RSA-Verschlüsselung

und weitere Anwendungen
elementarer Zahlentheorie
auf die **Kalenderrechnung**
Angewandte Mathematik für das Lehramt an Grund- und Mittelstufe sowie an Sonderschulen
Bodo Werner WiSe 02/03

1 Einführung

Die RSA-Verschlüsselung von Nachrichten zählt zur *Kryptographie*[1], die in unserem Leben eine bedeutende Rolle spielt, z.B. bei der Identifizierung einer EC-Karte mit Hilfe ihrer Geheimzahl, aber auch bei der Übertragung der Kreditkarten-Daten über das Internet oder bei der Übersendung von Nachrichten über Emails, Kurznachrichten etc. Eine sehr gute Einführung auch aus geschichtlicher Sicht geben die Bücher von A. Beutelspacher (Geheimsprachen Beck-Verlag 1997, Kryptologie Vieweg 1991) und R. Kippenhahn (Streng Geheim! Rowohlt, Verschlsselte Botschaften: Geheimschrift, Enigma und Chipkarte; Rowohlt, 1998).
Hier will ich nicht eine allgemeine Einführung versuchen, sondern ein ganz spezielles Verfahren beschreiben, das auf relativ einfache zahlentheoretische Konzepten beruht.

Übersichten im Internet:
Verschlüsselte Botschaften[2] (Recherchekompass)
Kryptographie im Verlauf der Geschichte[3] (Huynh, Thi Phuong-Thao)

2 Das Schema: magisches Tripel

Das RSA-Verfahren wurde 1977 von Ron **R**ivest, Adi **S**hamir und Leonard **A**dleman erfunden.

[1] krypto=geheim, graphie=schreiben
[2] http://home.nordwest.net/hgm/krypto/
[3] http://rhlx01.rz.fht-esslingen.de/projects/krypto/krypt_gesch/krypt_gesch.html

Wir identifizieren eine zu verschlüsselnde Botschaft (oder einen Teil hiervon) mit einer Zahl $m \in \mathbb{N}$ [4]. Um die Art und Weise, mit der eine Botschaft mit einer Zahl m gleichgesetzt wird, kümmern wir uns nicht. Dies ist zwar auch eine Art „Codierung“, soll aber nichts mit unserer „Verschlüsselung“ zu tun haben.
Mit dem RSA-Verfahren wird m zu einer anderen Zahl $f(m) := m' \in \mathbb{N}$ *verschlüsselt*. Die *Entschlüsselung* hat m' als Eingabe und (hoffentlich) m wieder als Ausgabe. Mathematisch handelt es sich bei der Verschlüsselung um eine Funktion $f : \mathbb{N} \to \mathbb{N}$ und bei der Entschlüsselung um deren Umkehrfunktion. Das reizvolle bei der folgenden Technik ist, dass die Funktion f „öffentlich“ ist, dass deren Umkehrfunktion aber in vertretbarer Zeit nicht berechnet werden kann, es sein denn, man verfügt über eine gewisse Information. Dies hat zur Folge, dass auch „sehr große Zahlen“[5] m und m' auftreten sollten.

Der Schlüssel wird durch ein *magisches Tripel* (d,e,n) dreier natürlicher Zahlen d, e und n definiert. Dabei ist

$$n = p \cdot q$$

das Produkt zweier großer Primzahlen. Ferner muss

$$d \cdot e \bmod (p-1)(q-1) = 1 \tag{1}$$

gelten.
(e, n)[6] ist der *öffentliche Schlüssel*, verschlüsselt wird mittels

$$m \to m' := m^e \bmod n.$$

(d, n) ist der *private Schlüssel*, entschlüsselt wird mittels

$$m' \to \tilde{m} := (m')^d \bmod n.$$

Die Primzahlen p und q dienen nur der Bestimmung des Schlüssels, sie können aus Sicherheitsgründen hernach vernichtet werden. Nur d muss geheim gehalten werden. In der Praxis müssen p und q und damit $n = pq$ sehr groß sein, damit die Menge aller möglichen Botschaften[7] hinreichend groß ist.
Man kann im Internet zu zwei Primzahlen p und q ein **magisches Tripel erzeugen**[8] (KEY GENERATION PAGE). Für $p = 5, q = 11$ z.B. erhält man das Tripel (3, 27, 55).

