Mathematik und Logik

6. Übungsaufgaben

2006-01-24, Lösung

1. Berechnen Sie für das Konto 204938716 bei der Bank mit der Bankleitzahl 54000 den IBAN. Das Verfahren ist z.B. auf `http://de.wikipedia.org/wiki/International_Bank_Account_Number` beschrieben. Wie hilft dabei modulare Arithmetik? Wiederholen Sie dies für Ihr eigenes Konto, und verwenden Sie `www.iban-rechner.de` oder ähnliche Tools nur zum Überprüfen.

   Lösung: Das Ergebnis muß die Form ATppbbbbbkkkkkkkkkkkk haben, wobei pp die noch zu bestimmende Prüfziffer ist, bbbbb die 5-stellige Bankleitzahl, und kkkkkkkkkkkk die auf 11 Stellen vorne mit Nullen aufgefüllte Kontonummer. Konkret: ATpp5400000204938716. Der ATpp Teil ist durch 102900 zu ersetzen und hinten anzufügen; Ergebnis 5400000204938716102900. Von dieser Zahl brauchen wir den Rest modulo 97. Dabei können wir diese lange Zahl in zweier Gruppen zerlegen, $54 \cdot 100^1 0 + 2 \cdot 10^7 + 4 \cdot 10^6 + \cdots + 10 \cdot 10^2 + 29 \cdot 10^1$, und modulo 97 evaluieren. Man beachte dabei, daß z.B. $10^2 \equiv_{97} 3$, daher $10^4 \equiv_{97} 9$, $10^8 \equiv_{97} 81 \equiv_{97} -16$. So läßt sich der Rest berechnen, ohne den Bereich der zweistelligen Zahlen wesentlich verlassen zu müssen. Als Rest ergibt sich schließlich 68, und die Differenz zu 98 ist 30, was die gesuchte Prüfziffer ergibt. Der IBAN ist somit AT30 5400 0002 0493 8716

2. Berechnen Sie $2^{340} \bmod 341$. Glauben Sie, daß 341 eine Primzahl ist?

   Lösung: Mittels der Methode des sukzessiven Quadrierend erhält man in ca 8 Schritten (Quadrieren und Multiplizieren von Zahlen bis maximal 340) das Ergebnis $2^{340} \equiv 1 \pmod{341}$. Da dieser Sachverhalt nur bei zusammengesetzen Zahlen verletzt ist, ergibt sich damit ein Hinweis, daß 341 tatsächlich eine Primzahl sein dürfte, aber kein Beweis dafür.

3. Wiederholen Sie das vorige Beispiel mit der Basis 3. Was stellen Sie fest?

Lösung: Wir gehen ähnlich wie im vorigen Beispiel vor und stellen fest $3^{340} \equiv 56 \not\equiv 1 \pmod{340}$. Da dies für keine Primzahl gelten kann, können wir uns sicher sein, daß 341 tatsächlich keine Primzahl ist, der Hinweis im vorigen Beispiel also irreführend war.

4. Wählen Sie $p = 11$ und $q = 23$ als „große“ Primzahlen. Sei $m = pq$; bestimmen Sie φ, und wählen Sie ein e relativ prim zu φ.

   Lösung: Wir rechnen: $m = p \cdot q = 11 \cdot 23 = 253$; $\phi = (p-1) \cdot (q-1) = 10 \cdot 22 = 220$. Nun wählen wir irgendein e mit $3 < e < \phi$, welches zu *phi* relativ prim ist (d.h. $\mathrm{ggT}(e, \phi) = 1$). Wir probieren etwa $e = 25$, und überprüfen mit dem Euklidschen Algorithmus; dies liefert $\mathrm{ggT}(25, 220) = 5$; damit ist dieser Versuch fehlgeschlagen, und wir probieren eine andere Zahl, etwa $e = 17$; hier liefert der Euklidsche Algorithmus $\mathrm{ggT}(17, 220) = 1$; somit ist $e = 17$ eine geeignete Wahl.

