Erweiterter Euklidischer Algorithmus um d zu bestimmen!

Primzahlen und passendes e wählen N und phi(N) berechnen!

p = 7

q = 11

N = p*q = 77

phi(N) = 60

e = 7

d =? → bestimmen

e * d mod phi(N) = 1

Rechenweg um s und t zu bestimmen

$s_{neu} = t_{alt}$

$t_{neu} = s_{alt} - (q_{neu} * t_{alt})$

a	b	q	r	s	t
7	60	q_{neu} 0	7	$s_{neu} = t_{alt}$ -17	$t_{neu} = s_{alt} - (q_{neu} * t_{alt})$ $t_{neu} = 2 - (0 * (-17))$ 2
60	7	q_{neu} 8	4	$s_{neu} = t_{alt}$ 2	$t_{neu} = s_{alt} - (q_{neu} * t_{alt})$ $t_{neu} = -1 - (8 * 2)$ -17
7	4	q_{neu} 1	3	$s_{neu} = t_{alt}$ -1	$t_{neu} = s_{alt} - (q_{neu} * t_{alt})$ $t_{neu} = 1 - (1 * (-1))$ 2
4	3	q_{neu} 1	1	$s_{neu} = t_{alt}$ 1	$t_{neu} = s_{alt} - (q_{neu} * t_{alt})$ $t_{neu} = 0 - (1 * 1)$ -1
3	1	3	0	$s_{neu} = t_{alt}$ 0	1

In der Spalte s und in der obersten Zeile ist nun die Zahl -17.
Da die Zahl negativ ist, wird diese um phi(N) erhöht.

- 17 + phi(N) = d
- 17 + 60 = 43 → d = 43!
Probe: e * d mod phi(N) = 1
7 * 43 mod 60 = 1 → 301 mod 60 = 1 → passt!

Public Key (e, N), (7, 77) und Private Key (d,N), (43, 77)

Verschlüssele m = 16

$c = m^e \bmod N$
$c = 16^7 \bmod 77$
$c = 58$

Entschlüssele c

$m = c^d \bmod N$
$m = 58^{43} \bmod 77$
$m = 16$

Die Sicherheit des RSA-Verfahrens beruht darauf, dass wir mit unseren aktuellen technischen Mitteln aus einer großen Zahl **N = p · q** die beiden Primfaktoren p und q nicht effizient berechnen können.

Sonst könnte jede Person aus dem öffentlichen Schlüssel auch den privaten Schlüssel berechnen und die Nachrichten entschlüsseln.

42-stellige Zahlen zerlegt GeoGebra im Bruchteil einer Sekunde in ihre Primfaktoren:
626 174 180 288 988 003 026 978 279 277 333 965 732 409 = 3·7·181·1049·2269·11 519·18 191·6 663 303 217·49 570 850 878 973

In der Praxis werden deshalb beim RSA-Verfahren Primzahlen mit rund 400 Stellen gewählt!