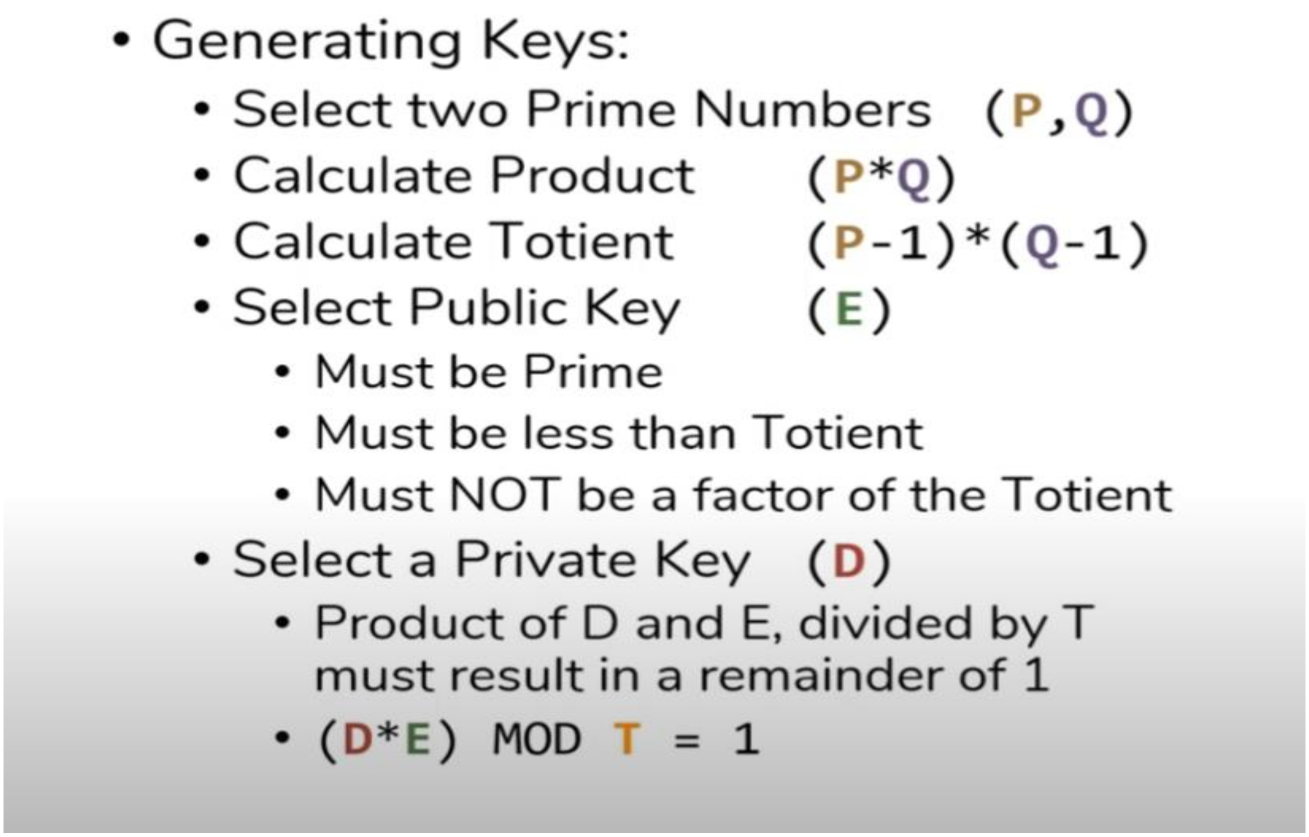
RSA

RSA was first introduced in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman, all being MIT colleagues. It is the most used public-key cryptography algorithm. RSA is an a-symmetric algorithm, this means that the key for encryption is not the same key for decryption. This adds an extra step to the security. RSA uses 2 large prime numbers (P, Q) to calculate n, this being the product of our prime numbers (P, Q). We then use these prime numbers (P, Q) to compute a public and private key (E, D), with all this information we can begin encrypting our target object. In order to get our public and private keys (E, D) we have some requirements for each key, these requirements for E include E must be prime, must be less than totient, and must not be a factor of the totient. The requirements for D are the product of E and D divided by T must result in a remainder of 1. First, we will take the public key (E) and raise our target object with the power of it. Next, we will take the remainder of it when divided by N, this will give us our cipher text. To use a private key to decrypt it, we will first take the cipher text and raise it to the power of the private key value (D). Then, we will find the remainder of that when divided by N.



- Generating Keys:
  - Select two Prime Numbers (P,Q)
  - Calculate Product (P*Q)
  - Calculate Totient (P-1)*(Q-1)
  - Select Public Key (E)
    - Must be Prime
    - Must be less than Totient
    - Must NOT be a factor of the Totient
  - Select a Private Key (D)
    - Product of D and E, divided by T must result in a remainder of 1
    - (D*E) MOD T = 1

REFERENCES

https://www.invent.org/inductees/leonard-adleman#:~:text=Introduced%20in%201977%20by%20MIT,Martin%20Hellman%2C%20and%20Ralph%20Merkle

https://www.practicalnetworking.net/series/cryptography/rsa-example/