary Access Control (DAC) bezeichnet. Wegen seiner Einfachheit ist DAC weit verbreitet. „discretionary" bedeutet in etwa „nach dem Ermessen des Subjekts".

Autorisierungsmodell (2)

- **Autorisierungs[9]- und Überwachungssystem**



- **Autorisierung**
  - zentrale Vergabe der Zugriffsrechte (DBA)
  - dezentrale Vergabe der Zugriffsrechte durch Eigentümer der Objekte
- **Objektgranulat**
  - wertunabhängige oder
  - wertabhängige Objektfestlegung
- **Wirksamkeit der Zugriffskontrolle beruht auf drei Annahmen:**
  - fehlerfreie Benutzer-Identifikation/-Authentisierung
  - erfolgreiche Abwehr von (unerwarteten) Eindringlingen (vor allem strikte Isolation der Benutzer- und DBS-Prozesse sowie Weitergabekontrolle)
  - Schutzinformation ist **hochgradig geschützt!**

9. Verfeinerungen des Modells gestatten eine implizite Autorisierung von Subjekten, Operationen und Objekten durch Nutzung entsprechender Hierarchien.

Autorisierungsmodell (3)

- **Realisierungstechniken für die Zugriffsmatrix**
  - Zugriffskontroll-Liste: $O_i : (B_j, P_k) , (B_m, P_l), \ldots$
    (in Unix als Bitliste realisiert)
  - Rechte-Liste: $B_j : (O_k, P_m) , (O_n, P_r), \ldots$
    (Capability-Liste)
  - Schlüssel/Schloss-Prinzip: $B_i : (O_k, K_j)$
    $O_k : (P_m, L_n)$

    Wenn $K_j = L_n$
    dann hat B_i Zugriff auf O_k mit P_m
- **Operationen**
  - neue Benutzer, Objekte
  - Vergabe und Entziehen von Rechten
  - Aufruf von Domänen (als Objekte modelliert)
    $\rightarrow$ Enter
  - Rollen in Unix: Eigentümer, Gruppe, Public
- **Autorisierungs-Stack (in SQL)**

Autorisierungs-Stack	
AuthID	**Rollenname**
–	–
Julia	(null)
(null)	AW-Prog.
Daniel	(null)

Stored Procedure
Embedded SQL
Betriebssystem-Login

  - Abbildung von BS- auf DBS-AuthIDs und Rollen
  - Rechtetransfer oder Ausführungserlaubnis (Enter)

Zugriffskontrolle in SQL

- **Sicht-Konzept erlaubt wertabhängigen Zugriffsschutz**
  - Untermengenbildung, Verknüpfung von Relationen, Verwendung von Aggregat-Funktionen
  - Umsetzung durch **Anfragemodifikation** möglich
- **Vergabe von Rechten**

```
GRANT   {privileges-commalist | ALL PRIVILEGES}
        ON accessible-object TO grantee-commalist
        [WITH GRANT OPTION]
```

- **Zugriffsrechte** (privileges)
  - SELECT, INSERT, UPDATE, DELETE, REFERENCES, USAGE, TRIGGER, CONNECT, EXECUTE, . . .
  - Attributeinschränkung bei INSERT, UPDATE und REFERENCES möglich
  - Erzeugung einer „abhängigen“ Relation erfordert REFERENCES-Recht auf von Fremdschlüsseln referenzierten Relationen.
  - USAGE erlaubt Nutzung spezieller Wertebereiche (character sets).
  - dynamische Weitergabe von Zugriffsrechten: WITH GRANT OPTION (GO: dezentrale Autorisierung)
- **Objekte** (accessible-object)
  - Relationen bzw. Sichten
  - aber auch: Domänen, Datentypen, Routinen usw.
- **Empfänger** (grantee)
  - Liste von Benutzern bzw. PUBLIC
  - Liste von Rollennamen

Zugriffskontrolle in SQL (2)

- **Beispiele:**
  - GRANT SELECT ON Abt TO PUBLIC
  - GRANT INSERT, DELETE ON Abt
    TO Mueller, Weber WITH GRANT OPTION
  - GRANT UPDATE (Gehalt) ON Pers TO Schulz
  - GRANT REFERENCES (Pronr) ON Projekt TO PUBLIC

- **Rücknahme von Zugriffsrechten:**

  REVOKE [GRANT OPTION FOR] privileges-commalist
  ON accessible-object FROM grantee-commalist
  {RESTRICT | CASCADE}

  Beispiele: REVOKE DELETE ON Abt FROM Weber CASCADE
  REVOKE GRANT OPTION FOR INSERT ON Abt FROM Mueller

- **Wünschenswerte Entzugssemantik:**
  *Der Entzug eines Rechtes ergibt einen **Schutzzustand**,*
  als wenn das Recht nie erteilt worden wäre.

