Verschlüsselte E-Mails — Wie geht das?

Ralf Hemmecke

Research Institute for Symbolic Computation
Johannes Kepler University Linz, Austria

08. Mai 2015

Outline

Outline

Wie funktioniert E-Mail?

- Besteht aus Kopf und Nachricht, durch Leerzeile getrennt.
- Kopf besteht aus Reihe von Schlüsselwörtern mit Werten.

```
Return-Path: <admin@risc.jku.at>
Date: Thu, 07 May 2015 11:24:38 +0200
From: Mr. Admin <admin@risc.jku.at>
To: Ralf HEMMECKE <hemmecke@risc.jku.at>
Subject: Mathe-Seminar
```

- Sendung per SMTP (simple mail transfer zum Server
- Empfang per POP3 (post office protocol) oder IMAP (internet message access protocol) vom Server
- SMTPS, POP3S, IMAPS (secure (?) von und zum Server)
- Übertragung zwischen verschiedenen Servern im Internet

Was ist das?

- Klartext
- Schlüssel
- verschlüsselter Text
- Schlüssellänge

Verschiedene einfache Verschlüsselungen

- Skytale (Transposition)
- Cäsar-Chiffre (Buchbstabenverschiebung, Chiffriescheibe) (5 bit)
- Vigenère-Chiffre (multi-Cäsar)
- Buchstabenvertauschung: Anzahl der Schlüssel=26! (88 bit)
- One Time Pad (beweisbar unknackbar)
- ...

Verschlüsselungsverfahren

- symmetrische Verschlüsselung
  - ein geheimer Schlüssel
  - Sender und Empfänger kennen geheimen Schlüssel
  - Problem: Übergabe des geheimen Schlüssels
- asymmetrische Verschlüsselung
  - öffentlicher Schlüssel zum Verschlüsseln
  - privater (geheimer) Schlüssel zum Entschlüsseln
  - Problem: Sicherstellung, dass öffentlicher Schlüssel zum Empfänger passt

Hybride Verfahren

- symmetrische Verschlüsselung
  - kürzere Schlüssel als bei asymmetrischen Verfahren
  - schnelle Ver- und Entschlüsselung
  - Problem: Sichere Übermittlung des Schlüssels
- asymmetrische Verschlüsselung
  - mathematisch eigentlich unsicher
  - praktisch aber nicht knackbar bei hinreichend langen Schlüsseln
  - langsam
- Hybrid-Verfahren
  - Erzeugung eines Zufallsschlüssels
  - Übertragung des Zufallsschlüssels mittels asymmetrischer Verschlüsselung
  - Nachricht wird mit Zufallsschlüssel (symmetrisch) ver- und entschlüsselt

symmetrische Verschlüsselung

- Data Encryption Standard
  - DES, 1975 (56 bit Schlüssel)
  - 3DES, 1981 (2 unabhängige Schlüssel, 112 bit)
- Advanced Encryption Standard (AES, 2000, 128-256 bit)
- Blowfish (1993) (Nachfolger: Twofish, Threefish)

asymmetrische Verschlüsselung

- RSA Kryptosystem, 1977 (Rivest, Shamir, Adleman)
  - ganzzahliges Faktorisierungsproblem
- ElGamal Kryptosystem, 1985
  - diskretes Logarithmusproblem in Z_p
- Kryptosysteme basierend auf elliptischen Kurven
  - $y^2 = x^3 + ax + b$
  - diskretes Logarithmusproblem „über elliptischen Kurven“

Outline

RSA Kryptosystem

Erfunden 1977 von Rives, Shamir, Adleman.

- Wähle 2 unterschiedliche Primzahlen p und q.
- $n = p \cdot q$
- $b = (p-1) \cdot (q-1)$
- Wähle $1 < e < b$ mit $\mathrm{ggT}(e, b) = 1$.
- $d \equiv e^{-1} \pmod{b}$
- öffentlich: (n, e)
- geheim: (p, q, b, d)
- Verschlüsselung: $c \equiv m^e \pmod{n}$
- Entschlüsselung: $m \equiv c^d \pmod{n}$

Sicherheit liegt im hohen Aufwand aus einem großen n die Primzahlen p und q zu bestimmen. Heutzutage ist $n > 2^{2000} > 10^{600}$ üblich.

