Elementare Zahlentheorie

Diskrete Strukturen

Winter Semester 2012

342 207

Prof. Armin Biere
Institut für Formale Modelle und Verifikation
Johannes Kepler Universität, Linz

http://fmv.jku.at/ds

Literatur für die Zahlentheorie

- Kapitel 2, *Mathematik explorativ*. J. Mühlbacher, G. Pilz, M. Widi. Trauner Verlag. Online Version unter `http://www.fim.at` erhältlich.
- Kapitel 2, Skriptum *Mathematik 2 für Informatiker (Algebra)*. J. Mühlbacher, G. Pilz, M. Widi.
- Kapitel 2, *Discrete Mathematics and its Applications*. K. Rosen. McGraw Hill.
- *Die Kleine Fibel der Arithmetik*. H. Lüneburg. BI-Wissenschaftsverlag.
- Diese Folien erhältlich unter `http://fmv.jku.at/ds`

Def. Sei $\mathbb{Z}$ die Menge der *Ganzen Zahlen* und $\mathbb{N}$ die Menge der *Natürlichen Zahlen*.

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

$$\mathbb{N} = \{1, 2, \ldots\}$$

$$\mathbb{N}_0 = \{\mathbf{0}, 1, 2, \ldots\} = \mathbb{N} \cup \{0\}$$ "natürlicher" in der Informatik

Def. Ein $b \in \mathbb{Z}$ ist ein *Teiler* eines $a \in \mathbb{Z}$ gdw. ein $q \in \mathbb{Z}$ gibt, so dass

$$a = b \cdot q$$

Man schreibt dann $b \mid a$.

Bsp. $5 \mid 30$ sowie $3 \mid 30$, $1 \mid a$ für alle $a \in \mathbb{Z}$, $0 \mid a$ nur für $a = 0$.

Anm. Im Skript wird "$/$" statt "$\mid$" verwendet.

Gemeinsamer Teiler

Def. Ein $c \in \mathbb{Z}$ heißt *gemeinsamer Teiler* von $a \in \mathbb{Z}$ und von $b \in \mathbb{Z}$, gdw. $c \mid a$ und $c \mid b$.

Bsp.

$c = 1$ ist ein gemeinsamer Teiler von allen $a, b \in \mathbb{Z}$. Wirklich?

15 ist ein gemeinsamer Teiler von 60 und von 90.

Fakt Wenn $c \mid a$, dann $c \leq a$ für alle $a, c \in \mathbb{N}$

Fakt Wenn c ein gemeinsamer Teiler von a und b ist, dann $c \leq \min(a, b)$ für $a, b, c \in \mathbb{N}$.

Def. Es heißt $d = ggT(a,b)$ der *größte gemeinsame Teiler* von a und b, gdw. erstens $d \mid a$ und $d \mid b$, also d ein gemeinsamer Teiler von a und b ist, und zweitens für jeden gemeinsamen Teiler c von a und b, also $c \mid a$ und $c \mid b$, gilt, dass $c \leq d$.

Bsp. $15 < 30 = ggT(60,90), \quad ggT(a,0) = a, \quad ggT(b,1) = 1.$ Fehler?

Fakt. $1 = ggt(u,g)$ für alle Zweierpotenzen $g = 2^i$ und alle ungeraden Zahlen u.

Anm. Im Englischen *Greatest Common Divisor* abgekürzt gcd.

Beispiel zum ggT

Ges. $d = ggT(56, 21)$.

Es gilt: $56 = 2 \cdot 21 + 14$.

Jeder Teiler von 56 und 21 ist auch ein Teiler von 21 und 14.

Erst recht ist d ein Teiler von 21 und 14.

Dann ist $21 = 1 \cdot 14 + 7$ und somit auch $d \mid 7$.

Schließlich ist $14 = 2 \cdot 7 + 0$ womit $d = 7$.

$\Rightarrow$ Beispiel lässt sich zum *Euklid'schen Algorithmus* generalisieren.