  ➥ ggf. fortgesetztes Zurücknehmen von Zugriffsrechten

- **Probleme:**
  - Rechteempfang aus verschiedenen Quellen
  - verschiedene Entzugssemantiken:
    - zeitabhängige Interpretation
    - zeitunabhängige Interpretation

  ➥ Führen der Abhängigkeiten in einem *Autorisierungsgraph* erforderlich

Zugriffskontrolle in SQL (3)

- **Autorisierungsgraph mit zeitabhängiger Interpretation:**
  Der Entzug eines Rechtes ergibt einen Schutzzustand,
  als wenn das Recht nie erteilt worden wäre.

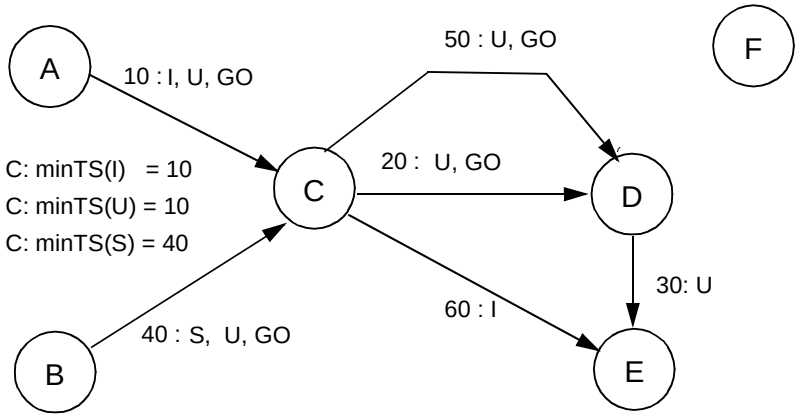
  ➥ Vergabe von Zeitstempeln für jedes Zugriffsrecht bei Autorisierung

- **Beispiel:**

  10 : A: GRANT INSERT, UPDATE ON Pers TO C WITH GRANT OPTION
  20 : C: GRANT UPDATE ON Pers TO D WITH GRANT OPTION
  30 : D: GRANT UPDATE ON Pers TO E
  40 : B: GRANT SELECT, UPDATE ON Pers TO C WITH GRANT OPTION
  50 : C: GRANT UPDATE ON Pers TO D WITH GRANT OPTION
  60 : C: GRANT INSERT ON Pers TO E

Weitere Vergabe:

70 : D: GRANT UPDATE ON Pers TO F



Rücknahme:

80 : A: REVOKE INSERT, UPDATE ON Pers FROM C CASCADE

Zugriffskontrolle in SQL (4)

- **Autorisierungsgraph mit zeitunabhängiger Interpretation:**
  Der rekursive Entzug eines Rechtes wird nicht fortgesetzt, sobald der Geber noch mindestens ein gleiches Recht für das Objekt von einer unabhängigen Quelle hat.

  ➥ Überprüfung der Quellenunabhängigkeit bei jeder Rechtevergabe

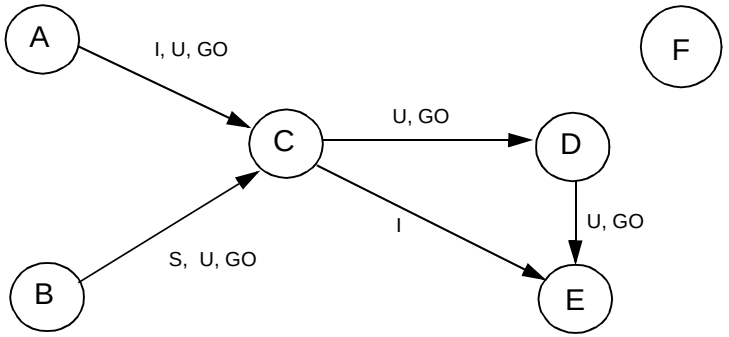
- **Beispiel:**

  A: GRANT INSERT, UPDATE ON Pers TO C WITH GRANT OPTION
  C: GRANT UPDATE ON Pers TO D WITH GRANT OPTION
  D: GRANT UPDATE ON Pers TO E WITH GRANT OPTION
  B: GRANT SELECT, UPDATE ON Pers TO C WITH GRANT OPTION
  C: GRANT UPDATE ON Pers TO D WITH GRANT OPTION
  C: GRANT INSERT ON Pers TO E

