Sicherheit im Internet mit Public-Key-Kryptographie

Prof. Dr. Michael Helbig

Agenda

- Kommunikation im Internet mit HTTP
- Public-Key-Kryptographie – Allgemein
- Public-Key-Kryptographie – RSA
- Sicherheit von RSA

Agenda

- Kommunikation im Internet mit HTTP
- Public-Key-Kryptographie – Allgemein
- Public-Key-Kryptographie – RSA
- Sicherheit von RSA

Kommunikation mit HTTP

Request: URL=http://www.fh-rosenheim.de
evtl. mit Daten

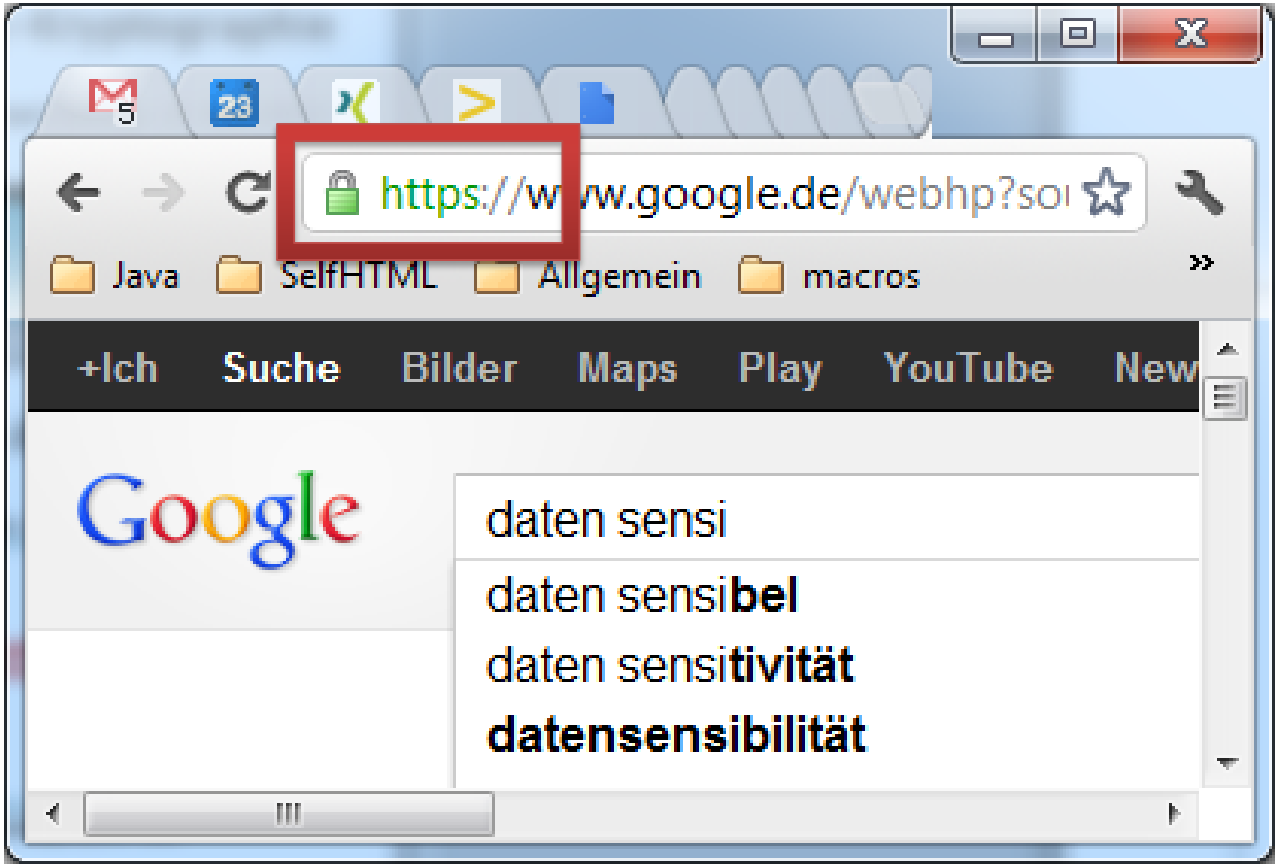
Response: Übertragung einer Seite/Daten

Sensible Daten

- Bestellung Online-Handel
- Online-Banking
- …

- **Lösung:** Verschlüsselung → HTTP Secure

HTTPS = HTTP Secure



Was steckt hinter HTTPS?

