- RSA Algorithm Function **(choose to prime number, get here factor)**
  - Popular **asymmetric key cryptosystem**
  - Developed in 1977 by Rivest, Shamir, and Adleman

If you understand modular arithmetic, you can appreciate the complexities of the RSA algorithm. It is based on the large amount of computational work required in factoring large composite numbers and computing the so-called **eth roots modulo**, a composite number for a specified odd integer **(e)**. Encryption in RSA is accomplished by raising the **message M** to a nonnegative integer power e. **The product is then divided by the nonnegative modulus n** (which should have a bit length of at least 1024), and the remainder is the **ciphertext C**. This process results in one-way operation (shown below) when n is a very large number.

- **M** represents **original message** / **m** represents the **(p – 1)(q - 1)**
- **C** represents **ciphertext**
- **e** represents the **encryption key**
- **d** represents the **decryption key** / **n** represents the **public modules**
- **p** and **q** represent **prime number** (share anyone) I will utilize my prime number

n = p * q / n: using for the modular function

m = (p – 1) (q – 1)

ed = 1 mod m

C = M^e mod n / d = e^-1 mod (p-1)(q-1)

M = C^d mod n => open the message

******* exercise*******

Alice decrypts the message and send to the Bob

Only Alice can decrypt the message

- Prime number

  P = 11 / q = 3

  e = 3 d = 7

n = 11 * 3 = **33**

m = (11 – 1) (3 -1) = (10)(2) = **20**

ed = 1 mod 20 / 21 = 1 mod 20 / 21 mod 20 = 1 reminder

send to n and e to Bob (public key)

n = 33 / e = 3 / M(P) = 14

C = 14^3 mod 33 = 5

Alice got message, she got ciphertext "C" (it is open to everyone)

She has "d" = 7 only Allice has it

M = C^d mod n = 5^7 mod 33 = 14

Alice wants to get the public and private key

P = 23 / q = 19

n = 23 * 19 = 437

m = (23 -1) (19-1) = (22)(18) = 396

e = 283 / d = ? = 7

ed = 1 mod 396

d = e ^ -1 mod 396 = (283) ^ -1 mod 396

e	*	**d**		**Until 1**
283	*	1	(283*1) mod 396	283
283	*	2	(283*2) mod 396	170
283	*	3	(283*3) mod 396	57
283	*	4	(283*4) mod 396	340
283	*	5	(283*5) mod 396	227
283	*	6	(283*6) mod 396	114
283	*	**7**	**(283*7) mod 396**	**1**

But this is not efficient way, GCD is much better

Greatest common divisors (GCD)

d = (283) ^ -1 mod 396

396 = 1 * 283 + 113 => 113 = (396 – 1 * 283)

283 = 2 * 113 + 57 => 57 = (283 – 2 * 113)

113 = 1 * 57 + 56 => 56 = (1 * 113 - 1 * 57)

57 = 1 * 56 + 1

1 = 1 * 57 – 1 * 56

= 1 * 57 – 1 * (1 * 113 - 1 * 57)

= 1 * 57 – 1 * 113 + 1 * 57

= 2 * 57 – 1 * 113

= 2 * (283 – 2 * 113) – 1 * 113

= 2 * 283 – 4 * 113 – 1 * 113

= 2 * 283 – 5 * 113

= 2 * 283 – 5 * (396 – 1 * 283)

= 2 * 283 – 5 * 396 + 5 * 283

= 7 * 283 – 5 * 396 mod 396 => 7 * 283 = (5 * 396 mod 396 = 0)

Alice

P = 17 / q = 13 / e = 77 / c = 19

m = ? / d = ? / n = 221

ed = 1 mod (p – 1)(q -1)

d = e ^ -1 mod (16)(12)

= 77 ^ -1 mod 192 => m = 192

GCD

192 = 2 * 77 + 38 => 38 = (1 * 192 – 2* 77)

77 = 2 * 38 + 1

1 = 1 * 77 – 2 * 38

= 1 * 77 – 2 * (1 * 192 – 2* 77)

= 1 * 77 – 2 * 192 + 4 * 77

= 5 * 77 – 2 * 192 mod 192 => (2 * 192 = 0)

⇨ d = 5

M = C ^ d mod n

M = 19 ^ 5 mod 221

M = 15

Verify the answer

C = m ^ e mod n

C = M ^ e mod 221

C = 15 ^ 77 mod 221

C = 19

Public key (n, e) = (323, 247)

P = 17 / c = 60 / M = ?

n = p * q

323 = 17 * q

q = 323 / 17 = 19

d = e ^ -1 mod (p – 1)(q - 1)

= 247 ^ -1 mod (16)(18)

= 247 ^ -1 mod 288

GCD

288 = 1 * 247 + 41 => 41 = (288 – 1 * 247)

247 = 6 * 47 + 1

1 = 247 – 6 * 41

= 247 – 6 * (288 – 1 * 247)

= 247 – 6 * 288 + 6 * 247

= 7 * 247 – 6 * 288 mod 288 => (6 * 288 = 0)

- **d = 7**

M = C ^ d mod n

M = 60 ^ 7 mod 323

M = 2

Verify the answer

C = m ^ e mod n

C = 2 ^ 247 mod 323

C = 60