[4] Der Teil kann z.B. ein Block aus drei alphanumerischen Zeichen sein. Jedes dieser drei Zeichen kann gemäß des ASCII-Codes durch eine 8-stellige Dualzahl dargestellt werden (Die letzte Ziffer ist eine Prüfziffer). Die drei Dualzahlen werden danach zu einer 24-stelligen Dualzahl zusammengestellt und diese mit einer Zahl $m \leq 2^{25}$ identifiziert

[5] Die Größe misst man mit der Anzahl von Bits im Dualsystem. Bei 128 Bits – die stärkste z.Zt. verfügbare Verschlüsselung – gibt es $2^{128} \approx 10^{38}$ (wegen $2^{10} \approx 10^3$) Zahlen

[6] **e** kommt von **e***ncrypt*, **d** von **d***ecrypt*

[7] Alle mögliches Reste bei Division durch $n = pq$. Von denen gibt es genau n

[8] http://www.orst.edu/dept/honors/makmur/alice.html

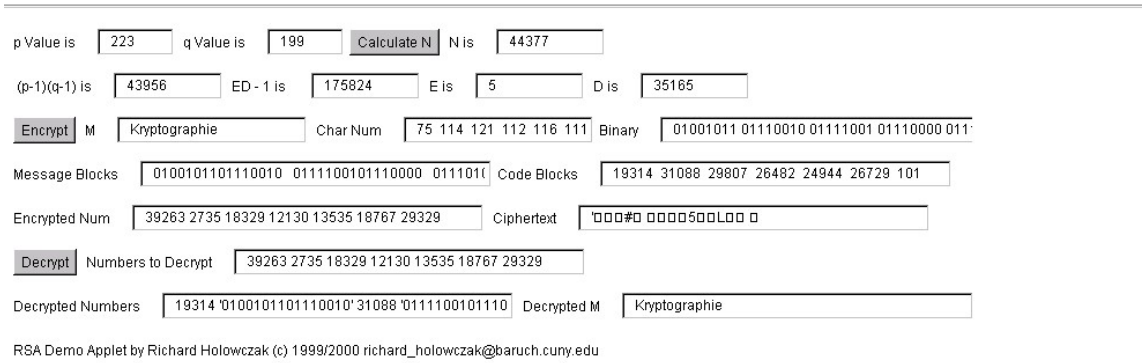

Abbildung 1: RSA-Applet von R.Holowczak

Die Public Sharware PVG (Pretty Good Privacy) zum Verschlüsseln von Emails bedient sich dieses Schemas.

Internet:
Asymmetrische/ moderne Kryptographie - Ein interaktiver Überblick[9] (V. Tiemann, Uni Bielefeld)
RSA-Algorithmus [10] (FH Flensburg)
RSA-Algorithmus[11]. Diese Seite ist Teil eines **Kryptologieprojektes**[12] der FH Esslingen.
Kryptographie: Wie geheim ist geheim?[13], ein Skript von Judith Plümer (FH Osnabrück), in dem auch der RSA-Algorithmus erläutert wird.
RSA-Algorithmus[14] (Pro-Privacy)

Es gibt im Internet ein **(englischsprachiges) RSA-Applet**[15] (Richard Holowczak, Barauch College, USA), in dem man p, q und e eingeben kann. Es wird d ermittelt. Ferner kann jede Botschaft im ASCII-Format in Zweier-Blöcken ver- und wieder entschlüsselt werden. Um 128 Zeichen ASCII-Zeichen darzustellen, muss $n \approx 40.000$[16]

Ansonsten gibt es eine unübersehbare Fülle von Internetinformationen über Kryptographie.

Im folgenden soll begründet werden, warum $\tilde{m} = m$, d.h., warum man wirklich von Entschlüsselung sprechen kann. Dies erfordert etwas Zahlentheorie. Des weiteren werden Fragen

[9] http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material_spezifisch/statalg00/rsa/rsa.html
[10] http://www.iti.fh-flensburg.de/lang/algorithmen/code/krypto/rsa.htm
[11] http://www-stud.fht-esslingen.de/projects/krypto/dig_unt/dig_unt-2.html
[12] http://www-stud.fht-esslingen.de/projects/krypto/dig_unt/dig_unt-2.html
[13] http://www.mathematik.uni-osnabrueck.de/staff/phpages/pluemerj/krypto/
[14] http://www.pro-privacy.de/pgp/tb/de/rsa.htm
[15] http://cisnet.baruch.cuny.edu/holowczak/classes/9444/rsademo/
[16] wegen eines Prüfsbit benötigt man für 2 Zeichen eine Zahl $m \leq\approx 40.000$

angesprochen, die sich auf das Auffinden magischer Tripel beziehen und die Algorithmen zur Ver- und Entschlüsselung betreffen.

