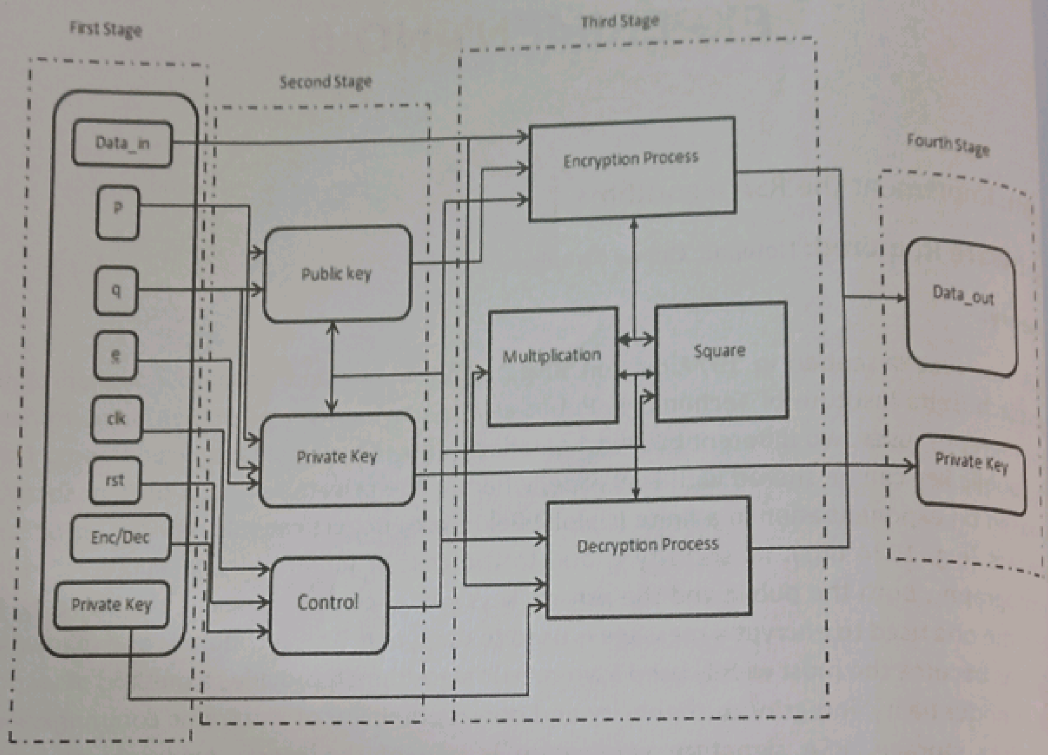

The RSA algorithm process.

The security of RSA is based on the belief that the encryption formula (c = me mod n) is a one-way function. The trapdoor that allows Bob to decrypt a Cipher text is the knowledge of factorization n = pq.

Example: Let p=3 and q=11 (both are prime numbers).

- Now, n = p*q = 3*11 = 33
- phi(n) = (p-1)*(q-1) = (3-1)*(11-1) = 2*10 = 20
- Value of e can be 7 since 1<7<20 and gcd(20, 7) = 1.
- Calculating d = 7^-1 mod 20 = 3.
- Therefore, public key = {7, 33} and private key = {3, 33}.

Suppose our message is M=31. You can encrypt and decrypt it using the RSA algorithm as follows:

Encryption: C = (M^e) mod n = 31^7 mod 33 = 4

Decryption: M = (C^d) mod n = 4^3 mod 33 = 31

Program:

```
import math
# step 1
p = 3
q = 7

# step 2
n = p*q
print("n =", n)

# step 3
phi = (p-1)*(q-1)

# step 4
e = 2
while(e<phi):
  if (math.gcd(e, phi) == 1):
    break
  else:
    e += 1

print("e =", e)
# step 5
k = 2
d = ((k*phi)+1)/e
print("d =", d)
print(f'Public key: {e, n}')
print(f'Private key: {d, n}')
```

plain text

msg = 11

print(f'Original message:{msg}')

encryption

C = pow(msg, e)

C = math.fmod(C, n)

print(f'Encrypted message: {C}')

decryption

M = pow(C, d)

M = math.fmod(M, n)

print(f'Decrypted message: {M}')

Output:

```
n = 21
e = 5
d = 5.0
Public key: (5, 21)
Private key: (5.0, 21)
Original message:11
Encrypted message: 2.0
Decrypted message: 11.0
```

Result:

We gained knowledge of symmetric encryption and the RSA algorithm in this experiment. We also saw how the RSA algorithm was implemented in Python.