   Anmerkung: Diese Wahl ist ganz beliebig. Tatsächlich gibt es weitere 75 Möglichkeiten. Oder anders herum: Es gibt 216 Zahlen zwischen 3 und 220; davon sind 78 geeigent (d.h. relativ prim zu 221); wählt man eine Zahl zufällig (gleichverteilt), dann erwischt man mit einer Wahrscheinlichkeit von $\frac{78}{216} \approx 0.36$ eine geeignete; nach 10 Schritten hat man dann mit einer Wahrscheinlichkeit von 99% eine geeignete gefunden. Dieses Probierverfahren funktioniert also tatsächlich, nämlich in dem Sinne, daß man mit großer Wahrscheinlichkeit schon nach ein paar Schritten am Ziel ist, und die Wahrscheinlichkeit, daß man wirklich lange probieren muß, extrem niedig ist.

5. Bestimmen Sie zu e aus dem vorigen Beispiel ein passendes d mit mit $de \equiv 1 \pmod{\varphi}$.

   Lösung: Der erweitere Euklidische Algorithmus liefert hier die Lösung $d = 13$. Dies ist leicht nachprüfbar: $13 \cdot 17 = 221 \equiv 1 \pmod{220}$.

   Damit haben wir kann man RSA-verschlüsseln: $(253, 17)$ ist der öffentliche Schlüssel, $(253, 13)$ der geheime.

   Anmerkung: Natürlich ist dieses konkrete Verschlüsselungsverfahren leicht zu knacken, weil 253 leicht zu faktorisieren (d.h. in Faktoren zerlegbar) ist. In der Realität verwendet man für p und q ca 150-stellie Zahlen, womit m ca 300 Dezimalstellen hat. Zahlen dieser Größenordnung können mit heutiger Technik und dem aktuellen Stand der Wissenschaft nicht faktorisiert werden. (Unbekannt, obwohl eher unwahrscheinlich, ist aber, ob nicht doch jemand (etwa ein Geheimdienst) ein Wissen, z.B. wie man schneller faktorisieren kann, für sich behält, und damit das RSA-Verfahren knacken kann).

6. Wählen Sie ein $x \colon \mathbb{Z}_m$, bestimmen Sie $y = x^e$ und rechnen Sie nach, ob $y^d = x$ ist. Wiederholen Sie das Beispiel für mehrere x.

Lösung: Sei $x = 118$; dann ist $y = x^e = 118^{17} = 200 : \mathbb{Z}_{253}$; und weiters $y^d = 200^{13} \equiv 1 : \mathbb{Z}_{253}$, so wie es schließlich sein soll.

7. Alice verwendet den/die Schlüssel aus den obigen Beispielen. Der öffentliche Schlüssel von Bob sei $m = 323$, $e = 5$. Alice will Bob die geheime Nachricht 16 mitteilen. Was muß sie versenden?

   Lösung: Alice's Schlüssel ist für dieses Beispiel irrelevant. Sie verwendet lediglich Bob's öffentlichen Schlüssel, rechnet $y = 16^5 = 118 : \mathbb{Z}_{323}$, und versendet daher die Zahl 118.

8. Bob hat die Nachricht empfangen; sie überrascht ihn aber, und er hat Zweifel, ob sie tatsächlich von Alice stammt. Er bittet sie daher, ihre Nachricht zusätzlich zu signieren. Was muß Alice nun schicken?

   Lösung: Bob empfängt also die Nachricht 2. Mit seinem geheimen Schlüssel d' kann er die ursprüngliche Nachricht rekonstruieren: $118^{d'} = 16 : \mathbb{Z}_{323}$; und das überrascht ihn. Er kennt den öffentlichen Schlüssel von Alice; diese signiert daher die Nachricht (durch Anwendung ihres geheimen Schlüssels), berechnet als $x' = x^d = 16^{13} = 26 : \mathbb{Z}_{253}$; dies ist die signierte Nachricht, welche als Element von Z_{323} aufgefaßt wird, und dann wie vorhin verschlüsselt versandt wird: $y' = x'^{e'} = 26^5 = 144 : \mathbb{Z}_{323}$. Tatsächlich auf die Reise geht hier somit die Zahl 144.