3 Etwas Zahlentheorie

Ganz zentral ist offensichtlich die **modulo**-Rechnung – sowohl zum Nachweis der Eigenschaft eines magischen Tripels als auch zur Durchführung der Ver- und Entschlüsselung des RSA-Algorithmus selbst.

3.1 modulo-Rechnung

Zur Erinnerung: $m \bmod n$ ist der Rest, wenn man m durch n teilt. In diesem Sinne ist „mod“ ein arithmetischer Operator wie auch die Addition und Multiplikation: zwei Zahlen m und n wird mittels $m \bmod n$ eine dritte zugeordnet, die als Rest bei der Division durch n in $\mathbb{Z}_n = \{0, 1, 2, ..., n-1\}$ liegt. Auf $\mathbb{Z}_n$ kann man auf ganz natürliche Weise eine modulo-Addition $+_n$ und eine modulo-Multiplikation $\cdot_n$ definieren, indem man erst wie üblich addiert bzw. multipliziert und dann zu den Resten bei Division durch n übergeht.
Beispiele:

$$7 +_{11} 10 = 17 \bmod 11 = 6, \quad 7 \cdot_{11} 10 = 70 \bmod 11 = 4.$$

Nun ist $\mathbb{Z}_n$ eine abelsche Gruppe bezgl. der modulo-Addition $+_n$ und ein Ring bzgl. der modulo-Multiplikation $\cdot_n$, sogar ein Körper, wenn n eine Primzahl ist. Dabei ist es wesentlich, dass man die Reihenfolge von Addition (Multiplikation) und modulo-Rechnung beliebig wählen kann. Z.B. ist

$$(8 \cdot 27) \bmod 11 = (8 \bmod 11)(27 \bmod 11) \bmod 11 = (8 \cdot 5) \bmod 11 = 40 \bmod 11 = 7.$$

Allgemein gelten die modulo-Gesetze

$$(m_1 + m_2) \bmod n = ((m_1 \bmod n) + (m_2 \bmod n)) \bmod n,$$

$$(m_1 \cdot m_2) \bmod n = ((m_1 \bmod n) \cdot (m_2 \bmod n)) \bmod n$$

für alle $m_1, m_2, n \in \mathbb{N}$.
Dies hat auch Auswirkungen auf die Potenzierung (s. Kap. 3.3). So gilt

$$7^4 \bmod 6 = (7 \bmod 6)^4 \bmod 6 = 1,$$

allgemein

$$m^r \bmod n = ((m \bmod n)^r) \bmod n \tag{2}$$

für alle $r, m, n \in \mathbb{N}$.

Ein weiterer wichtiger Begriff ist der der *Kongruenz*: Zwei Zahlen m und n heißen **kongruent modulo** p genau dann, wenn $m \bmod p = n \bmod p$. Man schreibt

$$m \equiv n \bmod p.$$

Hierbei handelt es sich um eine **Äquivalenzrelation**, es gibt also **Äquivalenzklassen**. Jedes Element („Rest“) von $\mathbb{Z}_n$ entspricht einer Restklasse. Daher kann man die Operationen $+_n$ und $\cdot_n$ auch als „Restklassenoperationen“ auffassen.

3.2 Inverse bei der modulo-Multiplikation

Für das Verständnis des folgenden ist die folgende Frage hilfreich: Wann besitzt $r \in \mathbb{Z}_n$ ein Inverses $s \in \mathbb{Z}_n$ bzgl. der modulo-Multiplikation $\cdot_n$? Oder etwas allgemeiner: wann gibt es zu $r \in \mathbb{N}$ ein $s \in \mathbb{N}$ mit $(r \cdot s) \bmod n = 1$? Dieser Frage soll in den Übungen nachgegangen werden. Hier wollen wir nur anregen, $\mathbb{Z}_7$ für die Primzahl $n = 7$ dahingehend zu untersuchen, dass alle Elemente bis auf die Null eine solche Inverse besitzen.
Man beachte, dass die Eigenschaft (1) eines magischen Tripels gerade besagt, dass d das Inverse zu e in $\mathbb{Z}_{(p-1)(q-1)}$ ist (und umgekehrt).
Wir halten hier ohne Beweis fest:
r und n sind genau dann prim zueinander, wenn es ein s gibt mit $(r \cdot s) \bmod n = 1$. In einem solchen Fall kann s als Potenz von r gewählt werden.

