# Algorithmen und Datenstrukturen (für ET/IT)

## Sommersemester 2014

Dr. Tobias Lasser

Computer Aided Medical Procedures
Technische Universität München


# Programm heute

7. Fortgeschrittene Datenstrukturen
8. Such-Algorithmen
9. Graph-Algorithmen
10. Numerische Algorithmen
11. Datenkompression
12. Kryptographie
    - Einführung
    - Verfahren mit privaten Schlüsseln
    - Verfahren mit öffentlichen Schlüsseln

# Motivation

## Zitat von Phil Zimmermann (PGP Autor)

"Cryptography used to be an obscure science, of little relevance to everyday life. Historically, it always had a special role in the military and diplomatic communications.
But in the Information Age, cryptography is about political power and in particular, about the power relationship between a government and its people. It is about the right to privacy, freedom of speech, freedom of political association, freedom of the press, freedom from unreasonable search and seizure, freedom to be left alone."

# Kryptologie

- Kryptologie: Wissenschaft der geheimen Kommunikation
  - Kryptographie: Verschlüsselung von Informationen
  - Kryptanalyse: Knacken von Verschlüsselung

- Ziel: Informations-Sicherheit
  - Geheimhaltung: Kommunikation ist privat
  - Integrität: unautorisierte Änderungen entdecken
  - Authentifizierung: Identität des Senders bestätigen
  - Autorisierung: Zugriffsrechte feststellen
  - Nachweisbarkeit: Erhalt der Nachricht beweisen

# Wozu Informations-Sicherheit?

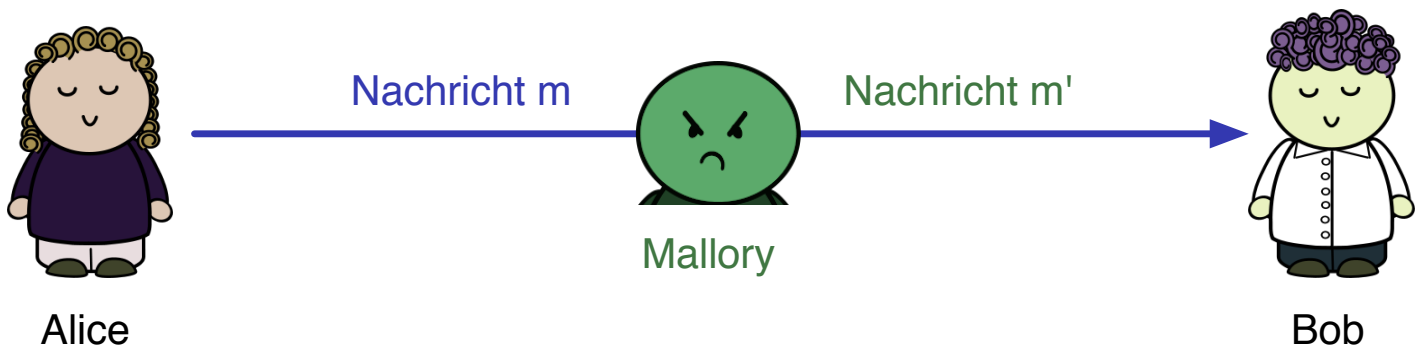
- Sicherheit in der "analogen" Welt
  - Schlösser und Schlüssel
  - Unterschriften, notarielle Beglaubigung
  - Ausweis, Fingerabdruck, DNA-Informationen
  - Geldscheine, EC-Karten, Kreditkarten
  - anonyme Wahlzettel
  - ...
- Sicherheit in der digitalen Welt?
  - Accounts und Passwörter
  - verschlüsselte Emails, Dokumente
  - elektronische Unterschrift, Signatur, Zertifikate
  - eCommerce, eBanking
  - ...

