Kryptologie, SS03 Aufgabenblatt 2

Teilbarkeit und Euklidischer Algorithmus

Für alle im folgenden auftauchenden Ringe $(R,+,\cdot)$ sei die Multiplikation kommutativ und besitze ein Einselement 1 („Kommutativer Ring mit 1“) und *R* sei auch ein *Integritätsring*, d.h. für alle $a,b\in R$ gelte: $ab=0\rightarrow(a=0$ oder $b=0)$. (Man sagt auch, der Ring ist „nullteilerfrei“.)

Ein *Körper* $(K,+,\cdot)$ ist ein Identitätsring, für den gilt: $\forall x\in K, x\neq 0\ \exists x'\in K:\ x\cdot x'=x'\cdot x=1$. Beispielsweise ist der Ring der ganzen Zahlen $\mathbb{Z}$ ein Integritätsring, aber kein Körper, $\mathbb{Z}_n$ ist ein Körper, wenn *n* prim ist; sonst ist $\mathbb{Z}_n$ nicht einmal ein Integritätsring. Ist *R* ein Integritätsring, so auch der zugehörige Polynomring $R[X]$.

Ist *R* ein Integritätsring, so schreibt man für Elemente $a,b\in R$: a/b wenn $\exists c\in R: bc=a$ („*a* teilt *b*, oder auch *a* ist Teiler von *b*“). Besitzt ein Element $a\in R$ in *R* ein multiplikativ Inverses, so nennt man a eine *Einheit*. Die Menge R^* der Einheiten ist eine Gruppe bzgl. der Multiplikation mit neutralem Element 1. Es ist z.B. $\mathbb{Z}^*=\{-1,1\}$, für den Ring der *Gaußschen Zahlen* $\mathrm{G}:=\{a+ib\in\mathbb{C}\,|\,a,b\in\mathbb{Z}\}$ – das ist der Teilring der komplexen Zahlen mit ganzzahligem Real- und Imaginärteil – ist $G^*=\{1,i,-1,-i\}$.

Gilt für $a,b\neq 0$: a/b und b/a, so gibt es offenbar $c,d\in R$ mit $b=ca$ und $a=db$, also $b=cdb$, also $0=b-cdb=b(1-cd)$, woraus $cd=1$ folgt. *c,d* sind dann also Einheiten.

Sind $a,b\in R$ Elemente eines Integritätsringes so heißt $c\in R$ größter gemeinsamer Teiler von *a* und *b*, wenn c/a, c/b und $(\forall d\in R:\ (d/a$ und $d/b)\rightarrow d/c)$. Es ist nicht unbedingt klar, ob in jedem Integritätsring zu beliebigen Elementen *a* und *b* ein größter gemeinsamer Teiler existiert. Zwei größte gemeinsame Teiler c,c' von *a* und *b* können durchaus verschieden sein, aber wegen c/c' und c'/c unterscheiden sie sich nur um eine Einheit, also $c'=ec$ mit $e\in R^*$. In den im folgenden definierten Euklidischen Ringen können wir aber mit dem sog. Euklidischen Algorithmus immer einen größten gemeinsamen Teiler konstruieren.

Ein Integritätsring heißt *Euklidischer Ring*, wenn es eine Abbildung $d: R-\{0\}\rightarrow\mathbb{N}_0:=\mathbb{N}\cup\{0\}$ gibt mit folgender Eigenschaft:

$$\forall a,b\in R, b\neq 0\ \exists q,r\in R, (r=0 \text{ oder } d(r)<d(b)):\ a=qb+r \quad \text{(Division mit Rest)}$$

Ein wichtiges Beispiel für einen Euklidischen Ring sind die ganzen Zahlen $\mathbb{Z}$ mit $d(n):=|n|$ oder ein Polynomringring $K[X]$ über einem Körper mit $d(f):=\mathrm{grad}(f)$. Auch die Gaußschen Zahlen sind ein Euklidischer Ring mit $d(a+ib):=|a+ib|=\sqrt{a^2+b^2}$.