3.3 Potenzen in der modulo-Rechnung

Im RSA-Verfahren hat man es sowohl bei der Ver- als auch bei der Entschlüsselung mit Potenzen $m^r \bmod n$ zu tun[17], z.B. mit $17^{27} \bmod 55$ beim magischen Tripel (3,27,55). Es wäre nun töricht, erst die immens große Zahl 17^{27} auszurechnen und dann den Rest bei Division durch 55 zu ermitteln. Stattdessen wird man *nacheinander*

$$(17^2) \bmod 55 = 289 \bmod 55 = 14, \quad (17^3) \bmod 55 = (17 \cdot (17^2 \bmod 55)) \bmod 55 \tag{3}$$

$$= (17 \cdot 14) \bmod 55 = 238 \bmod 55 = 18, \tag{4}$$

allgemein

$$17^{k+1} \bmod 55 = (17 \cdot (17^k \bmod 55)) \bmod 55,$$

bzw. noch allgemeiner

$$m^{k+1} \bmod n = (m \cdot (m^k \bmod n)) \bmod n,$$

benutzen. Es ist sehr einfach, einen solchen *rekursiven* Rechenschritt mit Hilfe von *Schleifen* zu programmieren, inbesondere, wenn die Programmiersprache einen *arithmetischen „modulo-Operator“* vorsieht, der der den Rest ergibt, wenn man zwei Zahlen durcheinander dividiert. In Java macht dies gerade der %-Operator. Ein Java-Programm zur Berechnung von $ms = m^d \bmod n$ kann so aussehen:

[17] $r = e$ oder $r = d$

```
//Berechnung vom m^d % n:
int ms=1;
for (int k=1;k<=d;k++) {ms = (ms*m) % n};
```

3.4 Größte gemeinsame Teiler (ggT)

Was der **größte gemeinsame Teiler** (ggT(m,n)) zweier Zahlen m und n ist, dürfte klar sein. Zwei Zahlen m und n heißen **prim** zueinander, wenn ggT(m,n)=1. Eine **Primzahl** ist zu jeder anderen Zahl prim.
Es gibt einen wunderbaren Algorithmus zur Berechnung von ggT(m,n), den **Euklidischen Algorithmus**: Hierzu sei $m > n$. Es ist sinnvoll, $m_0 = m$ und $m_1 = n$ zu setzen. Nun sein $m_2 := m_0 \bmod m_1$. Falls $m_2 = 0$, so gilt $\mathrm{ggT}(m_0, m_1) = m_1$. Sonst setze $m_3 := m_1 \bmod m_2$. Falls $m_3 = 0$, so ist $\mathrm{ggT}(m_1, m_2) = m_2$, sonst setze $m_4 := m_2 \bmod m_3$. Die entscheidende Beobachtung ist, dass $\mathrm{ggT}(m_1, m_2) = \mathrm{ggT}(m_0, m_1)$, allgemein[18]

$$\mathrm{ggT}(m, n) = \mathrm{ggT}(n, m \bmod n).$$

So kann man fortfahren, bis man einen Index k findet mit $m_k = 0$. Dann ist $\mathrm{ggT}(m_0, m_1) = m_{k-1}$.
Beispiel: $m_0 = 27, m_1 = 15$. Dann ist $m_2 = 27 \bmod 15 = 12, m_3 = 15 \bmod 12 = 3$ und $m_4 = 12 \bmod 3 = 0$. Also gilt $27 \bmod 15 = 3$.

3.5 Kleiner Satz von Fermat

Von zentraler Bedeutung zum Verständnis des magischen RSA-Tripels ist der

Kleine Satz von Fermat: Für alle Primzahlen p und alle zu p primen $m \in \mathbb{N}$ gilt

$$m^{p-1} \bmod p = 1.$$

Beweis: Da $\mathbb{Z}_p$ ein Körper ist, ist $\{1, 2, .., p-1\}$ zusammen mit der Modulo-Multiplikation $\cdot_p$ eine abelsche Gruppe[19]. Die von m erzeugte Unterruppe hat Ordnung r, wenn r die kleinste Zahl mit $m^r \bmod p = 1$ ist. Die Ordnung einer Untergruppe teilt die Ordnung der Obergruppe, also teilt r die Zahl $p-1$, und es gilt erst recht die Behauptung, da wegen (2)

$$m^{rs} \bmod p = (m^r)^s \bmod p = (m^r \bmod p)^s \bmod p = 1 \bmod p = 1.$$

Der Beweis ist noch kürzer, wenn man $g^{|G|} = e$ für endliche Gruppen G mit Ordnung $|G|$, einem $g \in G$ und neutralem Element e benutzt.