Probleme

- Wie kommt man von einem Klartext zu einer Zahl m und zurück?
- Wie findet man zwei passende Primzahlen?
- Was bedeutet $d \equiv e^{-1} \pmod{b}$?
- Wie berechnete man $c \equiv m^e \pmod{n}$ effizient?

Klartext $\longleftrightarrow$ ganze Zahl

- Zeichenkodierung
  - ASCII (0-0x7F):
    0=48, 1=49, A=65, I=73, J=74, a=97, LEERZEICHEN=32, !=33, "=34, ...
  - EBCDIC (0-0xFF):
    0=240, 1=241, A=193, I=201, J=209, a=129, LEERZEICHEN=40, !=90, "=127, ...
  - ISO-8859-1 (0-0xFF):
    ASCII + westeuropäische Sonderzeichen (Ä=196, Ö=213, ...)
  - ISO-8859-5 (0-0xFF):
    ASCII + kyrillische Buchstaben
  - Unicode (0-0x1FFFFF) Codierungen: UTF-8, UTF-16, UTF-32
- Text wird gepackt (zip), um weniger Redundanz zu erzeugen.
  (Häufigkeitsanalyse verhindern: E=17,40 N=9,78 I=7,55 S=7,27 R=7 A=6,51 T=6,15 ...)
- Textblöcke als Binärdarstellung einer ganzen Zahl interpretieren.
- Alle Schritte sind umkehrbar.

Primzahlen

- Was sind Primzahlen?
- Aufgaben
  1. Wähle Zahl n von 10-1000 und berechne $z = (n-1)^{(n-1)}$. Welchen Rest lässt z bei Division durch n?
  2. Was stellen wir fest, wenn n eine Primzahl ist?
  3. Wähle Zahl n von 10-1000 und $1 < a < n$. Berechne $z = a^{(n-1)}$. Welchen Rest lässt z bei Division durch n?
  4. Welche Reste entstehen bei Primzahlen n?
  5. Welche Reste entstehen bei Nichtprimzahlen n?

Kleiner Satz von Fermat

Satz (Fermat)

Sei p eine Primzahl. Dann gilt für alle $a \in \mathbf{Z}$

$$a^p \equiv a \pmod{p}.$$

Gilt zusätzlich $p \nmid a$, dann ist

$$a^{p-1} \equiv 1 \pmod{p}.$$

Fermatscher Primzahltest

Der Fermatsche Primzahltest ist ein probabilistischer Test und beruht auf dem kleinen fermatschen Satz.

- geg: n, k (k Iterationen ausführen)
- ges: Ist n prim?
- Wähle $0 < a < n$ zufällig.
- Wenn $a^{n-1} \not\equiv 1$, dann ist n mit Sicherheit nicht prim.
- Wiederhole obigen Test k mal.
- Wenn kein Test n als nicht-prim erkannt hat, melde: n ist wahrscheinlich prim.

Problem: Test versagt für Carmichael-Zahlen (kleinste: $561 = 3 \cdot 11 \cdot 17$).
Es gibt verbesserte Tests, z.B. von Miller-Rabin oder Solovay-Strassen.

Restklassen

- Zwei ganze Zahlen sind kongruent modulo b, wenn ihre Differenz ein Vielfaches von b ist.

$$a_1 \equiv a_2 \pmod{b} \iff \exists c \in \mathbf{N} : a_1 - a_2 = cb$$

- Alle ganzen Zahlen, die (modulo b) kongruent zu einer gegebenen Zahl sind, können wir zu einer Restklasse zusammenfassen.

$$[a]_b = \{\, n \in \mathbf{N} \mid n \equiv a \pmod{b} \,\}$$

- Da Restklassen Mengen sind, gilt:

$$[a_1]_b = [a_2]_b \iff a_1 \equiv a_2 \pmod{b}$$

- Es gibt b verschiedene Restklassen, die wir durch Zahlen

$$0, 1, 2, \ldots, b-1 \tag{1}$$

repräsentieren können.