  Weitere Vergabe:

  D: GRANT UPDATE ON Pers TO F

  E: GRANT UPDATE ON Pers TO C



  Rücknahme:

  A: REVOKE INSERT, UPDATE ON Pers FROM C CASCADE

  B: REVOKE SELECT, UPDATE ON Pers FROM C CASCADE

Verfeinerungen des Autorisierungsmodells

- **Implizite Autorisierung**
  - hierarchische Anordnung von Subjekten, Objekten und Operationen
  - explizite Autorisierung auf einer Hierarchiestufe bewirkt implizite Autorisierungen auf anderen Hierarchiestufen

- **Negative Autorisierung**
  - stellt ein Verbot des Zugriffs ($\neg p$) dar
  - kann explizit und implizit erfolgen

- **Schwache Autorisierung**
  - kann als Standardeinstellung verwendet werden
    (Leserecht eines Objektes für gesamte Gruppe, die aus Teilgruppen besteht)
  - erlaubt Überschreibung durch starkes Verbot
    (Teilgruppe erhält explizites Leseverbot)
  - Schreibweise: [. . .]

- **Autorisierungsalgorithmus**

  wenn es eine explizite oder implizite **starke** Autorisierung (o, s, p) gibt,
  dann erlaube die Operation

  wenn es eine explizite oder implizite **starke negative** Autorisierung ($o, s, \neg p$) gibt
  dann verbiete die Operation

  ansonsten

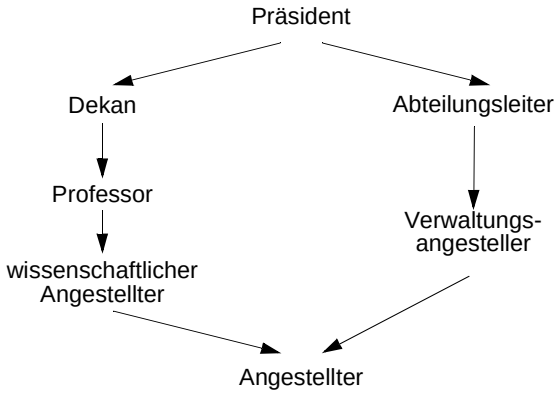
  wenn es eine explizite oder implizite **schwache** Autorisierung [o, s, p] gibt,
  dann erlaube die Operation

  wenn es eine explizite oder implizite **schwache negative** Autorisierung [$o, s, \neg p$] gibt,
  dann verbiete die Operation

  sonst verbiete die Operation

Verfeinerungen des Autorisierungsmodells (2)

- **Implizite Autorisierung von Subjekten**
  - Einführung von Rollenhierarchien
  - zwei ausgezeichnete Positionen
    - eine eindeutige Rolle mit der maximalen Menge an Rechten (z.B. Präsident, Systemadministrator)
    - eine eindeutige grundlegende Rolle (z.B. Angestellter, Hiwi)



- **Explizite positive Autorisierung**

  ➥ implizite positive Autorisierung auf allen höheren Hierarchiestufen

- **Explizite negative Autorisierung**

  ➥ implizite negative Autorisierung auf allen niedrigeren Hierarchiestufen

Rollenkonzept in SQL

- **Rollenkonzept**
  - bisher: (explizite) Zuordnung von Zugriffsrechten zu Benutzern
  - SQL:1999 erlaubt die Definition von Rollen
  - **Ziel:** Vereinfachung der Definition und Verwaltung komplexer Mengen von Zugriffsrechten
    - Erzeugung von Rollen und Vergabe von Zugriffsrechten (Autorisierungen)
    - Kontrolle der Aktivitäten (Einhaltung der vorgegebenen Regeln)