# Verschlüsselung

- Verschlüsselung - Basisproblem der Kryptographie
  - Alice will Bob eine private Nachricht m senden

# Verschlüsselung

- Verschlüsselung - Basisproblem der Kryptographie
  - Alice will Bob eine private Nachricht m senden

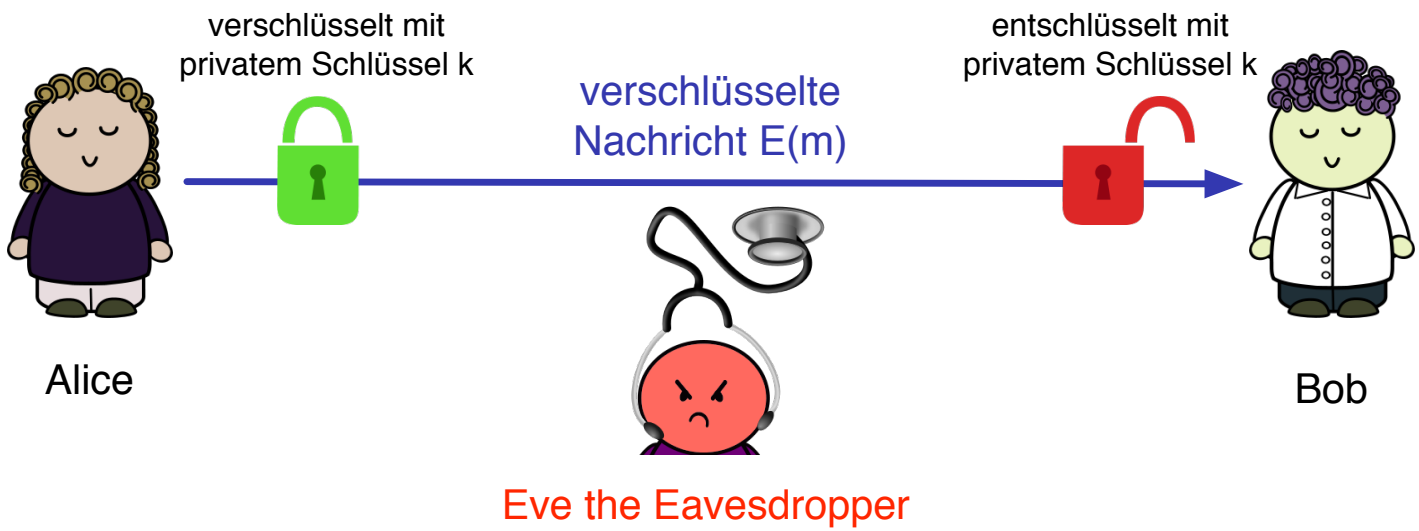


# Programm heute

7. Fortgeschrittene Datenstrukturen
8. Such-Algorithmen
9. Graph-Algorithmen
10. Numerische Algorithmen
11. Datenkompression
12. Kryptographie
    - Einführung
    - Verfahren mit privaten Schlüsseln
    - Verfahren mit öffentlichen Schlüsseln

# Verschlüsselung mit privaten Schlüsseln

- Nachricht wird mit privatem Schlüssel k verschlüsselt
- Privater Schlüssel muss Alice und Bob bekannt sein



# Symmetrische Verschlüsselung: One Time Pads

- Schlüsselverteilung: Alice und Bob müssen Schlüssel k (Länge n Bit) austauschen
  - Schlüssel k wird nur einmal verwendet! (One Time Pad)

- Verschlüsselung: Alice verschlüsselt Nachricht m (Länge n Bit) mit mit Schlüssel k via bit-weisem XOR,

$$E(m) = m \text{ XOR } k$$

- Entschlüsselung: Bob entschlüsselt codierte Nachricht c (Länge n Bit) mit Schlüssel k via bit-weisem XOR,

$$D(c) = c \text{ XOR } k$$

- Funktionsweise:

$$D(E(m)) = D(m \text{ XOR } k) = (m \text{ XOR } k) \text{ XOR } k = m$$

# One Time Pads: Eigenschaften

Vorteile:

- beweisbar sicher, vorausgesetzt Schlüssel S ist zufällig
- sehr einfaches Verfahren

Nachteile:

