**RSA public/private key generation**

Choose 2 random prime numbers (should be large) and compute produce: n = p\*q => 3\*11 = 33

Choose random integer e with e < (p-1)\*(q-1): 20

Compute the unique inverse: d = e-1 mod(p-1)(q-1))

d \* e mod ((p-1)(q-1))) = 1 *mathematical operations in finite fields: y = x-1mod(n) => x\*y mod(n) = 1*

(d \* e) mod (20) = 1

* Multiple possibilities for d and e
  + 3 \* 7 (and always the reverse)
  + 9 \* 9
  + Etc.

Public key: modulus n, e: 33, 3

Private key d: 7

**RSA Encryption/Decryption example**

Plaintext x: 16

Encryption: y = xe mod(n) => 163 mod(33) = 4096 mod(33) = 4

Decryption: x = yd mod(n) => 47 (mod33) => 16384 mod(33) => 16

**RSA: trying to find private key and plaintext when low numbers are used**

Alice sent Bob an encrypted number: 587. Mallory has intercepted the message and also holds a copy of Bob's RSA public key (n,e 2773,17). Proof that small hard problems are not hard and calculate the plaintext number.

y = xe mod(n) => x17 mod(2773) = 587

n -> is the product of 2 prime numbers: Found via trial and error: 47 + 59

d \* e mod ((p-1)(q-1))) -> d \* 17 mod(46\*58) = 1 => (d \* 17) mod(2668) = 1

Possibilities for d\*17 ->

* 1
* 2669 -> 157 -> d
* 5337 -> 313.94… (not a candidate)

Decryption: x = yd mod(n) => 587157 mod(2773) => 31 (calculation large numbers done with a little program: see <https://stackoverflow.com/questions/2177781/how-to-calculate-modulus-of-large-numbers> )

Encrypte again to check x = yd mod(n) => 160917mod(2773) => 587 (verified with <https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html> )