Seien nun $a,b\in R, b\neq 0$ gegeben. Man setze $r_0:=a$, $r_1:=b$, $s_0:=1, s_1:=0$, $t_0:=0$, $t_1:=1$ und führe nun rekursiv eine Division mit Rest von r_{i-1} durch r_i durch: $r_{i-1}=q_i r_i+r_{i+1}$ und berechne anschließend, falls nicht $r_{i+1}=0$, das nächste Glied der beiden anderen Folgen: $s_{i+1}=-q_i s_i+s_{i-1}$, $t_{i+1}=-q_i t_i+t_{i-1}$.

Solange für $i \geq 1$ $r_{i+1} \neq 0$ ist, gilt $d(r_{i+1}) < d(r_i)$.
Zwangsläufig wird in diesem Prozeß daher irgendwann $r_{i+1} = 0$. Weil offenbar jeder Teiler von a, b auch Teiler jedes r_i ist, muß das letzte $r_i \neq 0$ ein größter gemeinsamer Teiler von *a* und *b* sein.

In der Vorlesung wurde gezeigt: $s_i a + t_i b = r_i$.

Aufgabe 1.

a) Man schreibe obigen Algorithmus in einer Programmiersprache, also
Input: Natürliche Zahlen $a,b \neq 0$.
Output: *c,x,y*, so daß *c*>0 größter gemeinsamer Teiler von *a,b* ist und $c = xa + yb$

b) Für $n \in \mathbb{N}$, $0 < a < n$ sei 1 größter gemeinsamer Teiler von *a,n*. Man formuliere einen Algorithmus (Cut und Paste aus Aufgabe 1a) mit
Input: *n,a*
Output: *b*, wobei $0 < b < n$ und $ba=1$ in $\mathbb{Z}_n$.

c) $n = 2^{31} - 1$ ist eine sog. Mersennesche Primzahl. Dies ist eine Größenordnung, in der die übliche 32 Bit-Arithmetik auf Computern noch funktioniert.
Man berechne mit dem Algorithmus aus Aufg. 1b) 143^{-1} in $\mathbb{Z}_n$, zeige aber auch die Zwischenergebnisse r_i, s_i.

Aufgabe 2

$p = 37463$, $q = 21893$ sind Primzahlen. Man setze $n := pq$, $e := 127$ und interpretiere (n, e) als öffentlichen Schlüssel für das RSA-Kryptosystem.

a) Man berechne den geheimen Exponenten *d*.
b) Man schreibe ein Computerprogramm, welches auch für bis zu 32 Bit große Exponenten *a* praktisch in der Lage ist, für $x \in \mathbb{Z}_n$: die Potenz x^d in $\mathbb{Z}_n$ zu berechnen. Orientieren Sie sich an dem Beispiel $x^{22} = \left(\left(\left(x^2\right)^2 x\right)^2 x\right)^2$, welches nur wenige Quadrierungen und Multiplikationen benötigt. Achten Sie darauf, Zwischenergebnisse sofort wieder „modulo *n*“ zu „reduzieren“, so daß keine Zahlen auftauchen, die eine möglicherweise benutzte 32Bit-Arithmetik sprengen.
c) Zeigen Sie an einem Beispiel, daß auch für ein $x \in \mathbb{Z}_n$ mit $\mathrm{ggT}(x, n) > 1$ in $\mathbb{Z}_n$ gilt:
$\left(x^e\right)^d = x$

Aufgabe 3
Studieren Sie das Programmpaket PariGP, um eine interaktive Langzahlarithmetik zur Verfügung zu haben und Aufgaben wie die obigen leicht bearbeiten zu können:
http://www.parigp-home.de
Man kann aber auch die Pari-Bibliothen z.B. in C/C++/Java Programme einbinden.