[18] Ein Teiler von m und n teilt stets $m \bmod n$. Ein Teiler von n und $m \bmod n$ teilt auch stets m

[19] insbesondere hat jedes Element ein Inverses. Das kann man für $p = 7$ überprüfen, indem man eine Verknüpfungstafel erstellt

Abbildung 2: Französische Briefmarke

Wem die Körper- bzw. Gruppeneigenschaften zu undurchsichtig sind, kann auch so argumentieren: Wenn man alle Potenzen $m^j \bmod p, j = 1, 2, ...$ betrachtet, muss es ein (kleinstes) $r > 0$ geben mit $m^r \bmod p = 1$ (dann ist $m^{r-1} \bmod p$ das Inverse von m in $\mathbb{Z}_p$). Denn da es nur endlich viele verschiedene dieser Potenzen gibt, gibt es $k > j$ mit $m^k \bmod p = m^j \bmod p$, also

$$0 = (m^k - m^j) \bmod p = m^j(m^{k-j} - 1) \bmod p = (m^j \bmod p) \cdot ((m^r - 1) \bmod p),\ r := k - j.$$

Da $m^j \bmod p \neq 0$ (warum?), haben wir die Behauptung gezeigt.
Als letztes muss man zeigen, dass r ein Teiler von $p - 1$ ist. Nun, die von m erzeugte Gruppe hat gerade r Elememente, sie ist eine Untergruppe von $(\{1, 2, ..., p - 1\}, \cdot_p)$, die Ordnung einer Untergruppe teilt die Ordnung der Gruppe[20].

Als **Korollar des kleinen Satzes von Fermat** haben wir: Seien p, q zwei Primzahlen und sei m prim zu p und prim zu q. Dann gilt

$$m^{(p-1)(q-1)} \bmod (p \cdot q) = 1$$

Beweis: Wegen $m^{(p-1)(q-1)} = (m^{p-1})^{q-1} = (m^{q-1})^{p-1}$ gilt nach dem kleinen Fermat

$$m^{(p-1)(q-1)} \bmod p = 1, \quad m^{(p-1)(q-1)} \bmod q = 1.$$

Es gibt also natürliche Zahlen r, s mit

$$m^{(p-1)(q-1)} = r \cdot p + 1, \quad m^{(p-1)(q-1)} = s \cdot q + 1.$$

Daher ist q ein Teiler von r (und p ein Teiler von s), woraus sofort die Behauptung abgelesen werden kann.

[20] Dies ist nicht tiefliegend

3.6 RSA-Satz

Jetzt können wir die Basis-Eigenschaft des RSA-Algorithmus zeigen:
RSA-Satz: Sei (d, e, n) ein magisches Tripel, also insbesondere sei $n = p \cdot q$. Sei die „Botschaft“ m prim zu p und q. Dann gilt für $m' := m^e \bmod n$, dass

$$m = (m')^d \bmod n.$$

Beweis: Zu zeigen ist

$$(m^e \bmod n)^d \bmod n = m,$$

bzw.

$$m^{e \cdot d} \bmod n = m.$$

Nun gilt wegen $d \cdot e \bmod (p-1)(q-1) = 1$, dass $d \cdot e = k(p-1)(q-1) + 1$ für ein $k \in \mathbb{N}$. Also gilt

$$m^{d \cdot e} \bmod n = (m^{k(p-1)(q-1)} \cdot m) \bmod n = (m^{k(p-1)(q-1)} \bmod n) \cdot (m \bmod n) = 1 \cdot m = m$$

wegen des Korollars aus dem kleinen Fermat, das hier

$$m^{k(p-1)(q-1)} \bmod n = 1$$

liefert. Man beachte, dass wir dauernd von dem modulo-Prinzip Gebrauch gemacht haben, nach dem es nämlich gleich ist, ob man erst modulo rechnet und dann multipliziert oder umgekehrt.

Bemerkung: Der kleine Satz von Fermat kann verallgemeinert werden, wenn man die **Eulersche Funktion**

$$\varphi(n) := \text{Anzahl der zu } n \text{ teilerfremden Zahlen}$$

einführt. Für sie kann man den *Eulerschen Satz*

$$m^{\varphi(n)} \, mod \, n = 1 \quad \text{für alle } m, \text{ die prim zu } n \text{ sind}$$

zeigen. Wegen $\varphi(p) = p - 1$ und $\varphi(pq) = (p-1)(q-1)$ folgt hieraus der kleine Fermat und sein Korollar. Der Beweis des Eulerschen Satzes ist einfach, wenn man die endliche Gruppe $\mathbb{Z}_n^\star := \{k \in \mathbb{N} : ggT(k, n) = 1, k < n\}$ (zusammen mit der modulo-Multiplikation) betrachtet: ihre Ordnung ist gerade $\varphi(n)$, der Satz folgt aus dem gruppentheoretischen Resultat $g^{|G|} = e$ für alle $g \in G$ und für alle endlichen Gruppen G (mit neutralem Element e).