   Bob verwendet kann dann mit seinem privaten Schlüssel $144^{d'} : \mathbb{Z}_{m'}$ berechnen, um x' zu erhalten, und mit Alice's öffentlichem Schlüssel erhält er schließlich die Nachricht x selbst. Da allgemein angenommen wird, daß für die Berechnung von x' aus x der geheime Schlüssel d tatsächlich notwendig ist, nimmt Bob an, daß die Nachricht tatsächlich von Alice stammt. (Genauer, von jemanden, der Zugriff auf Alice's geheimen Schlüssel hat, oder das Verfahren geknackt hat.)

9. Alice erfährt, daß Bob ein Verhältnis mit Carol hat, was ihr überhaupt nicht paßt. Sie will daher eine gefälschte von Bob signierte Nachricht (die Zahl 66 bedeute: „ich will nichts mehr mit dir zu tun haben“) an Carol schicken. Was muß sie tun?

   Lösung: Offensichtlich braucht sie den geheimen Schlüssel von Bob. Sie kann ihm diesen in einer schwachen Stunde entlocken oder sonstwie stehlen, oder das Verfahren knacken, etwa indem sie m' faktorisiert. So schwierig letzteres für wirkclich große Zahlen ist, so leicht erhalten wir aber hier speziell $m' = 323 = 17 \cdot 19$. Damit ist der Schlüssel geknackt. Leicht zu berechnen sind nun $\varphi' = (17 - 1) \cdot (19 - 1) = 288$, und $d' = 173$. Sie versendet daher $66^{173} = 138 : \mathbb{Z}_{323}$ an Carol, welche dann mit Bob's öffentlichen Schlüssel die Authentizität der Nachricht sicherstellt (und schwer enttäuscht ist).

10. Schreiben Sie ein kurzes rekursives Programm, welches $x^e \bmod m$ für beliebige ganze Zahlen x, e, m effizient berechnet.

    Lösung: Das Potenzieren erfüllt die folgenden rekursiven Gleichungen:

$$\begin{aligned} x^0 &= 1, \\ x^{2e} &= (x^e)^2, \\ x^{2e+1} &= x \cdot (x^e)^2. \end{aligned}$$

    Man kann diese Beziehungen noch etwas einfacher und allgemeiner anschreiben: Sei $e = q \cdot b + r$; dann gilt:

$$x^e = x^{q \cdot b + r} = (x^q)^b \cdot x^r.$$

    Für den Spezialfall $b = 2$ erhält man sukzessives Quadrieren.

    Für negative Exponenten verwendet man

$$x^{-e} = (x^{-1})^e,$$

    um sie auf postive Exponenten zurückzuführen. Die Berechnung von x^{-1} erfolgt mit dem erweiterten Euklidschen Algorithmus.

11. Dasselbe für den (erweiterten) Euklidischen Algorithmus. (Input: (m, n), Output: (d, x, y), sodaß $d = m \cdot x + n \cdot y$)

    Lösung: Man verwendet die Beziehungen:

$$\begin{aligned} &\mathrm{ggT}(m, 0) = m, \\ &\mathrm{ggT}(m, n) = \mathrm{ggT}(n, r) \qquad \text{falls } m = r \pmod{n}. \end{aligned}$$

    Verwendet man statt der ersten Gleichung die Gleichung $\mathrm{ggT}(m, 0) = |m|$ (Absolutbetrag), dann wird garantiert, daß das Ergebnis stets positiv ist, auch bei negativen Eingaben.

    Für den erweiterten Euklidschen Algorithmus verwendet man

$$\begin{aligned} &\mathrm{xggT}(m, 0) = (|m|, \operatorname{sgn} m, 0), \\ &\mathrm{xggT}(m, n) = (d, y, x - y \cdot q) \qquad \text{falls } m = r \pmod{n} \\ &\qquad\qquad\qquad\qquad\quad \text{und } (d, x, y) = \mathrm{xggT}(n, d). \end{aligned}$$

12. Verwenden Sie diese Programme, um die vorigen Beispiele mit zumindest 4-stelligen Zahlen zu berechen. Noch besser arbeiten Sie mit 100-stelligen Zahlen, falls ihr Programm das unterstützt.

    Lösung: Siehe Realisierung in Python.

13. Wieviele Leute kennen sich mit Hexadezimalzahlen aus, wenn nur Sie und fade Leute sich damit auskennen?

    Lösung: fade + 1 = fadf Leute. Oder, in Dezimalschreibweise, 54223.