Hilfssatz zum Euklid'schen Algorithmus

Fakt $c \mid a$ und $c \mid b$ mit $a \geq b$ dann auch $c \mid (a+b)$, $c \mid (a-b)$ und $c \mid a \cdot d$ für bel. d.

Bew. Ansatz $a = p \cdot c$ und $b = q \cdot c$ ...

Satz Aus $a = b \cdot q + r$ mit $0 \leq r < b$, folgt $ggT(a,b) = ggt(b,r)$.

Bew.

Gelte $d \mid a$ und $d \mid b$, dann ist auch $d \mid r$, denn $r = a - b \cdot q$.

umgekehrt, wenn $c \mid b$ und $c \mid r$, dann auch $c \mid a$ nach Vor.

Die Mengen der gemeinsamen Teiler von a und b, bzw. von b und r sind identisch.

Deren Maximum auch. q.e.d.

Simulation des Euklid'schen Algorithmus

Wähle Quotienten q_i mit Rest r_i, so dass

$$a = r_0$$

$$b = r_1$$

$$r_0 = r_1 \cdot q_2 + r_2 \qquad 0 \leq r_2 < r_1$$

$$r_1 = r_2 \cdot q_3 + r_3 \qquad 0 \leq r_3 < r_2$$

$$r_2 = r_3 \cdot q_4 + r_4 \qquad 0 \leq r_4 < r_3$$

$$\vdots$$

$$r_{i-2} = r_{i-1} \cdot q_i + r_i \qquad 0 \leq r_i < r_{i-1}$$

Es gibt keine unendlich echt absteigende Kette natürlicher Zahlen.

$$r_0 > r_1 > r_2 > \ldots r_{i-1} > r_i$$

Damit gibt es ein n mit $r_n = 0$.

$$ggT(a,b) = ggT(r_0,r_1) = ggT(r_1,r_2) = \ldots = ggT(r_{n-1},r_n) = ggT(r_{n-1},0) = r_{n-1}$$

Iterativer Euklids'cher Algorithmus

```
int ggT (int a, int b) {

  int r;

  while (b > 0) {

    r = a % b;          Was passiert, wenn a < b?

    a = b;

    b = r;

  }

  return a;

}
```

Rekursiver Euklids'cher Algorithmus

```
int ggT (int a, int b) {

   if (b == 0) return a;

   return ggT (b, a % b);

}
```

Bei der rekursiven Variante ist der Speicherplatz-Verbrauch nicht konstant!

Optimierende Compiler könnten Code mit konstantem Speicherverbrauch erzeugen.

Die Rekursions-Tiefe ist aber logarithmisch beschränkt in der Größe der gegebenen Zahlen.

Def. Ein $u \in \mathbb{Z}$ heißt *gemeinsames Vielfaches* von $a \in \mathbb{Z}$ und $b \in \mathbb{Z}$, gdw. $a \mid u$ und $b \mid u$.

Def. Analog zum ggT heißt $v = kgV(a,b)$ das *kleinste gemeinsame Vielfache* von $a \in \mathbb{Z}$ und $b \in \mathbb{Z}$, gdw. erstens, u ein gemeinsames Vielfache von a und b ist, und jedes weitere gemeinsame Vielfache w von a und b, größer ist als v.

Satz $kgV(a,b) \cdot ggT(a,b) = a \cdot b$

Bsp.

$$kgV(15,12) = 60 = \frac{180}{3} = \frac{15 \cdot 12}{ggT(15,12)}$$

Anm. Im Englischen *Least Common Multiple* abgekürzt lcm.

Hauptsatz über den ggT

Satz Zu $d = ggT(a,b)$ gibt es $x,y \in \mathbb{Z}$ mit $a \cdot x + b \cdot y = d$.

