RSA:

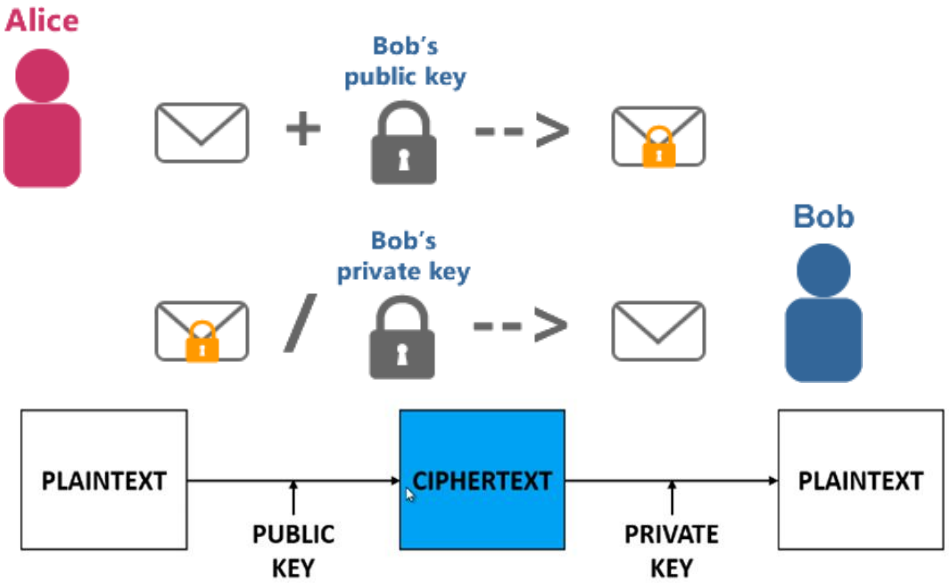
Is a Public Key cipher:

1) Alice will encrypt the data using the Bob's Private key and send it.
2) Bob will decrypt the data using its own Private Key

RSA is a popular public-key encryption algorithm that is widely used to secure digital communication. In RSA, two keys are used: a public key and a private key. Here's how they work:

3) Public key: The public key is used to encrypt data that is sent to the owner of the private key. The public key is generated by performing some mathematical operations on two large prime numbers. The resulting number is used as the modulus for the public key. The public key also has an associated exponent. The public key is widely distributed and can be freely shared.
4) Private key: The private key is used to decrypt data that has been encrypted using the corresponding public key. The private key is generated using the same two large prime numbers that were used to generate the public key. However, the private key is kept secret and is never shared with anyone.

The strength of RSA lies in the fact that it is very difficult to factor large numbers into their prime factors. This means that it is difficult for someone to compute the private key from the public key. As a result, RSA is considered a secure encryption algorithm that is widely used for secure communication over the internet.



Mathematical Concept

Important formulas

1) $N = p \times q$

2) $\Phi = (p-1)(q-1)$

3) $d \times e \bmod \Phi = 1$

4) Encryption: $C = P^e \bmod n$

Decryption: $P = C^d \bmod n$

Index:

P= Plain Text

C= Cipher Text

p and q are Prime Numbers

Φ = Euler's Phi Function #Ignore the name, just a fancy word

e= Public Key

d= Private Key

Let us take an example

Step 1)

p= 7 and q= 11

Φ = (p-1)(q-1) = 60

Step 2)

e can be any number such that

1) 1<e< Φ

2) e and Φ are co prime

Step 3)

d x e mod Φ = 1

ax + by = gcd(a,b) a= Φ and b = e

60x + 13y = gcd(60,13) = 1

We have One equation and 2 Variables

To find : y

Value of d = answer for y

This can be solved using Extended Euclid's Algorithm

Step I) Default Values

Sr NO	a	b	d	k
1	1	0	Φ = ?	----------
2	0	1	e= ?	

STEP II) Now Substitute Φ and e

Sr NO	a	b	d	k
1	1	0	Φ = 60	----------
2	0	1	e= 13	int (k1/k2)

Sr NO	a	b	d	k
1	1	0	60	----------
2	0	1	13	4

STEP III) Calculate using formula

1) a3= a1 – a2 x k2
2) b3= b1 – b2 x k2
3) d3= d1 – d2 x k2
4) k3= int (d2/d3)

Sr NO	a	b	d	k
1	1	0	60	----------
2	0	1	13	4
3	1	-4	8	1

1) a4= a2 – a3 x k3
2) b4= b2 – b3 x k3
3) d4= d2 – d3 x k3
4) k4= int (d3/d4)

Sr NO	a	b	d	k
1	1	0	60	----------
2	0	1	13	4
3	1	-4	8	1
4	-1	5	5	1

This process will continue till we get d =1
d= 1 #STOP

Sr NO	a	b	d	k
1	1	0	60	----------
2	0	1	13	4
3	1	-4	8	1
4	-1	5	5	1
5	2	-9	3	1
6	-3	14	2	1
7	5	-23	1	

STEP IV) Assign d

if (d > Φ):

d = b mod Φ

if (d is –Ve):

d = b + Φ

Now we have all values:

p = 7 (assumed)
q = 11 (assumed)
Φ = 60 (calculated)
n = 77 (calculated)
e = 13 (Assumed according to conditions)
d = 37 (calculated)

Let Plain Text (P) = 40

Encryption:

$C = P^e \bmod n$

$C = 40^{13} \bmod 77 = 68$

Decryption:

$P = C^d \bmod n$

$P = 68^{37} \bmod 77 =$ **40**