# Kryptografie RSA-Verfahren mit TI Nspire

**RSA–Verfahren, Version mit kleinen Primzahlen** Haftendorn Nov.2010

Anton möchte, jeder ihm Nachrichten schicken kann, die nur er selbst lesen.

Niemand, der die Kommunikation abfängt. soll eine Chance haben, den Klartext herauszubekommen.

**Anton bereitet seine Schlüssel vor**:

Er wählt zwei Primzahlen **kry\nextprime**(randInt(1,6)) ▸ 7

**p**:=**kry\nextprime**(randInt(50,200)) ▸ 67 und
**q**:=**kry\nextprime**(randInt(50,200)) ▸ 149.

Das Produkt wird der erste Teil seines öffentlichen Schlüssels **n**:=**p**·**q** ▸ 9983.

Im der Gruppe Z*(n) wird später potenziert, daher braucht er die Ordnung von Z*(n).

**phi**:=(**p**−1)·(**q**−1) ▸ 9768. Nun wählt er ein "gutes" **e** aus Z*(phi). "Schlechte" **e** sind solche mit zu kleiner Ordnung. das passiert bei unseren "kleinen" Beispielen leicht.

**e**:=randInt(50,**phi**−2) ▸ 2339 und prüft sofort **li**:=**kry\ggte**(**e**,**phi**) ▸ [1 3011 -721]

Zu diesem **e**muss der num das Inverse in Z*(phi) bestimmen. Es ist das zweite Element dieser Liste **d**:=mod(**li**[1,2],**phi**) ▸ 3011 , modulo phi genommen.

Probe mod(**e**·**d**,**phi**) ▸ 1. Sein d hält er geheim, sein öffentlicher Schlüssel ist das Zahlenpaar [**e** **n**] ▸ [2339 9983]

---

**Anwendungsphase**

Berta will Anton eine Nachricht senden

**m**:=1988 ▸ 1988

**c**:=**kry\pmod**(**m**,**e**,**n**) ▸ 7691 Dies erhält Anton.

---

**Entschlüsselung, Anton kann die Nachricht lesen**

**mm**:=**kry\pmod**(**c**,**d**,**n**) ▸ 1988

Datei rsa-klein.tns