Bew. konstruktiv durch Angabe eines Verfahrens

$$\begin{array}{lll}
a = r_0 & x_0 = 1 & y_0 = 0 \\
b = r_1 & x_1 = 0 & y_1 = 1 \\
r_0 = r_1 \cdot q_2 + r_2 & x_2 = x_0 - x_1 \cdot q_2 & y_2 = y_0 - y_1 \cdot q_2 \\
r_1 = r_2 \cdot q_3 + r_3 & x_3 = x_1 - x_2 \cdot q_3 & y_3 = y_1 - y_2 \cdot q_3 \\
r_2 = r_3 \cdot q_4 + r_4 & x_4 = x_2 - x_3 \cdot q_4 & y_4 = y_2 - y_3 \cdot q_4 \\
\vdots & \vdots & \vdots \\
r_{i-2} = r_{i-1} \cdot q_i + r_i & x_i = x_{i-2} - x_{i-1} \cdot q_i & y_i = y_{i-2} - y_{i-1} \cdot q_i
\end{array}$$

Nun gilt die Invariante: $r_i = a \cdot x_i + b \cdot y_i$ denn gilt schon

$$r_{i-1} = a \cdot x_{i-1} + b \cdot y_{i-1} \quad \text{bzw.} \quad r_{i-2} = a \cdot x_{i-2} + b \cdot y_{i-2}$$

so erhält man zunächst $r_i = r_{i-2} - r_{i-1} \cdot q_i$ was sich auch wie folgt darstellen lässt

$$a \cdot x_{i-2} + b \cdot y_{i-2} - (a \cdot x_{i-1} + b \cdot y_{i-1}) \cdot q_i = a \cdot \underbrace{(x_{i-2} - x_{i-1} \cdot q_i)}_{x_i} + b \cdot \underbrace{(y_{i-2} - y_{i-1} \cdot q_i)}_{y_i}$$

Beispiele für Erweiterten Euklid'schen Algorithmus

x	y	q	r
1	0		56
0	1		21
1	-2	2	14
-1	3	1	7

$$56 \cdot (-1) + 21 \cdot 3 = -56 + 63 = 7$$

x	y	q	r
1	0		29
0	1		21
1	-1	1	8
-2	3	2	5
3	-4	1	3
-5	7	1	2
8	-11	1	1

$$29 \cdot 8 + 21 \cdot (-11) = 232 - 231 = 1$$

Iterative Formulierung Erweiterter Euklid'scher Algorithmus

```
(int,int) ggTextended (int a, int b) {

  int r, q;

  int tx, lx = 1, ly = 0;

  int ty,  x = 0,  y = 1;

  while (b > 0) {

    r = a % b; q = a / b; a = b; b = r;

    tx = x; x = lx - x * q; lx = tx;

    ty = y; y = ly - y * q; ly = ty;

  }

  return (lx,ly);

}
```

Satz

Sind $a,b,c \in \mathbb{Z}$ mit $c \mid (a \cdot b)$ und $ggT(a,c) = 1$, so folgt $c \mid b$.

Bew.

Da $ggT(a,c) = 1$, gibt es x,y mit $a \cdot x + c \cdot y = 1$.

Also gilt auch $b \cdot (a \cdot x + c \cdot y) = b \cdot a \cdot x + b \cdot c \cdot y = b$.

Nach Vor. gilt $c \mid (a \cdot b)$ also auch $c \mid (b \cdot a \cdot x)$.

Weiter gilt trivialerweise $c \mid (b \cdot c \cdot y)$ und damit auch $c \mid (b \cdot a \cdot x + b \cdot c \cdot y) = b$.

Primzahlen und Erster Satz von Euklid

Def. $p \in \mathbb{N}$ mit $p > 1$ heißt *Primzahl*, wenn p außer 1 und p keinen weiteren Teiler hat.

Bsp. $2, 3, 5, 7, \ldots$ sind Primzahlen, $0, 1, 4, 6, 8, 9, \ldots$ keine.

Satz Ist p prim und $p \mid (a \cdot b)$, so folgt $p \mid a$ oder $p \mid b$.

Bew. Sei p prim, $p \mid (a \cdot b)$ und $p \not| a$.

Dann gilt $ggT(p, a) = 1$.