3.7 Weitere Anwendungen der modulo-Rechnung

Wenn wir die Nachkommastellen einer Dezimalzahl $x > 0$ wissen wollen, müssen wir $x \bmod 1$ rechnen. Hier ist i.a. $x \notin \mathbb{N}$. Wenn man die Position des großen Zeigers einer Uhr nach einer gewissen Anzahl von Minuten wissen will, muss man „modulo 60“ rechnen. Überall da, wo Kreisbewegungen bzw. periodische Abläufe vorkommen, ist die modulo-Rechnung gefragt. Unser Leben auf der Erde wird im wesentlichen durch drei solcher Vorgänge geprägt: die Rotation der Erde um ihre Achse (Tag), der Umlauf des Mondes um die Erde mit den verschiedenen Mondphasen (Monat) sowie der Umlauf der Erde um die Sonne (Jahr). Hinzu kommt noch ein Minuten-, Stunden- und Wochenrhythmus. Will man z.B. wissen welcher Wochentag in 1500 Tagen ist, so wird man einfach nur modulo 7 rechnen müssen.

3.7.1 Kalender

Frühe Kulturen benutzten oft einen Mondkalender, weil sich der Mond gut beobachten lässt und der Mondzyklus relativ kurz ist. Seit der Einführung der Zeitrechnung basieren Kalender auf dem Sonnenumlauf, der **365.242199 Tage** lang ist (tropisches Jahr, bezieht sich auf den Durchgang der Erde durch den Frühlingspunkt). Die alten Ägypter wussten, dass ein Jahr ungefhr 365 1/4 Tage dauert. Mit der Einführung des Julianischen Kalenders durch Julius Cäsar im Jahre 45 v. Chr. wurde daher die Jahreslänge auf 365,25 Tage festgelegt. Damit der Jahresanfang sich nicht im Laufe der Jahrhunderte in eine andere Jahreszeit verschiebt, wird alle vier Jahre im Februar ein Schalttag eingefügt (sonst würde nach 400 Jahren Weihnachten in den Spätsommer fallen). Die so festgelegte Jahreslänge ist um 0,007801 Tage zu lang. In 1600 Jahren wächst dieser Unterschied auf rund 12 Tage an (Ostern verschiebt sich immer mehr in zum Frühling). Daher hat im Jahr 1582 Papst Gregor XIII den Gregorianischen Kalender eingeführt. Die zehn Tage zwischen dem 4. und 15. Oktober 1582 entfielen. Für den neuen Kalender wurde eine Jahreslänge von 365,2425 Tagen zugrunde gelegt. Daher ist eine andere Schaltjahrregelung nötig. Es gilt: alle durch vier teilbaren Jahre sind Schaltjahre. Schalttage entfallen in den durch 100 teilbaren Jahren, ausgenommen sind die durch 400 teilbaren Jahre. Wann war das letzte, wann ist das nächste Schaltjahr?
Das Gregorianische Jahr ist um 0,0003 Tage oder 26 Sekunden zu lang. Erst anch 3.300 Jahre wächst sich dieser Unterschied auf einen Tag an, ein Schaltjahr muss dann entfallen.

Würde es keine Schaltjahre geben, also das Jahr aus genau 365 Tagen bestehen, wäre es eine leichte Aufgabe, den Wochentag zu einem vorbestimmten Datum zu bestimmen. Für den Gregorianischen Kalender hat ein Geistlicher (Christoph Zeller 1885) die **Zellersche Formel** aufgestellt:

Wochentag = (Tag + (13*Monat - 1) div 5 + Jahr + Jahr div 4 - Jahr div 100 + Jahr div 400) mod 7

„div“ ist die gannzahlige Division ohne Betrachtung des Restes, z.B. ist 17 div 5 = 3.

Die Monatszählung beginnt dabei mit Monat=1 fr den März. Januar und Februar sind Monat 11 bzw. 12 des Vorjahres, der Schalttag wird also fr eine einfachere Berechnung an das Jahresende gelegt. Als Wochentag ergibt sich 0 für einen Sonntag, 1 fr einen Montag,..., 6 fr einen Samstag[21]. Beispiel: Auf welchen Wochentag fällt der 24. Dezember 2222? Antwort: Der Dezember ist Monat 10, also (24+(129 div 5) +2222 +555 -22) mod 7= (24 + 25 + 2777 -1) mod 7 = 4, also ein Donnerstag.