- **Wichtige Rollen**
  - **Systemadministrator**
    - Sie „besitzt“ sämtliche Ressourcen des DBS und ist zur Ausführung einer jeden DB-Anweisung autorisiert.
    - Rolle verwaltet eine DBS-Instanz, die mehrere DBS umfassen kann.
    - Bei DB2/UDB gibt es beispielsweise zwei Untergruppen: Systemkontrolle und Systemwartung.
  - **DB-Administrator**
    - Rolle gilt für eine spezielle DB mit allen Zugriffsrechten.
  - **Anwendungsentwickler**
    - typische Zugriffsrechte: Verbindung zur DB herstellen (CONNECT), Tabellen erzeugen, AWPs an DB binden
    - Zugriffsrechte beziehen sich auf Menge spezieller DB-Objekte.
    - Kapselung von Rechten durch AWP bei statischem SQL
  - **Endbenutzer**
    - Rechte für Ad-hoc-Anfragen
    - CONNECT- und EXECUTE-Rechte für AWPs

Rollenkonzept in SQL (2)

- **Definition von Rollen**

  CREATE ROLE Revisor
  CREATE ROLE Hauptrevisor

  ➥ **keine Hierarchie mit impliziter Vergabe!**

- **Vergabe von Rechten**

  GRANT INSERT ON TABLE Budget TO Revisor

- **Zuweisung von Rollen**

  GRANT role-granted-commalist TO grantee-commalist
  [WITH ADMIN OPTION]

  - Rollen werden Benutzern und Rollen **explizit** zugewiesen.
  - WITH ADMIN OPTION erlaubt die Weitergabe von Rollen.
  - Beispiel:
    GRANT Revisor TO Weber WITH ADMIN OPTION

- **Entzug von Rollen**

  REVOKE [ADMIN OPTION FOR] role-revoked-commalist
  FROM grantee-commalist
  {RESTRICT | CASCADE}

  - Beispiele:
    REVOKE Revisor FROM Weber RESTRICT
    REVOKE ADMIN OPTION FOR Revisor FROM Weber CASCADE

  - WITH ADMIN OPTION ist „vorsichtig“ einzusetzen

Rollenkonzept in SQL (3)

- **Anwendung**

  - Momentaner Rechtebesitz

Revisor	:	P1, P2, P5
Hauptrevisor	:	P3, P4
Benutzer Schmidt	:	P1

  - Zuweisung von Rollen:

    GRANT Revisor TO Hauptrevisor WITH ADMIN OPTION

    Hauptrevisor:

  - „A role can contain other roles“!

    GRANT Hauptrevisor TO Schmidt

    Schmidt:

  - Evolution von Rollen:

    Grant P6 ON TABLE X TO Revisor

    ➥ **Wer bekommt aktuell P6?**

  - Entzug von Rollen:

    Revoke Revisor FROM Hauptrevisor RESTRICT

    Revoke Revisor FROM Hauptrevisor CASCADE

    Revisor:

    Hauptrevisor:

    Schmidt:

    ➥ **Implementierung der Rollenvergabe erfolgt sinnvollerweise referenziert und nicht materialisiert!**

Verfeinerungen des Autorisierungsmodells (3)

- **Implizite Autorisierung von Operationen**
  - Festlegung von Operationshierarchien

Erzeugen/Löschen
↓
Schreiben
↓
Lesen

- **Explizite positive Autorisierung**

➥ implizite positive Autorisierung auf allen niedrigeren Ebenen

- **Explizite negative Autorisierung**

➥ implizite negative Autorisierung auf allen höheren Ebenen

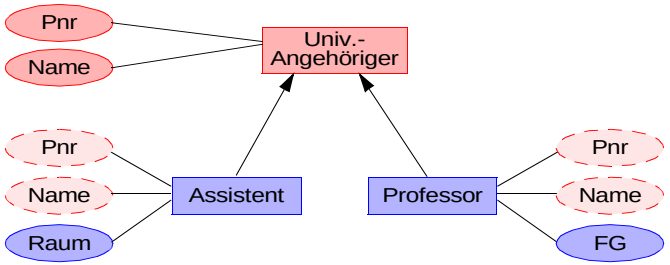
- **Implizite Autorisierung von Objekten**
  - Festlegung von Granulatshierarchien

Datenbank
↓
Relationenschema
↓
Relation
↓
Tupel
↓
Attribut

- **Implikationen sind abhängig von Operationen**
  - Explizites Erlaubnis zum Lesen einer Relation impliziert das Recht, das Schema der Relation zu lesen
  - Explizite Erlaubnis zum Lesen eines Objektes impliziert automatisch, alle Objekte feinerer Granularität zu lesen
  - Definition einer Relation hat keine Implikationen

Verfeinerungen des Autorisierungsmodells (4)

- **Implizite Autorisierung in Typhierarchien**
  - weitere Dimension der Autorisierung durch Generalisierung/Spezialisierung