Mit dem Fundamentalsatz ergibt sich $p \mid b$. q.e.d.

Satz

Jede natürliche Zahl $n \in \mathbb{N}$, $n > 1$, lässt sich in Primfaktoren zerlegen,

d.h. es gibt ein $k \in \mathbb{N}$ und k Primzahlen $p_1, \ldots, p_k$, so dass $n = p_1 \cdot p_2 \cdots p_k$.

Diese Darstellung ist eindeutig, bis auf die Reihenfolge der Faktoren.

Anm.

Nach dem Hauptsatz ist die Reihenfolge zunächst nicht festgelegt.

Man kann aber die p_i immer der Größe nach ordnen, d.h. $p_1 \leq p_2 \leq \cdots \leq p_k$.

Dann fasst man gleiche p_i zusammen und bekommt eine *kanonische Darstellung*:

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdots p_s^{e_s} = \prod_{j=1}^{s} p_j^{e_j} \qquad \textit{Vielfachheit } e_j$$

Beweis des Hauptsatzes: Existenz einer Zerlegung

Induktionsanfang: der Fall $n = 2$ ist trivial.

Induktionsannahme: der Satz gelte für alle $m < n$.

Induktionsschritt: Sei T die Menge der Teiler von n, mit $t > 1$ für alle $t \in T$.

T ist nicht leer, da $n \in T$, d.h. es gibt ein kleinstes Element $p_1 \in T$.

Sei $c \in \mathbb{N}$ bel. mit $c \mid p_1$, so folgt $c \mid n$, also $c \in T \cup \{1\}$

da p_1 minimal in T, gilt $c = 1$ oder $c = p_1$,

also ist p_1 eine Primzahl

Im Falle $n = p_1$ sind wir fertig, sonst sei $p_2 \cdots p_k$ eine Zerlegung von n/p_1 und

$p_1 \cdot p_2 \cdots p_k$ ist eine Zerlegung von n

Beweis des Hauptsatzes: Eindeutigkeit der Zerlegung

Es seien $p_1 \cdots p_k$ und $q_1 \cdots q_r$ zwei Zerlegungen von n.

O.B.d.A. sei $n > 1$ und $k \leq r$. Zunächst gilt $p_1 \cdots p_k = n = q_1 \cdots q_r$.

Da $p_1 \mid n$, somit auch $p_1 \mid q_1 \cdots q_r$.

Mit dem ersten Satz von Euklid folgt o.B.d.A. $q_1 = p_1$, da die Reihenfolge egal ist.

Somit hat man $p_2 \cdots p_k = q_2 \cdots q_r$.

Nach $k-1$ Schritten erhält man $1 = q_{k+1} \cdots q_r$.

Damit ergibt sich $k = r$ und die Eindeutigkeit. q.e.d.

Charakterisierung von Relativ Prim durch Zerlegungen

Def. Zwei Zahlen $a, b \in \mathbb{Z}$ heißen *relativ prim*, gdw. $ggT(a,b) = 1$.

Satz Zwei $a, b \in \mathbb{N}$, $a, b > 1$ sind relativ prim gdw. jede Primzahl p, die in der Zerlegung von a vorkommt, nicht in der Zerlegung von b vorkommt, und umgekehrt.

$$a = p_1^{e_1} \cdot p_2^{e_2} \cdots p_s^{e_s} \qquad b = q_1^{d_1} \cdot q_2^{d_2} \cdots q_t^{d_t}$$

Satz Es gibt unendlich viele Primzahlen

Bew. *indirekt durch Widerspruch*

Ann. es gäbe nur k verschiedene Primzahlen $p_1, p_2, \ldots, p_k$.

Man bilde die natürliche Zahl $n = p_1 \cdot p_2 \cdots p_k + 1$.

Nach dem Hauptsatz hat n eine Zerlegung welche, o.B.d.A, p_1 enthält.

Damit würde $p_1 \mid n$ und somit auch $p_1 \mid (p_1 \cdot p_2 \cdots p_k + 1)$ gelten.