- Informatik:
  - Zertifikate
  - Software
  - …
- Mathematik:
  - Public-Key-Kryptographie (PKC)

Agenda

- Kommunikation im Internet mit HTTP
- Public-Key-Kryptographie – Allgemein
- Public-Key-Kryptographie – RSA
- Sicherheit von RSA

Public-Key? - Problemstellung

- Schlüsselaustausch im Internet

Verschlüsselt Daten mit Schlüssel

Schlüssel zum Entschlüsseln?

Public-Key? - Lösung

- Empfänger: **öffentlicher** & **privater** Schlüssel

Verschlüsselt Daten mit
öffentlichem Schlüssel

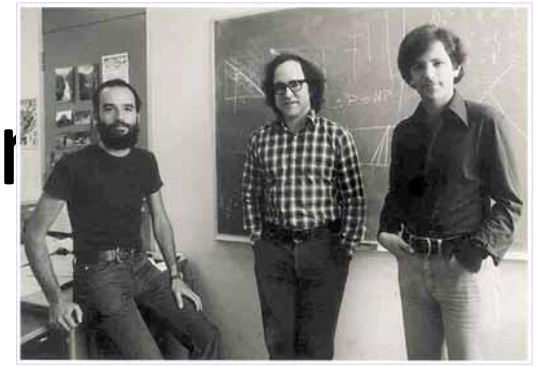
Entschlüsselt mit
privatem Schlüssel

- **öffentlicher** Schlüssel
- **privater** Schlüssel

Agenda

- Kommunikation im Internet mit HTTP
- Public-Key-Kryptographie – Allgemein
- Public-Key-Kryptographie – RSA
- Sicherheit von RSA

PKC r



RSA SECURITY™

- 1977: Rivest, Shamir, Adleman
- am weitesten verbreitete PKC-Methode
- Kongruenz/Modulo-Rechnung:
  - Def: Für $a,b \in \mathbb{Z}$: $a \equiv b \bmod N \;:\Leftrightarrow\; a=b+kN \quad k \in \mathbb{Z}$
  - Bsp „Uhrzeit“:
    - $14 \equiv 2 \bmod 12$, da $14=2+1\cdot12$
    - $25 \equiv 1 \bmod 12$, da $25=1+2\cdot12$

RSA

- Vorbereitung: Wähle
  - Primzahlen p, q
  - RSA-Modul $N \coloneqq pq$
  - $e, d \in \mathbb{Z}$, so dass $ed \equiv 1 \, mod \, (p-1)(q-1)$

- Public Key: (N, e)
- Private Key: (N, d)

- Verschlüsselung: $M \longmapsto C \equiv M^e \, mod \, N$
- Entschlüsselung: $C \longmapsto M \equiv C^d \, mod \, N$

Agenda

- Kommunikation im Internet mit HTTP
- Public-Key-Kryptographie – Allgemein
- Public-Key-Kryptographie – RSA
- Sicherheit von RSA

Sicherheit von RSA

- beruht auf „Faktorisierungsproblem“

$$N = p \cdot q$$

- Ist das sicher?

„Teste Teilbarkeit durch
alle kleineren Primzahlen!“

„Naiver Test“: RSA 128 bit

- $\mathrm{N} \approx 2^{128} \approx 3{,}4 \cdot 10^{38} \approx$

  340.282.366.920.938.463.463.374.607.431.770.000.000

- Test: Für alle Primzahlen $1 < p < N$: p teilt N?

- Wie viele solche Primzahlen gibt es? ca.

$$\frac{N}{\ln N} \approx 3{,}84 \cdot 10^{36} \approx$$

  3.835.341.275.459.348.169.893.510.517.860.100.000

- Supercomputer: prüfe 10^{12} Primzahlen pro Sekunde

Dauer „naiver Test“

121.617.874.031.562.000 Jahre

- Vergleich: Universum existiert 10 Milliarden Jahre (12-Mio.-fach!)

Bessere Faktorisierungen

- Algorithmen für spezielle Formen der Zahl N
- Allgemeine Methode: **Zahlkörpersieb**

- Was ist damit möglich?
- **RSA Factoring Challenge**
  - www.rsa.com
  - http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-factoring-challenge.htm

1999: RSA FC 512 bit

- $N \approx 2^{512} \approx 10^{155}$
- 300 Computer, 7,5 Monate
- 187 CPU-Jahre

10941738641570527421809707322040357612003732945449
20599091384213147634998428893478471799725789126733
24976257528997818337970765372440271467435315933543
33897
=
10263959282974110577205419657399167590071656780803
80668033419335217907113077 79
*
10660348838016845482092722036001287867920795857598
92915222706082371930628086 43

2009: RSA FC 768 bit

- $\mathrm{N} \approx 2^{768} \approx 10^{232}$
- 660 Computer, 3 Jahre
- 2000 2.2GHz-CPU Jahre

123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3

=

334780716989568987860441698482126908177047949837137685689124313889828837938780022876147116525317430877737814467999489

*

367460436667995904282446337996279526322791581643430876426760322838157396665112792333734171433968102700927987363089 17

Dann: Standard 1024 bit

- $N \approx 2^{1024} \approx 10^{309}$, z.B.