- **Benutzergruppen**
  - Verwaltungsangestellte (V) dürfen Namen aller Univ.-Angehörigen lesen
  - Wiss. Angestellte (W) dürfen Namen, Fachgebiet (FG) aller Profs lesen
- **Anfragen**
  - Q1: Lese die Namen aller Univ.-Angehörigen
  - Q2: Lese Namen und Fachgebiet aller Professoren

- **Drei Grundregeln**
  - Ein Zugriffsrecht auf einen Objekttyp impliziert ein gleichartiges Zugriffsrecht auf die vererbten Attribute in den Subtypen
  - Ein Zugriffsrecht auf einen Objekttyp impliziert auch ein gleichartiges Zugriffsrecht auf alle ererbten Attribute
  - Ein in einem Objekttyp definiertes Attribut ist mit den Zugriffs-berechtigungen auf seinem Supertyp nicht zugreifbar

Verfeinerungen des Autorisierungsmodells (5)

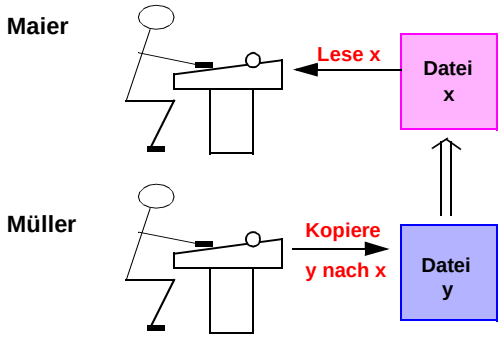
- **Hierarchische Objekt-/Subjektklassifikation nach Sicherheitsstufen**
  - Sicherheitshierarchie: streng geheim, geheim, vertraulich, öffentlich
  - hierarchische Klassifikation von Vertrauenswürdigkeit (bei Subjekten) und Sensitivität (bei Objekten)
    - clear (s), mit s als Subjekt (clearance)
    - class (o), mit o als Objekt (classification)

- **Mandatory Access Control** (Bell-LaPadula-Modell):
  - MAC-Modell realisiert hierarchische Objekt-/Subjektklassifikation mit zwei Zugriffsregeln
    - Ein Subjekt s darf ein Objekt o nur lesen, wenn class (o) ≤ clear (s)
    - Ein Objekt o muss mit mindestens der Einstufung des Subjektes s geschrieben werden: clear (s) = class (o) (oder clear (s) ≤ class (o))

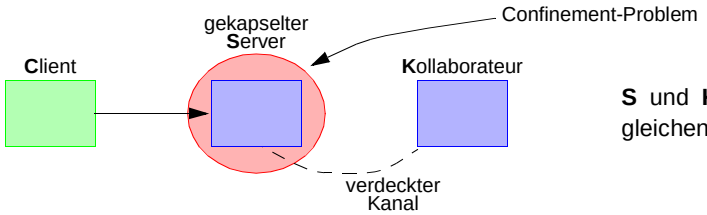
    ➥ kein „write down“ möglich
  - bietet potentiell größere Sicherheit, aber Benutzer unterschiedlicher Sicherheitsstufen können nur schwer zusammenarbeiten

- **Sicherheitshierarchie**

Datenflusskontrolle



- **Verdeckte Kanäle (covert channels)**



S und **K** haben den gleichen Eigentümer

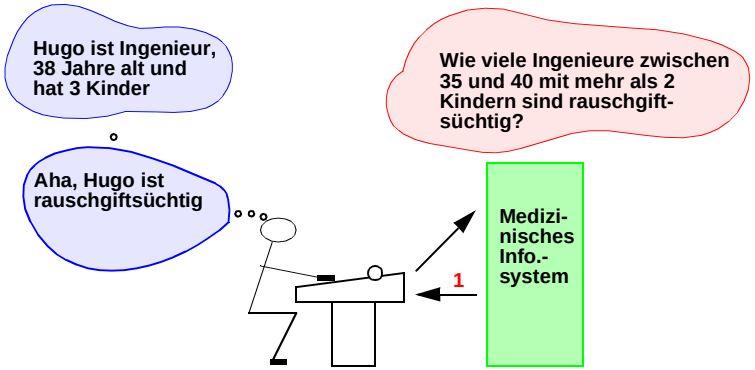
- **Beispiel**
  - Steuerabrechnung von C durch S
  - S darf nichts in Datei aufzeichnen oder an anderen Prozess schicken

