Prof. J. Rothe Düsseldorf, 20.10.2009

Übung zur Vorlesung
Kryptologie 2
Blatt 2, Abgabe: 29.10.2009 bis 15:00 Uhr

Aufgabe 1 (5 Punkte): Betrachten Sie das RSA-Kryptosystem über einem beliebigen Alphabet Σ mit RSA-Modul n. In der Vorlesung wurde gezeigt, dass RSA Blöcke der Länge $\lfloor \log_{|\Sigma|} n \rfloor$ auf Blöcke der Länge $\lceil \log_{|\Sigma|} n \rceil$ abbildet. Zeigen Sie, dass stets

$$\lceil \log_{|\Sigma|} n \rceil = \lfloor \log_{|\Sigma|} n \rfloor + 1$$

gilt. Mit anderen Worten: Zeigen Sie, dass $\log_{|\Sigma|} n$ keine ganze Zahl ist.

Aufgabe 2 (5 Punkte): In der Vorlesung wurde gezeigt, wie eine Zahl a in $\mathbb{Z}_n$ mit dem erweiterten Algorithmus von Euklid invertiert werden kann, wenn $ggT(a, n) = 1$ gilt.
Zeigen Sie, dass das Inverse von a dann eindeutig in $\mathbb{Z}_n$ ist.

Aufgabe 3 (10 Punkte): Betrachten Sie das RSA-Kryptosystem.
Erich hat $(n, e) = (17583, 3907)$ und $c =$ `AMDU` empfangen.
Welche Nachricht m hat Alice verschickt?

Aufgabe 4 (10 Punkte): Zeigen Sie, dass beim RSA-Kryptosystem stets

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

gilt. Dabei sind wie üblich n der RSA-Modul, e der öffentliche Exponent, d der private Exponent und m der Klartext.
Hinweis: Nach dem kleinen Satz von Fermat gilt die Kongruenz $a^{p-1} \equiv 1 \mod p$ für jede Primzahl p und jede natürliche Zahl a, die kein Vielfaches von p ist.

Aufgabe 5 (freiwillig): Implementieren Sie eine Methode

```
BigInteger ggT(BigInteger a, BigInteger b)
```

in der Programmiersprache Java, die iterativ den größten gemeinsamen Teiler zweier ganzer Zahlen a und b berechnet. Schreiben Sie auch eine Methode, die den ggT rekursiv berechnet und vergleichen Sie die Laufzeit (für große Zahlen).
Wer diese Aufgabe ernst nimmt, der verwendet nicht die Methode `gcd` *der BigInteger-Klasse.*