Internet:
Eine schier unerschöpfliche Quelle: **Kalender - Computus (Links zum Thema Kalender)** [22]

Ewiger Kalender 1583-2199[23] (H. Greschner)
Kalender in der Geschichte[24]

3.8 Teilbarkeitskriterien

Wegen $10 \bmod 3 = 1$ gilt wegen (2) $10^n \bmod 3 = 1$, entsprechend, wenn man 3 durch 9 ersetzt. Hieraus folgt leicht die 3er- und 9er-Regel. So gilt z.B.

$$1442 \bmod 3 = (1 \cdot 10^3 + 4 \cdot 10^2 + 4 \cdot 10 + 2) \bmod 3 =$$

$$(1 \cdot (10^3 \bmod 3) + 4 \cdot (10^2 \bmod 3) + 4 \cdot (10 \bmod 3) + 2) \bmod 3 =$$

$$(1 + 4 + 4 + 2) \bmod 3.$$

Die nächst einfachere ist die 11er-Regel, die aus $10 \bmod 11 = -1 \bmod 11$, $100 \bmod 11 = 1$ und danach $10^n \bmod 11 = 1$, falls n gerade und -1 sonst, sodass man eine alternierende Quersumme nehmen muss.
Dann gibt es noch eine kompliziertere 7er-Regel, die auf $1000 \bmod 7 = -1$ beruht. Man unterteile daher die Zahl in 3er-Blöcke, die man alternierend gewichtet. Z.B. 12208 führt auf 208-12=196, welche durch 7 teilbar ist.

Internet
Teilbarkeitskriterien Mathe-Planet

4 Praktische Anmerkungen zum RSA-Algorithmus

Zwei große Primzahlen p und q zu finden und $n = pq$ zu bestimmen, ist kein Problem. Grundsätzlich ist die Faktorisierung großer Zahlen (z.B. der Länge von 128 Bits) in vertretbarer Zeit unmöglich. Daher könnte man $n = pq$ veröffentlichen, ohne dass man befürchten muss, dass p und q (und damit d) bekannt werden.

[21] Seit 1976 beginnt die Woche an einem Montag, vorher an einem Sonntag
[22] http://www.computus.de/kalenderlinks/kalenderlinks.htm
[23] http://home.t-online.de/home/Hgreschner/ewkal.htm
[24] http://www.computus.de/kalenderlinks/kalenderlinks.htm#Kalender in der Geschichte

Als nächstes muss man das Zahlenpaar (e, d) finden, das ja

$$d \cdot e \bmod (p-1)(q-1) = 1 \tag{5}$$

genügen soll. Wie geht das? Notwendigerweise müssen beide Zahlen teilerfremd (prim) zu der geraden Zahl $(p-1)(q-1)$, also ungerade sein. Nun bestimmt man relativ leicht *eine* solche Zahl, z.B. e, die

$$\mathrm{ggT}(e, (p-1)(q-1)) = 1$$

erfüllt. Wie findet man hierzu d mit (5)? Nun, wir müssen ein inverses Element von e bzgl. der modulo-$(p-1)(q-1)$-Multiplikation finden. Da e und $(p-1)(q-1)$ teilerfremd sind, gibt es solch ein Inverses in Gestalt einer Potenz von e. Hierzu gibt es einen effizienten Algorithmus, den *erweiterten Euklidischen Algorithmus*: Seien e und m teilerfremd, also $\mathrm{ggT}(e, m) = 1$. Dies kann man durch den Euklidischen Algorithmus zur Berechnung von $\mathrm{ggT}(e, m)$ feststellen. Diesen lasse man sodann „rückwärts laufen“.
Beispiel: $e = 11, m = 47 : 47 = 4 \cdot 11 + 3, 11 = 3 \cdot 3 + 2, 3 = 1 \cdot 2 + 1$.
Invers: $1 = 3 - 1 \cdot 2, 1 = 3 - 1 \cdot (11 - 3 \cdot 3) = -11 + 4 \cdot 3 = -11 + 4 \cdot (47 - 4 \cdot 11) = 4 \cdot 47 - 17 \cdot 11$.
Nehmen wir beide Seiten modulo 47, so erhalten wir $1 \equiv -17 \cdot 11 \bmod 47 \equiv 30 \bmod 47$. Also ist 30 das Inverse von 11 modulo 47.
Es gibt das Lemma von Bezout , nachdem $\mathrm{ggT}(a, b)$ sich als Linearkombination von a und b darstellen lässt, $\mathrm{ggT}(a, b) = ua + vb$ Ist $\mathrm{ggT}(a, b) = 1$, so ist v das Inverse von b modulo a.
Auch die Potenzierung modulo n kann effizient gemacht werden, indem der Exponent als Dualzahl dargestellt wird.