- **Subtile Kommunikationskanäle: Bitströme auf Zeitraster abbilden**

	‘O’	‘1’	
S	rechnet	idle	**K** beobachtet seine Antwortzeit
S	produziert viele Seitenfehler	keine	**K** beobachtet Systemleistung
S	sperrt Datei (Band, Plotter)	gibt Datei frei (Band, Plotter)	**K** fragt Sperrzustand ab

S teilt **K** Abrechnungsdaten für **C** mit: bei 95 K $ eine Abrechnungssumme von 100.95 $

Inferenzkontrolle



- **Datenschutzforderung:**
  Zu Forschungszwecken sind personenbezogene Daten zu anonymisieren.
  Es ist nur der Einsatz statistischer Funktionen erlaubt wie
  AVG, MIN, MAX, COUNT, ...

  ➥ Einzelwerte dennoch oft ableitbar!

Sicherheitsprobleme in statistischen DB

- **Ableitungsbedingungen**
  - selektiven Anfragen (kleine Treffermengen)
  - Ergebnisverknüpfung mehrerer Anfragen

- **Beispiel:**
  - statistische Anfragen auf Pers ohne Attribute Pnr und Name
  - Wissen über bestimmte Personen (z. B. Alter, Beruf, Familienstand, Firmenzugehörigkeit) kann leicht für gezielte Anfragen genutzt werden.

```
SELECT COUNT (*)
FROM Pers
WHERE Alter = 51 AND Beruf = 'Operateur'

SELECT AVG (Gehalt)
FROM Pers
WHERE Alter = 51 AND Beruf = 'Operateur'
```

  ➥ Bei mehr als einem Treffer kann Treffermenge durch weitere Bedingungen reduziert werden.

  ➥ Eine leere Treffermenge enthält auch Information!

- **Abhilfemöglichkeiten:**
  - Antwortausgabe nur, wenn Treffermenge über festgelegtem Grenzwert liegt
  - Überprüfung, ob mehrere Anfragen aufeinander aufbauen
  - gezielte Einstreuung von kleineren Ungenauigkeiten

Sicherheitsprobleme in statistischen DB (2)

- **Bsp: N = 13, M = 5**

Name	Geschlecht	FB	Beginn	Abi-Note	BA-Note
Abel	W	Inf	2004	1.6	1.5
Bebel	W	Etech	2003	2.7	2.2
Cebel	M	Etech	2001	1.5	1.3
Damm	W	Inf	2004	1.0	1.0
Ehrlich	M	Bio	2002	2.8	2.6
Fuchs	M	Etech	1999	2.5	1.8
Grommel	M	Inf	2000	1.3	1.2
Heinrich	W	Chem	2004	2.5	2.0
Ibsen	M	Inf	2003	1.6	1.6
Jahn	W	Bio	2004	1.3	1.2
Kramer	W	Math	2004	2.8	2.2
Lustig	M	Etech	2002	1.6	1.8
Müller	M	Inf	1995	1.4	1.3

- **Statistische DB**

#Werte:	Geschlecht	:	2
	FB	:	10
	Abi-Note	:	31

Sicherheitsprobleme in statistischen DB (3)

- **Zuordnung von anonymisierten Daten**
  - Voraussetzung:
    - B kennt I
    - Daten von I sind in SDB repräsentiert und erfüllen C
  - Gesucht: Eigenschaft D von C

- **Charakteristische Formel C:**
  C = (Geschlecht='W' $\wedge$ FB='Etech'), kurz: (W $\wedge$ Etech)

  1. COUNT (C) =

  2. SUM (C, BA-Note) =

  oder

  1. COUNT (C $\wedge$ BA-Note = 2.1) =

  2. COUNT (C $\wedge$ BA-Note = 2.2) =
     . . .

- **Forderung**

  COUNT (C) > 1

  SUM (C, BA-Note) verboten!
  $\underbrace{\quad}_{1}$

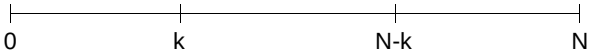
  SUM (Etech, BA-Note) - SUM (Etech $\wedge$ M, BA-Note)

  $$COUNT(C) = N - COUNT(\neg C)$$

Sicherheitsprobleme in statistischen DB (4)

- **Forderung: Anfrage nur erlaubt, falls**

  $k \leq |C| = \text{Count}(C) \leq N\text{-}k, \qquad 1 < k < N/2$



- **Einsatz von Trackern**

  - Vorgehensweise:
    Antwortmenge wird mit zusätzlichen Sätzen aufgebläht, deren Beitrag zu den Statistiken anschließend wieder herausgefiltert wird

  - Gesucht: Eigenschaft D von C

    $\text{Count}(C \wedge D) \leq \text{Count}(C) < k$ verboten!