Da $p_1 \mid (p_1 \cdots p_k)$ führt $p_1 \mid 1$ zum Widerspruch. q.e.d.

Def. Eine *Lineare Diophantische Gleichung* hat die Form

$$a \cdot x + b \cdot y = c$$

mit Konstanten $a, b, c \in \mathbb{Z}$ und Variablen x, y. Gesucht ist eine ganzzahlige Lösung, also Werte für $x, y \in \mathbb{Z}$, welche die Gleichung erfüllen.

Bsp. $65 \cdot x + 20 \cdot y = 15$ hat die Lösung $x = 3$ und $y = -9$, aber auch $x = -1$ und $y = 4$.

Bsp. $3 \cdot x + 6 \cdot y = 3 \cdot (x + 2 \cdot y) = 1$ hat keine Lösung, sonst wäre 3 ein Teiler von 1.

Anm. nicht-lineare Diophantische Gleichungen, z.B. $x^2 + y^2 = z^2$.

Satz (Großer Satz von Fermat)

Diophantische Gleichungen $x^n + y^n = z^n$ haben für $n > 2$ keine Lösung.

Satz Die lineare Diophantische Gleichung $a \cdot x + b \cdot y = c$ ist lösbar, gdw. $d = ggT(a,b) \mid c$.

Bew.

Sei $a \cdot x + b \cdot y = c$ lösbar, dann gilt $d \mid a$ und $d \mid b$, somit $d \mid c$.

Umgekehrt gelte $d \mid c$.

Hauptsatz über den ggT (erw. Euklid'scher Alg.) gibt einem $u, v \in \mathbb{Z}$ mit $a \cdot u + b \cdot v = d$.

Vor. zeigt $e = c/d \in \mathbb{Z}$ und Durchmultiplizieren mit e ergibt

$$a \cdot \underbrace{u \cdot e}_{x} + b \cdot \underbrace{v \cdot e}_{y} = d \cdot e = d \cdot \frac{c}{d} = c$$

q.e.d.

Anm. falls (x,y) eine Lösung ist, so auch (x',y'), mit

$$x' = x + \frac{b}{d} \cdot t, \qquad y' = y - \frac{a}{d} \cdot t, \qquad \text{für alle } t \in \mathbb{Z}$$

Def. Zu $a, b \in \mathbb{Z}$ und $m \in \mathbb{N}$ ist a *kongruent zu* b *modulo* m, wenn $m \mid (a-b)$.

Anm. Wenn a kongruent zu b modulo m ist, so schreibt man $a \equiv b \bmod m$, $a \equiv_m b$, oder auch $a \equiv b\ (m)$. Oftmals wird auch "$=$" statt "$\equiv$" verwendet, und sagt einfach "gleich modulo m" statt "kongruent modulo m".

Bsp. 3660 Sekunden später ist zur gleichen Minute wie in einer Minute, da 3660 kongruent zu 60 ist modulo 3600 (1 Stunde) ist.

Bsp. 31 Stunden später als jetzt in Linz (CEST) ist zur gleichen Stunde wie vor zwei Stunden in San Francisco (PDT), da $31 - 9 = 22$ kongruent zu -2 modulo 24 ist.

Satz $a \equiv b \bmod m$, gdw. die positiven Reste von a und b bei Division durch m gleich sind.

Bew. Ansatz $a = m \cdot p + r, \quad b = m \cdot q + s$ und o.B.d.A. $0 \leq r, s < m$ etc.