- zufällige Schlüssel sind schwierig zu generieren
- jede Nachricht benötigt einen neuen Schlüssel
- sicherer Schlüsselaustausch ist sehr problematisch
  - größtes Hindernis in der Praxis!
- ungeeignet zur Authentifizierung und Nachweisbarkeit

# Verschlüsselungs-Standards mit privaten Schlüsseln

- DES - Data Encryption Standard
  - Schlüssellänge 56 Bit
  - Algorithmus basiert auf Permutationen, Substitutionen, XOR
  - Standard der US Regierung von 1976
  - entwickelt von IBM in Kooperation mit NSA
  - Einsatz bei alten Geldautomaten, Polizeifunk
  - kann seit 2008 in weniger als einem Tag geknackt werden
    - basiert auf einer FPGA-Lösung, entwickelt an den Universitäten Bochum und Kiel

- 3DES - Triple DES
  - dreifache Anwendung von DES mit drei verschiedenen Schlüsseln
  - Schlüssellänge 168 Bit
  - durch NIST als effektiv 112 Bit bewertet
  - von NIST bis 2030 als sicher eingestuft
  - Einsatz bei Geldautomaten, EC Karten

# Verschlüsselungs-Standards mit privaten Schlüsseln

- AES - Advanced Encryption Standard
  - Schlüssellänge 128, 192 oder 256 Bit
  - Algorithmus basiert auf Permutationen, Substitutionen, XOR
  - keine praktischen Angriffe öffentlich bekannt
  - NSA empfiehlt 256 Bit Schlüssellänge für top secret Nachrichten
    - neues NSA Daten-Center in Utah soll alle verschlüsselten Daten speichern bis NSA AES-256 geknackt hat
      `http://nsa.gov1.info/utah-data-center/`
  - Einsatz bei WPA, SSH, IPsec, Disk-Verschlüsselung Windows/MacOS X, 7zip, PGP, ...
  - Hardware-Unterstützung z.B. in Intel/AMD Prozessoren

- zahlreiche andere Standards (Blowfish, RC4, IDEA, ...)

# Probleme mit privaten Schlüsseln

## Generelle Probleme:

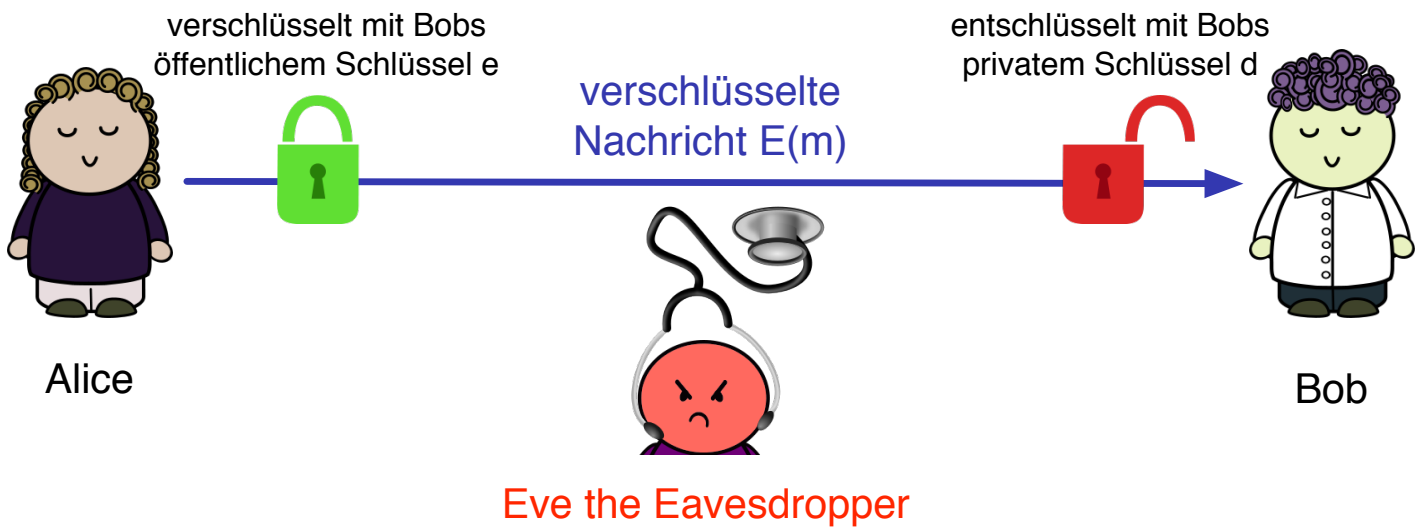
- Schlüssel-Austausch
- ungeeignet für Authentifizierung / Signatur
- ungeeignet für Nachweisbarkeit