135066410865995223349603216278805969938814756056670275244851438515265106048595338339402871505719094417982072821644715513736804197039641917430464965892742562393410208643832021103729587257623585096431105640735015081875106765946292055636855294752135008528794163773285339061097505443349998111500569772368909275 63

= p * q

Seit 2014: Empfehlung 2048 bit

- $N \approx 2^{2048} \approx 10^{620}$
- z.B. mit WolframAlpha:

nextprime[2^1024]* nextprime[2^1024+2^10]

Web Apps Examples Ra

Input:

$\text{NextPrime}[2^{1024}]\ \text{NextPrime}[2^{1024}+2^{10}]$

Result:

32 317 006 071 311 007 300 714 876 688 669 951 960 444 102 669 715 484 032 130 ⋱
345 427 524 655 138 867 890 893 197 201 411 522 913 463 688 717 960 921 898 019 ⋱
494 119 559 150 490 921 095 088 152 386 448 283 120 630 877 367 300 996 091 750 ⋱
197 750 389 652 106 796 057 638 384 067 568 276 792 218 642 619 756 161 838 094 ⋱
338 476 170 470 581 645 852 036 305 042 887 575 891 541 065 808 607 552 399 123 ⋱
930 385 831 836 629 839 931 604 913 217 189 678 592 433 570 595 407 204 979 975 ⋱
197 471 265 575 262 159 281 392 120 754 939 673 496 694 769 217 137 019 932 616 ⋱
744 005 961 351 912 588 355 903 082 072 668 035 686 677 658 498 973 949 178 916 ⋱
826 070 183 694 275 197 585 406 793 551 975 206 643 412 017 773 626 759 006 299 ⋱
241 727 738 775 594 205 159 882 555 660 007 770 314 370 476 543 983 057 729 710 ⋱
165 883 222 009 486 123

nextprime[2^1024]

Web Apps Examples Random

Input:

$\text{NextPrime}[2^{1024}]$

Open code

Result:

179 769 313 486 231 590 772 930 519 078 902 473 361 797 697 894 230 657 273 430 ⋱
081 157 732 675 805 500 963 132 708 477 322 407 536 021 120 113 879 871 393 357 ⋱
658 789 768 814 416 622 492 847 430 639 474 124 377 767 893 424 865 485 276 302 ⋱
219 601 246 094 119 453 082 952 085 005 768 838 150 682 342 462 881 473 913 110 ⋱
540 827 237 163 350 510 684 586 298 239 947 245 938 479 716 304 835 356 329 624 ⋱
224 137 859

nextprime[2^1024+2^10]

Web Apps Examples Ran

Input:

$\text{NextPrime}[2^{1024}+2^{10}]$

Result:

179 769 313 486 231 590 772 930 519 078 902 473 361 797 697 894 230 657 273 430 ⋱
081 157 732 675 805 500 963 132 708 477 322 407 536 021 120 113 879 871 393 357 ⋱
658 789 768 814 416 622 492 847 430 639 474 124 377 767 893 424 865 485 276 302 ⋱
219 601 246 094 119 453 082 952 085 005 768 838 150 682 342 462 881 473 913 110 ⋱
540 827 237 163 350 510 684 586 298 239 947 245 938 479 716 304 835 356 329 624 ⋱
224 138 297

Vielen Dank für Ihre Aufmerksamkeit!

Notabene: größte bekannte Primzahl

Zahl	Anzahl der Dezimalziffern	Jahr	Entdecker (genutzter Computer)
$2^{17}-1$	6	1588	Cataldi
$2^{19}-1$	6	1588	Cataldi
$2^{31}-1$	10	1772	Euler
…	…	…	…
$2^{43.112.609}-1$	12.978.189	2008	Smith, Woltman, Kurowski et al. (GIMPS, Core 2 Duo 2,4 GHz)
$2^{57.885.161}-1$	17.425.170	2013	Cooper, Woltman, Kurowski et al. (GIMPS)
$2^{74.207.281}-1$	22.338.618	2016	Cooper, Woltman, Kurowski et al. (GIMPS)

https://de.wikipedia.org/wiki/Primzahl#Gr.C3.B6.C3.9Fte_bekannte_Primzahl