Einfache Modulo Rechenregeln

Reflexivität: $a \equiv a \bmod m$

Symmetrie: wenn $a \equiv b \bmod m$, dann auch $b \equiv a \bmod m$

Transitivität: wenn $a \equiv b \bmod m$, und $b \equiv c \bmod m$ dann auch $a \equiv c \bmod m$

$\Rightarrow$ also ist "$\equiv$" eine *Äquivalenzrelation*

Klassen-Einteilung geschieht nach den positiven Resten bei Division mit m

Kongruenz bezüglich der Rechenarten:

wenn $a \equiv c \bmod m$ und $b \equiv d \bmod m$,

dann gilt auch $a+b = c+d \bmod m$, $a-b = c-d \bmod m$, $a \cdot b = c \cdot d \bmod m$

und $p(a) \equiv p(b) \bmod m$ falls $p(x) = c_n \cdot x^n + \cdots + c_1 \cdot x + c_0$

Satz Ist $k \cdot a \equiv k \cdot b \bmod m$ und $d = ggT(k,m)$, so gilt $a \equiv b \bmod (m/d)$.

Bsp. Gilt $2 \cdot a \equiv 2 \cdot b \bmod 256$ für zwei Bytes a und b so ist $a \equiv b \bmod 128$, d.h. $a \equiv b \bmod 256$ oder $a \equiv b + 128 \bmod 256$. Falls $3 \cdot a \equiv 3 \cdot b \bmod 256$ dann sind a und b sicherlich gleich.

Satz Die Kongruenz $a \cdot x \equiv b \bmod m$ ist lösbar mit $x \in \mathbb{Z}$ gdw. $d = ggT(a,m) \mid b$.

Bew. Ist die Kongruenz lösbar, dann gilt $m \mid a \cdot x - b$.

Es gibt somit ein $y \in \mathbb{Z}$ mit $a \cdot x - b = m \cdot y$, woraus sich $b = a \cdot x - m \cdot y$ ergibt.

Diese Lineare Diophantische Gleichung ist lösbar gdw. $d = ggT(a,-m) = ggT(a,m) \mid b$.

Satz Ist die Kongruenz lösbar, so hat sie d Lösungen.

Bew. Aus Anm. zum Satz über Lösbarkeit Linearer Diophantischer Gleichungen sind

$$x' \equiv x + \frac{m}{d} \cdot t \bmod m$$ auch Lösungen, für alle $t \in \mathbb{Z}$

Korollar Die Kongruenz $a \cdot x \equiv b \bmod m$ mit $ggT(a,m) = 1$ ist eindeutig lösbar.

Korollar Ist p prim, dann ist die Kongruenz $a \cdot x \equiv b \bmod p$ eindeutig lösbar.

Def. Wenn $a \cdot x \equiv 1 \bmod m$ lösbar, so heißt a eine *Einheit* modulo m.

Fakt Die Menge der Einheiten modulo m bildet eine Gruppe bezüglich Multiplikation.

Bsp. Die Menge der Einheiten modulo 10 ist $\{1,3,7,9\}$.

Bsp. Menge der Einheiten modulo Primzahl p besteht aus allen Restklassen außer 0.

Satz (Kleiner Satz von Fermat) Ist p prim, so gilt $a^p \equiv a \bmod p$, für alle $a \in \mathbb{Z}$.

Korollar Ist p eine Primzahl und sind a und p relativ prim, dann ist $a^{p-1} \equiv 1 \; mod p$.

Bew. Korollar folgt aus dem Kleinen Fermat und dem ersten Korollar auf dieser Folie.

Satz (Kleiner Satz von Fermat) Ist p prim, so gilt $a^p \equiv a \bmod p$, für alle $a \in \mathbb{Z}$.

Wir zeigen den Satz nur für $a \geq 0$, der Rest als Übung!

Induktionsanfang: $a = 0$ trivial.

Induktionsschritt:

$$(a+1)^p \equiv \sum_{i=0}^{p} \binom{p}{i} a^i \equiv a^p + \underbrace{p \cdot a^{p-1} + \binom{p}{p-2} \cdot a^{p-2} + \cdots + p \cdot a}_{\equiv 0 \bmod p} + 1 \equiv a+1 \bmod p$$

In der ersten Gleichung verwendet man den Binomial-Satz.

In der zweiten Gleichung kürzt man Summen-Terme, die kongruent 0 modulo p sind.

In der dritten Gleichung benützt man die Induktionsannahme.