# Programm heute

7. Fortgeschrittene Datenstrukturen
8. Such-Algorithmen
9. Graph-Algorithmen
10. Numerische Algorithmen
11. Datenkompression
12. Kryptographie
    - Einführung
    - Verfahren mit privaten Schlüsseln
    - Verfahren mit öffentlichen Schlüsseln

# Verschlüsselung mit öffentlichen Schlüsseln

- Nachricht wird mit Bobs öffentlichem Schlüssel e verschlüsselt
- Nachricht wird mit Bobs privatem Schlüssel d entschlüsselt



# Verschlüsselung mit öffentlichen Schlüsseln

Schlüsselverteilung:

- öffentlicher Schlüssel e von Bob in "digitalem Telefonbuch"
- privater Schlüssel d nur Bob bekannt

Kommunikation:

- Verschlüsselung von Nachricht m mit e: E(m)
- Entschlüsselung von codierter Nachricht c mit d: D(c)

Annahmen:

- $D(E(m)) = m$
- Verschlüsselung und Entschlüsselung effizient
- Entschlüsselung nicht mit öffentlichem Schlüssel e möglich!

# RSA Verfahren

- RSA: asymmetrisches Verschlüsselungsverfahren
  - 1977 entwickelt von Rivest, Shamir, Adleman
  - basiert darauf, daß Primfaktorzerlegung von großen Zahlen (derzeit) nicht effizient implementierbar ist

- universeller Einsatz, zum Beispiel
  - Protokolle: IPSec, TLS, SSH
  - Email-Verschlüsselung: PGP, S/MIME
  - eBanking: HBCI
  - ePass
  - ...

# RSA: Schlüsselerzeugung

Schlüsselerzeugung:

- wähle zufällig zwei große Primzahlen $p$ und $q$
- berechne $N = pq$

**Zahlentheorie**

Sind $p$, $q$ zwei Primzahlen, dann gibt es effizient zu berechnende Ganzzahlen $e$ und $d$, so daß für alle Ganzzahlen m gilt:

$$(m^e)^d = m \pmod{N}.$$

- öffentlicher Schlüssel: $(e, N)$
- privater Schlüssel: $(d, N)$

# RSA: Beispiel Schlüssel

- wähle $p = 11$, $q = 29 \quad \Rightarrow \quad N = 319$.
- dann ist:

$$(m^3)^{187} = m \pmod{319}$$

- öffentlicher Schlüssel: $(3, 319)$
- privater Schlüssel: $(187, 319)$

# RSA angewendet

## Verschlüsselung:

- Alice holt öffentlichen Schlüssel von Bob $(e, N)$
- Alice berechnet $E(m) = m^e \pmod{N}$

## Entschlüsselung:

- Bob erhält verschlüsselte Nachricht $c$ von Alice
- Bob kennt seinen privaten Schlüssel $(d, N)$
- Bob berechnet $D(c) = c^d \pmod{N}$

## Funktionsweise:

$$\begin{aligned} D(E(m)) &= D(m^e) \pmod{N} \\ &= (m^e)^d \pmod{N} \\ &= m \pmod{N} \end{aligned}$$

# RSA: Beispiel fortgesetzt

- wähle $p = 11$, $q = 29 \quad \Rightarrow \quad N = 319$.
- öffentlicher Schlüssel: $(3, 319)$
- privater Schlüssel: $(187, 319)$

