

RSA Calculator

JL Popyack, October 1997

This guide is intended to help with understanding the workings of the RSA Public Key Encryption/Decryption scheme. No provisions are made for high precision arithmetic, nor have the algorithms been encoded for efficiency when dealing with large numbers.

Step 1. Compute N as the product of two prime numbers p and q :

p

q

Enter values for p and q then click this button:

The values of p and q you provided yield a modulus N , and also a number $r=(p-1)(q-1)$, which is very important. You will need to find two numbers e and d whose product is a number equal to $1 \bmod r$. Below appears a list of some numbers which equal $1 \bmod r$. You will use this list in Step 2.

$N = p \cdot q$

$r = (p-1) \cdot (q-1)$

```
353 705 1057 1409 1761 2113 2465 2817 3169 3521 3873
4225 4577 4929 5281 5633 5985 6337 6689 7041 7393 7745
8097 8449 8801 9153 9505 9857 10209 10561
```

Candidates ($1 \bmod r$):

Step 2. Find a number equal to $1 \bmod r$ which can be factored:

K

Enter a candidate value K in the box, then click this button to factor it:

factors of K :

Step 3. Find two numbers e and d that are relatively prime to N and for which $e \cdot d = 1 \bmod r$:

Use the factorization info above to factor K into two numbers, e and d . Click button to check correctness:

e

d

```
e   = 3
d   = 235
N   = 391
r   = 352
e*d = 705
e*d mod r = 1
e and r are relatively prime
d and r are relatively prime
```

Consistency check:

If your choices of e and d are acceptable, you should see the messages, " $e \cdot d \bmod r = 1$ ", " e and r are relatively prime", and " d and r are relatively prime" at the end of this box.

Step 4. Use e and d to encode and decode messages:

Enter a message (in numeric form) here. Click button to encode. Break your message into small chunks so that the "Msg" codes are not larger than **N**. (See [ASCII Code Chart](#) for ASCII code equivalences.)

Encode/Decode

Msg

Encrypted Cipher = (Msg)^e mod N

Decrypted Msg = (Cipher)^d mod N