  - Zerlegung

    $C = C_1 \wedge C_2$

    mit

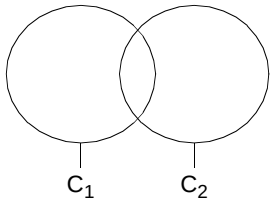
    $k \leq \text{Count}(C_1 \wedge \neg C_2) \leq \text{Count}(C_1) \leq N\text{-}k$

  - **$T = (C_1 \wedge \neg C_2)$ heißt individueller Tracker von I**

Sicherheitsprobleme in statistischen DB (5)

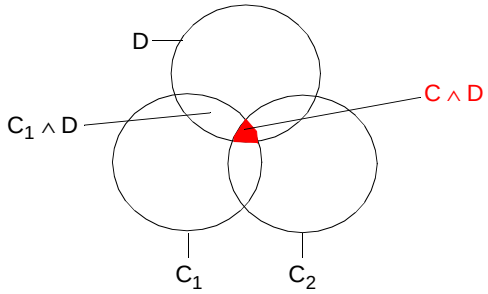
- **Angriff mit Trackern**

  $\text{Count}(C) = \text{Count}(C_1 \wedge C_2) = \text{Count}(C_1) - \text{Count}(C_1 \wedge \neg C_2)$



  - **Berechne**

    $\text{Count}(C \wedge D)$ $= \text{Count}(T \vee C_1 \wedge D) - \text{Count}(T)$



also:

$\text{Count}(C \wedge D) = 0$	$\rightarrow$	I hat nicht Eigenschaft D
$\text{Count}(C \wedge D) = \text{Count}(C)$	$\rightarrow$	I hat Eigenschaft D
$\text{Count}(C) = 1$	$\rightarrow$	Attributwert von A berechenbar über $\text{Sum}(C, A) = \text{Sum}(C_1, A) - \text{Sum}(T, A)$

Sicherheitsprobleme in statistischen DB (6)

- **Allgemeiner Tracker**

  ist jede charakteristische Formel T mit

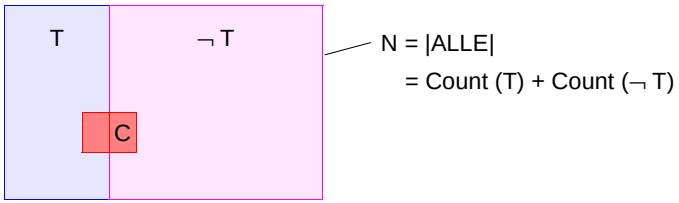
  $$k \leq \text{Count}(T) \leq N - k \qquad \text{mit } k < N/2$$

  - aus Symmetriegründen: $\neg T$ ist auch ein Tracker

- **Angriff mit allgemeinem Tracker**

  - Enthüllung für Count (C) < k

  $$\textbf{Count (C)} = \text{Count}(C \vee T) + \text{Count}(C \vee \neg T) - N$$



N = |ALLE|
= Count (T) + Count (¬ T)

  - also:

  $$\text{Count}(C) = \text{Count}(C \vee T) + \text{Count}(C \vee \neg T) - \text{Count}(T) - \text{Count}(\neg T)$$

  Beispiel: $C = W \wedge Etech$, $T = W$

  $$\text{Count}(C) = \text{Count}(W \wedge Etech \vee W) + \text{Count}(W \wedge Etech \vee \neg W)$$

  $$- \text{Count}(W) - \text{Count}(\neg W) =$$

Zusammenfassung

- **BDSG regelt die Verarbeitung personenbezogener Daten**
  - **Verbotsprinzip mit Erlaubnisvorbehalt**
  - Technische Maßnahmen (urspr. die sog. Zehn Gebote) sind stets den veränderten Randbedingungen der IT anzupassen und neu zu interpretieren

- **Aufeinander abgestimmte Sicherheitskonzepte sind wesentlich**
  - Zugangskontrolle
  - starke Verfahren zur Authentisierung
  - kryptographische Maßnahmen zur Datenübertragung
  - Isolation der Prozesse
  - Prinzip der Zugriffskontrolle: **Least Privilege Principle**
  - Sicherungsanforderungen gelten allgemein in Rechensystemen und insbesondere zwischen Anwendung und DBS.