Addieren und Multiplizieren modulo b

- Mit Restklassen kann man rechnen.

$$[a_1]_b \oplus [a_2]_b = [a_1 + a_2]_b$$
$$[a_1]_b \otimes [a_2]_b = [a_1 \cdot a_2]_b$$

- Meist rechnet man mit den Repräsentanten (Beispiel):

$$[6]_{13} \oplus [3]_{13} = [9]_{13} \qquad 6 + 3 \equiv 9 \pmod{13}$$
$$[8]_{13} \oplus [9]_{13} = [4]_{13} \qquad 8 + 9 \equiv 17 \equiv 4 \pmod{13}$$
$$[6]_{13} \otimes [3]_{13} = [5]_{13} \qquad 6 \cdot 3 \equiv 18 \equiv 5 \pmod{13}$$
$$[6]_{13} \oplus [-11]_{13} = [8]_{13} \qquad 6 - 11 \equiv -5 \equiv 8 \pmod{13}$$

- Statt $\oplus$ und $\otimes$ verwendet man auch einfach $+$ und $\cdot$ als Zeichen.

Beispielaufgaben

- Multiplikationstabelle modulo 5:

	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

- Berechne eine Multiplikationstabelle modulo 11, 12, und 13.
- Welche Besonderheiten stellen wir fest?
  Hinweis: Suche 0 und 1 in Tabelle.

Dividieren modulo b

Satz

Seien $a, b \in \mathbf{Z}$. Dann gilt:

- a ist invertierbar modulo $b \iff \mathrm{ggT}(a, b) = 1$.
- Es existieren $u, v \in \mathbf{Z}$ mit $\mathrm{ggT}(a, b) = u \cdot a + v \cdot b$.

Berechnung des Inversen von a modulo b, wenn a und b teilerfremd sind.

- Bestimme u und v (Kofaktoren) so, dass

$$1 = \mathrm{ggT}(a, b) = u \cdot a + \underbrace{v \cdot b}_{\text{Rest 0 bei Division durch } b.}$$

- $1 \equiv u \cdot a \pmod{b}$
- $u \equiv a^{-1} \pmod{b}$

Erweiterter Euklidischer Algorithmus

Algorithmus EEA: größten gemeinsamen Teiler mit Kofaktoren berechnen

- geg: $a, b \in \mathbf{N}$
- ges: $d \in \mathbf{N}$, $u, v \in \mathbf{Z}$ mit $d = \mathrm{ggT}(a, b) = u \cdot a + v \cdot b$

EA

```
EA(a, b) ==


 while not(b = 0) repeat
     -- Bestimme q so, dass a = q*b + r mit |r|<|b|.
     q := a quo b;      -- ganzzahlige Division
     r  := a  - q*b;      a  := b;      b  := r


 return  a
```

Erweiterter Euklidischer Algorithmus

Algorithmus EEA: größten gemeinsamen Teiler mit Kofaktoren berechnen

- geg: $a, b \in \mathbf{N}$
- ges: $d \in \mathbf{N}$, $u, v \in \mathbf{Z}$ mit $d = \mathrm{ggT}(a, b) = u \cdot a + v \cdot b$

EEA

```
EEA(a, b) ==
  ua := 1; va := 0;    --Invariante: a = ua * a + va * b
  ub := 0; vb := 1;    --Invariante: b = ub * a + vb * b
  while not(b = 0) repeat
      -- Bestimme q so, dass a = q*b + r mit |r|<|b|.
      q := a quo b;       -- ganzzahlige Division
      r  := a  - q*b;       a  := b;       b  := r
      ur := ua - q*ub;      ua := ub;      ub := ur;
      vr := va - q*vb;      va := vb;      vb := vr
  return (a, ua, va)
```

Beispiel: Berechnung von $12^{-1} \pmod{37}$

$37 = 3 \cdot 12 + 1$
Also $1 = 1 \cdot 37 + (-3) \cdot 12$.
Es gilt $-3 \equiv 34 \pmod{37}$.
Probe:
$34 \cdot 12 = 408 = 11 \cdot 37 + 1 \equiv 1 \pmod{37}$,
also $12^{-1} \equiv 34 \pmod{37}$.