Anm. damit kann man immer "modulo $p-1$" Exponenten berechnen.

Satz Es gelte $x \equiv c_i \bmod m_i$ für $i = 1, \ldots, n$, wobei die m_i relativ prim sind, dann gibt es eine eindeutige Lösung x modulo $m = \prod_{i=1}^{n} m_i$.

Beweis konstruktiv am Beispiel $x \equiv 1 \bmod 2$, $x \equiv 3 \bmod 5$, und $x \equiv 5 \bmod 7$.

Ansatz $M_i = \prod_{j \neq i} m_j$.

Bestimme y_i mit $M_i \cdot y_i \equiv 1 \bmod m_i$ durch den erweiterten Euklid'schen Algorithmus.

Also $(5 \cdot 7) \cdot 1 \equiv 1 \bmod 2$, $(2 \cdot 7) \cdot 4 \equiv 1 \bmod 5$ $(2 \cdot 5) \cdot 5 \equiv 1 \bmod 7$

Nun bildet man die Linearkombination von den so gefundenen Inversen y_i von dem Produkt M_i der "anderen" m_j modulo m_i mit $M_i \cdot c_i$ wie folgt

$$x \equiv \sum_{i=1}^{n} c_i \cdot M_i \cdot y_i \mod m$$

Also $x \equiv 1 \cdot (5 \cdot 7) \cdot 1 + 3 \cdot (2 \cdot 7) \cdot 4 + 5 \cdot (2 \cdot 5) \cdot 5 = 35 + 168 + 250 \equiv 453 \equiv 33 \bmod 70$

und tatsächlich $33 \equiv 1 \bmod 2$, $33 \equiv 3 \bmod 5$, und $33 \equiv 5 \bmod 7$.

Anwendung: RSA Public Key Verschlüsselungs-Verfahren

[RivestShamirAdleman'77]

Nachricht a kodiert als große natürliche Zahl.

Zwei große (≥ 500 Bit) geheime Primzahlen p, q. Veröffentliche nur ihr Produkt $p \cdot q$.

Erzeuge öffentlichen Schlüssel e relativ prim zu $p-1$ und $q-1$.

Berechne privaten Schlüssel d als Inverses von e modulo $(p-1)\cdot(q-1)$.

Verschlüssle a als a^e modulo $p \cdot q$ und verschicke a^e über unsicheren Kanal.

Entschlüssle durch $(a^e)^d$ modulo $p \cdot q$,

denn $a^{e \cdot d} \equiv a \bmod p$ und $a^{e \cdot d} \equiv a \bmod q$

nach kleinem Fermat und $e \cdot d = 1 + k \cdot (p-1) \cdot (q-1)$.

Rest folgt mit CRT.

Annahme: $a, e, p \cdot q$ öffentlich, p, q, d nicht.

Nachricht a signieren mit Signatur $s = a^d$. Beides wird veröffentlicht.

Gültigkeit der Signatur: $s^e = a$?

Out-of-the-Box Randomization Attack

Implementierungen rechnen s mit CRT aus: (da ungefähr viermal schneller)

$$a^d \equiv a^{d_p} \bmod p \quad \text{mit} \quad d \equiv d_p \bmod (p-1)$$

$$a^d \equiv a^{d_q} \bmod q \quad \text{mit} \quad d \equiv d_q \bmod (q-1)$$

$$s \equiv a^d \equiv c_1 \cdot M_1 \cdot y_1 + c_2 \cdot M_2 \cdot y_2 \equiv a^{d_p} \cdot q \cdot y_1 + a^{d_q} \cdot p \cdot y_2 \bmod (p \cdot q)$$

Störe Berechnung von a^{d_q} (ersetze a^{d_q} durch $E \neq a^{d_q}$, z.B. durch Hitze/Strahlung):

$$t = a^{d_p} \cdot q \cdot y_1 + E \cdot p \cdot y_2, \qquad t^e = a \bmod p, \qquad t^e \neq a \bmod q, \qquad ggT(t^e - a, p \cdot q) = p$$