**RSA-Verfahren**

**1. Verschlüsseln**

Man benötigt zwei (große) Primzahlen p und q, die man multipliziert, um die Zahl N zu erhalten. Außerdem benötigt man eine Verschlüsselungszahl e (*encipher* = *verschlüsseln*). Dafür kann man fast jede Zahl nehmen. Sie muss nur kleiner als n und teilerfremd zu (p – 1)(q – 1) sein.

Beide Zahlen sind öffentlich.

Wenn man die Zahl M (*Message*) verschlüsseln will, berechnet man die Zahl C (*Cipher*) durch

C = Me  (mod N)

**Beispiel:**Nimm p = 11 und q = 17, dann ist N = pq = \_\_\_\_\_\_\_\_.

Für e nimm die Zahl 7.

*Verschlüssele nun die Zahl M = 59.*Dann ist C = \_\_\_\_\_\_.

*Verschlüssele weitere Zahlen mit diesem Schlüssel. Diese müssen kleiner als 187 sein.*

**2. Entschlüsseln**

Hier benötigt man die (geheime) Entschlüsselungszahl d (*decipher* = *entschlüsseln*). (Was für Eigenschaften diese haben muss, steht unter 3.)

Wenn man die Zahl C entschlüsseln will, berechnet man

M = Cd (mod N)

**Beispiel:**Hier ist d = 23.

*Entschlüssele damit die Zahlen, die dein Nachbar verschlüsselt hat.*

*Was ist, wenn man Zahlen verschlüsselt, die größer als 187 sind?*

**3. Eigenschaft der Zahl d**

Die Entschlüsselungszahl d muss folgende Eigenschaft haben:

ed ≡ 1 ( mod (p – 1)  (q – 1) ) ,

d.h. modulo der Zahl (p – 1)  (q – 1) muss e  d ≡ 1 sein.

**Beispiel:**

*Überprüfe, ob dies für die Zahlen aus unserem Beispiel erfüllt ist.*

Ein anderes Beispiel: p = 3, q = 11, also N = \_\_\_\_\_\_.

Es soll e = 7 gewählt werden.

*Suche eine Zahl d, die die geforderte Eigenschaft erfüllt.*

Wenn p und q groß sind, kann man d nicht mehr so einfach durch Probieren finden.

Es gibt aber ein Verfahren, mit dem man d schnell bestimmen kann, wenn p und q sowie e bekannt sind.

**4. Warum funktioniert das entschlüsseln?**

Die verschlüsselte Nachricht ist C = Me (mod N)

Das Entschlüsseln funktioniert also, wenn Cd ≡ M (mod N) ist.

Nach den Potenzgesetzen ist Cd = (Me)d = Med.

Behauptung: Med≡M (mod N)

Beweis: Die Zahlen e und d sind so gewählt, dass

e  d ≡ \_\_\_\_\_\_ (mod (p – 1)  (q – 1) )

Also gibt es eine Zahl k, so dass e  d = k  (\_\_\_\_\_\_\_\_\_\_\_\_\_\_) + \_\_\_\_.

Folglich ist Med = M\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ + 1 = M\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  M = (M(p – 1))k\_\_\_\_\_\_\_  M.

Nach dem kleinen Satz von Fermat ist Mp – 1 ≡ \_\_\_\_ (mod p).

Damit: Med – M = (M(p – 1))k\_\_\_\_\_\_\_  M – M ≡ \_\_\_k  (q – 1) M – M ≡ M – M ≡ \_\_ (mod p).

Genauso ist nach dem kleinen Satz von Fermat: Mq – 1 ≡ \_\_\_\_\_ (mod q).

Damit Med – M = (M(q – 1))k\_\_\_\_\_\_\_  M – M ≡ \_\_\_k  (p – 1) M – M ≡ M – M ≡ \_\_ (mod q).

Die Primzahlen p und q teilen also beide die Zahl Med – M. Da p und q Primzahlen sind, folgt daraus, dass auch p  q die Zahl Med – M teilt.

Also ist Med – M ≡ 0 (mod N) oder anders ausgedrückt Med≡ \_\_\_\_\_ (mod N).

*q.e.d.*

**5. Warum ist das RSA-Verfahren sicher?**

Öffentlich sind nur die Zahlen N und e, die Zahl d ist geheim. N ist das Produkt zweier Primzahlen N = p  q. Wenn man p und q kennt, kann man d schnell berechnen.

Aber für große Zahlen ist es praktisch unmöglich, N in seine beiden Primfaktoren zu zerlegen. Und ohne p und q zu kennen, kann man d nicht bestimmen.