Beispiel: $p = 7, q = 11$, also $n = 77$. Es ist $(p-1)(q-1) = 60$. Wähle $e = 7$. Nun ist $e^3 = \bmod 60 = 43$ und $e^4 \bmod 60 = 1$. Wähle also $d = 43$.
Nun sei $m = 17$. Dann ist $m' = m^e \bmod n = 17^7 \bmod 77 = 52$ und $(m')^d \bmod n = 52^{43} \bmod 77 = 17$.

Internet:
Erweiterter Euklidischer Algorithmus[25] (FH Flensburg). Hier findet man auch weitere zahlentheoretische Grundlagen des RSA-Algorithmus. Desgleichen bei **Die algorithmischen Grundlagen des RSA-Algorithmus** [26] (Klaus Pommerening, Uni Mainz)

4.1 Anwendungen

In der Einführung waren wir davon ausgegangen, dass mit einem öffentlichen Schlüssel verund mit einem privaten Schlüssel entschlüsselt wird. Dies könnte z.B. sinnvoll sein, wenn man dem finanzamt seine Steuererklärungen schicken will: Das Finanzamt vergibt einen öffentlichen Schlüssel an alle, selbst behütet es den privaten Schlüssel.

[25] http://www.iti.fh-flensburg.de/lang/algorithmen/code/krypto/euklid.htm
[26] http://www.uni-mainz.de/p̃ommeren/Kryptologie/Asymmetrisch/1_RSA/rsa.pdf

Dabei kann das Finanzamt sich aber nicht sicher sein, wer die Unterlagen geschickt hat. Hierzu bedarf es einer *elektronischen Signatur*. Nehmen wir mal ein Beispiel, wo die Nachricht selbst ruhig bekannt sein darf, aber eine Signatur dennoch notwendig ist:
Ein Kunde A eines E-Commerce-Unternehmens U will eine Bestellung im Internet aufgeben. U muss sich vergewissern, dass A die Bestellung aufgegeben hat (Signatur) und dass das Bestellformular nicht nachträglich verändert wurde. A hat einen privaten, U einen zugehörigen öffentlichen Schlüssel. Nun bildet A gewisse *Prüfsummen (Hash-Funktionen)* des Dokuments (als Ergebnis erhält er eine Zahl m) und verschlüsselt diese mit dem privaten Schlüssel. A versendet das unverschlüsselte Dokument (die Bestellung) zusammen mit der verschlüsselten Prüfsumme. U entschlüsselt letztere mit dem öffentlichen Schlüssel und vergleicht das Ergebnis mit der Prüfsumme des Dokuments. Übereinstimmung bedeutet, dass A wirklich der Absender ist und dass das Dokument nicht verändert wurde – es sei denn, der private Schlüssel wurde geknackt und der Prüfsummenalgorithmus ist ebenfalls bekannt.
Noch raffinierter ist es, wenn die Nachricht selbst auch geheim bleiben und zusätzlich die Urheberschaft mittels Signatur festgestellt werden soll. Hier kann man zwei Paare von Schlüsseln verwenden, näheres siehe **RSA-Algorithmus als Signaturverfahren**[27] (Uni Bielefeld).

Bei den EC-Karten spielen andere Methoden als RSA eine Rolle (DES (Data Encryption Standard)). Hier wird mit Hilfe eines streng geheimen Schlüssels (*Institutsschlüssel*) aus der Kartenfolgenummer, der Kontonummer und der BLZ der Bank eine 4-stellige Geheimzahl ermittelt. Der Geldautomat in Deutschland ist mit einem Rechner verbunden, der diesen Schlüssel implementiert hat. Anders ist es mit Geldautomaten im Ausland. Hier gibt es sogenannte Poolschlüssel.
Internet:
Geheimzahlen bei EC-Karten[28]
Die digitale Signatur[29] (M. Schwarz, HU Berlin)

[27] http://www.wiwi.uni-bielefeld.de/StatCompSci/lehre/material_spezifisch/statalg00/rsa/
[28] http://www.informatik.uni-trier.de/ damm/Lehre/E-Money/ecCardsSecurityFAQ.html
[29] http://www.hu-berlin.de/rz/projekte/uvsec/berichte/vortrag/digsig/sld001.htm