- Alice schickt Nachricht $m = 100$ verschlüsselt:

$$E(m) = 100^3 \pmod{319} = 254$$

- Bob entschlüsselt codierte Nachricht $c = 254$:

$$D(c) = 254^{187} \pmod{319} = 100$$

# Modulare Exponentation

$$c = a^b \pmod{N}$$

Naiver Algorithmus:

- multipliziere $a$ $b$-mal mit sich selbst, dann rechne modulo $N$

Komplexität:

- angenommen $a, b, N$ haben $n$ Bits
- Anzahl Multiplikationen: $O(2^n)$
- Anzahl Ziffern in Zwischenergebnis: $O(2^n)$

# Effiziente modulare Exponentation

- berechne  mod $N$ nach jeder Multiplikation
  - Zwischenergebnisse haben maximal $2n$ Bits
- wiederholtes Quadrieren:

| Term | berechne | mod 319 |
|---|---|---|
| $254^1$ | $254^1$ | 254 |
| $254^2$ | $254^2$ | 78 |
| $254^4$ | $78^2$ | 23 |
| $254^8$ | $23^2$ | 210 |
| $254^{16}$ | $210^2$ | 78 |
| $254^{32}$ | $78^2$ | 23 |
| $254^{64}$ | $23^2$ | 210 |
| $254^{128}$ | $210^2$ | 78 |

$$
\begin{aligned}
254^{187} \pmod{319} &= 254^{59} \cdot 254^{128} \pmod{319} \\
&= 254^{59} \cdot 78 \pmod{319} \\
&= 78 \cdot 254^{27} \cdot 254^{32} \pmod{319} \\
&= 78 \cdot 254^{27} \cdot 23 \pmod{319} \\
&= 199 \cdot 254^{11} \cdot 254^{16} \pmod{319} \\
&= 199 \cdot 254^{11} \cdot 78 \pmod{319} \\
&= 210 \cdot 254^{3} \cdot 254^{8} \pmod{319} \\
&= 210 \cdot 254^{3} \cdot 210 \pmod{319} \\
&= 78 \cdot 254 \cdot 254^{2} \pmod{319} \\
&= 34 \cdot 78 \pmod{319} \\
&= 100
\end{aligned}
$$

- maximal $2n$ Multiplikationen und Modulo Operationen

# RSA Details

Größe von $N = pq$:

- $N$ sollte 2048 Bits oder mehr haben
- zu wenig: leicht zu knacken
- zu viel: längere Berechnungen zum ent- bzw. verschlüsseln

Wahl von $p$, $q$:

- erzeuge große Zufallszahlen, teste auf Primalität
- es gibt genug Primzahlen
  - zum Beispiel $10^{151}$ mit bis zu 512 Bit

# RSA Angriffspunkte

- Primfaktorzerlegung: Annahme, daß $N$ sich nicht in realistischer Zeit in die Primfaktoren $p, q$ zerlegen lässt
  - falls irgendwann möglich, lässt sich aus $p, q$ und öffentlichem Schlüssel $e$ der private Schlüssel $d$ einfach errechnen (Euklidscher Algorithmus)

- Semantische Sicherheit: wenn man weiss, daß Alice entweder ATTACKE oder RÜCKZUG senden wird, verschlüssele beides mit Bobs öffentlichem Schlüssel, um zu sehen, was Alice gesendet hat

- Gleicher Modulus: benutzt Bob $(d_1, e_1, N)$ und Ben $(d_2, e_2, N)$, dann kann Bob $d_2$ berechnen mit $e_2$, und Ben $d_1$ mit $e_1$

# RSA in der Praxis

Vorteile:

- Problem der Schlüsselverteilung gelöst
- lässt sich auf digitale Signaturen, Zertifikate etc. erweitern

Nachteile:

- Sicherheit letztendlich ungeklärt
- Entschlüsselung teurer als bei symmetrischen Verfahren

Praxis: hybrides System

- AES zur symmetrischen Verschlüsselung
- RSA zum Schlüsselaustausch für AES

# Zusammenfassung

Gutes Gelingen!