Beispiel: Berechnung von $27^{-1} \mod 37$ mit erweitertem Euklidischen Algorithmus

Wir führen eine abkürzende Schreibweise ein.

```
(r,ur,vr) := (a,ua,va) - q*(b,ub,vb)
```

```
r  := a  - q*b
ur := ua - q*ub
vr := va - q*vb
```

$$r = a - (a \text{ quo } b) \cdot b \qquad (r, p_r, q_r) = (a, p_a, q_a) - (a \text{ quo } b) \cdot (b, p_b, q_b)$$

$$10 = 37 - 1 \cdot 27 \qquad (10, 1, -1) = (37, 1, 0) - 1 \cdot (27, 0, 1)$$

$$7 = 27 - 2 \cdot 10 \qquad (7, -2, 3) = (27, 0, 1) - 2 \cdot (10, 1, -1)$$

$$3 = 10 - 1 \cdot 7 \qquad (3, 3, -4) = (10, 1, -1) - 1 \cdot (7, -2, 3)$$

$$1 = 7 - 2 \cdot 3 \qquad (1, -8, 11) = (7, -2, 3) - 2 \cdot (3, 3, -4)$$

$$0 = 3 - 3 \cdot 1 \qquad (0, 27, -37) = (3, 3, -4) - 3 \cdot (1, -8, 11)$$

Also: $27^{-1} \equiv 11 \pmod{37}$.

Korrektheit des RSA-Verfahrens

Satz

Seien p und q zwei verschiedene Primzahlen, $n = pq$, $b = (p-1)(q-1)$ und e, d so gewählt, dass $1 < e < b$, $\mathrm{ggT}(e, b) = 1$, $ed \equiv 1 \pmod{b}$. Dann gilt für jede natürliche Zahl m:

$$m \equiv (m^e)^d \pmod{n}.$$

Der Beweis benötigt 2 Hilfssätze:

- (kleiner) Satz von Fermat
- Chinesischer Restklassensatz

Chinesischer Restklassensatz (CRT)

Satz

Seien $n_1, n_2, \ldots, n_k$, positive natürliche Zahlen, die paarweise teilerfremd sind. Seien $r_1, r_2, \ldots, r_k$ beliebige ganzen Zahlen. Dann existiert $x \in \mathbf{N}$ mit

$$\begin{aligned} x &\equiv r_1 \pmod{n_1}, \\ x &\equiv r_2 \pmod{n_2}, \\ &\vdots \\ x &\equiv r_k \pmod{n_k}. \end{aligned}$$

Korrektheitsbeweis des RSA-Verfahrens

- Nach CRT genügt $m \equiv m^{ed} \pmod{p}$ und $m \equiv m^{ed} \pmod{q}$ zu beweisen.
- Da Problem symmetrisch in p und q, reicht es, $m \equiv m^{ed} \pmod{p}$ zu zeigen.
- Fall 1 ($p \mid m$): Trivial.
- Fall 2 ($p \nmid m$): Es gilt $ed = 1 + t(p-1)(q-1)$ für eine gewisse ganze Zahl t. Also

$$\begin{aligned} m^{ed} &\equiv m^{1+t(p-1)(q-1)} \pmod{p} \\ &\equiv m \cdot (m^{p-1})^{t(q-1)} \pmod{p} \\ &\equiv m \cdot 1^{t(q-1)} \pmod{p} \\ &\equiv m \pmod{p} \end{aligned}$$

Outline

Thunderbird + EnigMail + GnuPG

- Installiere Thunderbird

  `https://www.mozilla.org/de/thunderbird`

- Starte Thunderbird
- Einrichtung eines Mailkontos
- Installiere Add-On Enigmail
- (Installiere GPG 2.0)
- Enigmail > Einrichtungsassistent (neues Schlüsselpaar)
- Enigmail > Schlüssel verwalten > Öffentlichen Schlüssel per E-Mail senden > `mail@hemmecke.net`