➥ Das „schwächste Glied“ in der Kette der Sicherheitsmaßnahmen bestimmt die Sicherheit des Gesamtsystems!

- **Zugriffskontrolle in DBS**
  - wertabhängige Festlegung der Objekte (Sichtkonzept)
  - Vielfalt an Rechten erwünscht
  - zentrale vs. dezentrale Rechtevergabe
  - verschiedene Entzugssemantiken bei dezentraler Rechtevergabe
  - **Rollenkonzept:** vereinfachte Verwaltung komplexer Mengen von Zugriffsrechten

- **Sicherheitsprobleme**
  - Datenflusskontrolle und Inferenzkontrolle
  - Wenn Zusatzwissen vorhanden ist, lassen statistische DBs die Individualisierung von anonymisierten Daten zu.
  - **Allgemeine Tracker** sind „leicht“ anzuwenden!

Kryptographie: Basistechnologie der Informationsgesellschaft

- **Informationssicherheit und Kryptographie**
  - Paradigmenwechsel: Information wird zur bestimmenden Ressource von Wirtschaft und Gesellschaft
  - Digitalisierung der meisten Geschäfts- und Verwaltungsprozesse erfordern neue Schutzvorkehrungen: Sicherstellung der
    - Vertraulichkeit
    - Authentizität
    - Nichtkopierbarkeit von Information
    - Verbindlichkeit von Transaktionen/Verträgen
    - Unfälschbarkeit von digitalem Geld
  - Stark zunehmende Aktualität der Informationssicherheit
    - steigende Abhängigkeit der Unternehmen von Informationssystemen
    - rasante Erhöhung des Gefahrens- und Schadenpotenzials
    - immer komplexere Anforderungen an den Datenschutz

➥ Kryptographie ist eine der grundlegenden Technologien der Informationssicherheit

- **Politische und gesellschaftliche Aspekte**
  - Verschlüsselungsverfahren
    - Nutzen: Datenschutz
    - Risiken: Tarnung krimineller Aktivitäten
  - Benutzung, Import und Export kryptographischer Verfahren ist nicht kontrollierbar!
    - Hinterlegung von Schlüsseln (bei einer staatlichen Behörde) wirkungslos
    - andere Möglichkeiten: Steganographie

➥ Schlussfolgerung: sichere Kommunikation ist für jedermann (unkontrollierbar) möglich!

Kryptographie: Basistechnologie der Informationsgesellschaft (2)

- **Politische und gesellschaftliche Aspekte (Forts.)**
  - „Knacken“ kryptographischer Verfahren
    - Brute-Force-Attacken: Aufwand ~ f (Schlüssellänge)
    - Was würde passieren, wenn ein schneller Faktorisierungsalgorithmus gefunden wird, der das allgemein benutzte RSA-Verfahren bricht?

➥ Alle bisher geleisteten digitalen Signaturen wären fälschbar und somit ungültig!

  - Was sind Konsequenzen dieses Szenarios, wenn die gesamte Weltwirtschaft in einigen Jahren auf einem bestimmten kryptographischen System basiert?

➥ Erkennbare Entwicklung (trotz der Risiken): Aufbau einer globalen Public-Key-Infrastruktur

- **Anwendungen der Kryptographie**
  - nicht nur bloße „Verschlüsselungstechnik
  - digitale Signaturen
  - Verfahren für das Schlüssel-Management
  - Zero-Knowledge-Identifikationsverfahren
  - digitale Zahlungssysteme (anonymes digitales Geld)
  - Vielfalt von kryptographischen Protokollen für die sichere Kooperation mehrerer sich nicht trauender Partner

- **Beispiel: sichere Wahlen und Abstimmungen über das Internet**
  - nicht nur Schutz der Datenübertragung
  - Auswertungsmechanismus muss die Stimmen zählen,
    - ohne sie wirklich zu kennen (Datenschutz) und
    - auf eine öffentlich nachvollziehbare korrekte Art

➥ Auswertung muss z. B. durch mehrere unabhängige Systeme vorgenommen werden. Dabei ist die Vertraulichkeit und Korrektheit garantiert, selbst wenn ein beliebiger Teil (z.B. bis zur Hälfte) aller Teilsysteme durch einen Betrüger kontrolliert und manipuliert würde