Outline

- Nehmen wir an, der Tag Deiner Geburt trägt die Nummer 1. Der Tag 10000 war welcher Wochentag?
- Zeige $2^{100000} - 2^{100}$ ist teilbar durch 1000.
- Auf welche Ziffer endet $1234567^{7654321}$?
- Gibt es eine Lösung für $8x^2 + 5y^2 = 2015z^2 - 1$?
- Beweise. Wenn für beliebige natürliche Zahlen gilt $c \cdot a \equiv c \cdot b \pmod{m}$ und $\mathrm{ggT}(c, m) = 1$, dann gilt $a \equiv b \pmod{m}$.
- Zeige, dass $65^{63} + 63^{65}$ durch 64 teilbar ist.
- Zeige $(x + y)^p \equiv x^p + y^p \pmod{p}$ für alle Primzahlen p.
- Berechne $x \equiv \frac{17}{13} \pmod{5}$.
- Sei p Primzahl. Ist $a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \equiv 0 \pmod{p}$ immer lösbar in x? Kann man alle Lösungen x bestimmen? Wenn ja, wie? Wieviele Lösungen gibt es?

Wie berechnet man $c \equiv m^e \pmod{n}$ effizient?

- geg: $n, m, e \in \mathbf{N}$
- ges: $m^e \pmod{n}$
- Bilde Binärdarstellung von e.
- Ersetze 0 durch Q und 1 durch QM.
- Diese Zeichenkette ist Rechenvorschrift für m^e. Lies von links nach rechts.
  - Starte mit $c = 1$.
  - Wenn Q, dann $c := c^2$.
  - Wenn M, dann $c := c \cdot m$.
  - Reduziere in jedem Schritt modulo n.

CRT Lösungsalgorithmus (k=2)

- Gegeben: r_1, r_2, n_1, n_2
- Gesucht: x mit

$$x \equiv r_1 \pmod{n_1},$$
$$x \equiv r_2 \pmod{n_2},$$

- Berechne mit EEA i_1 und i_2 so dass mit $N = n_1 n_2$

$$1 = i_1 n_2 + i_2 n_1 = i_1 \frac{N}{n_1} + i_2 \frac{N}{n_2}$$

- Dann erfüllt

$$x = r_1 i_1 n_2 + r_2 i_2 n_1 = r_1 i_1 \frac{N}{n_1} + r_2 i_2 \frac{N}{n_2}$$

alle Bedingungen.

CRT Lösungsalgorithmus (k>2)

- Für $N = n_1 \cdots n_k$ und $j = 1, \ldots k$ berechne i_j mittels EEA, so dass

$$1 = i_j \frac{N}{n_j} + c_j n_1$$

  Bemerkung: Berechnung von c_j ist unnötig.

- Dann erfüllt

$$x = r_1 i_1 \frac{N}{n_1} + \cdots r_k i_k \frac{N}{n_k}$$

  alle Bedingungen.

Diffie-Hellman (1976)

- A und B vereinbaren eine Primzahl p und eine Basis $g > 1$.
- A wählt Geheimzahl a und sendet $g^a \pmod p$ an B.
- B wählt Geheimzahl b und sendet $g^b \pmod p$ an A.
- A berechnet $(g^b)^a \pmod p$.
- B berechnet $(g^a)^b \pmod p$.
- gemeinsames Geheimnis ist: $g^{ab} \pmod p$
- öffentlich: p, g

Authentifizierung

Wie kann man feststellen, ob die Person, mit der man kommuniziert, die *richtige* Person ist?

- Fingerprint des Schlüssels in persona überprüfen (d.h. persönliches Treffen der beteiligten Personen im Vorfeld)
- Fingerprint auf anderem Kommunikationsweg überprüfen.
- Geheimnis über Telefon/Email ... austauschen (Frage/Antwort).
- Web of Trust