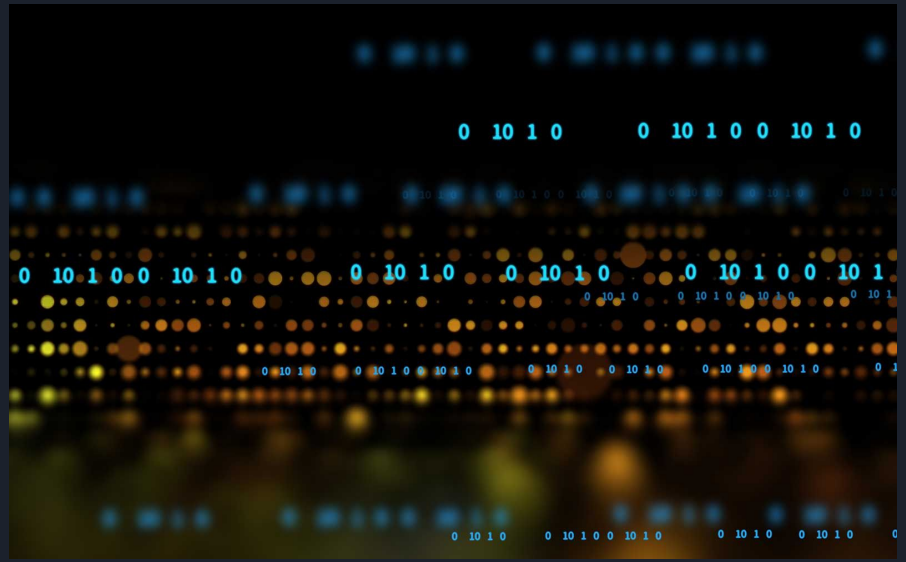
RSA Encryption

By Andrei Bato

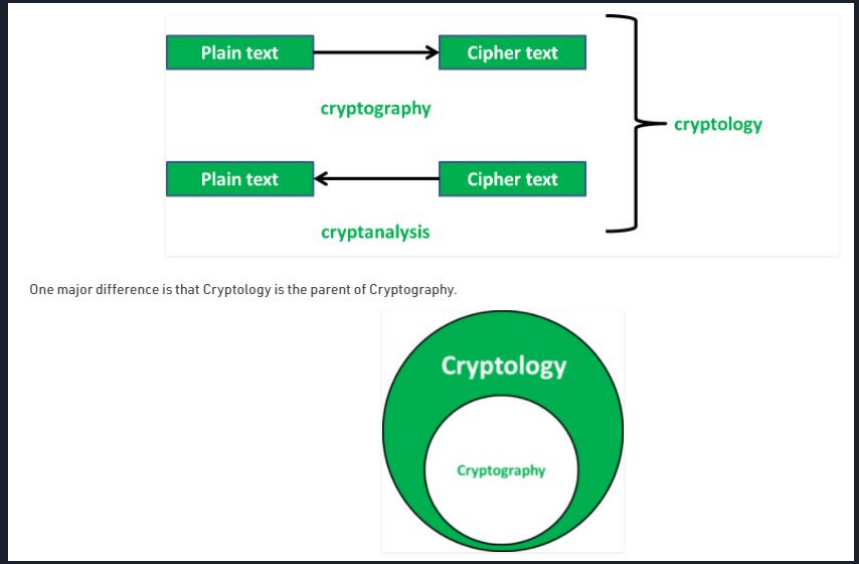
Presentation Overview

- Cryptology versus Cryptography
- What is RSA Encryption, Decryption, and RSA as a Public Key System.
- How does it work?
- What is the math involved ?
- My C++ Project



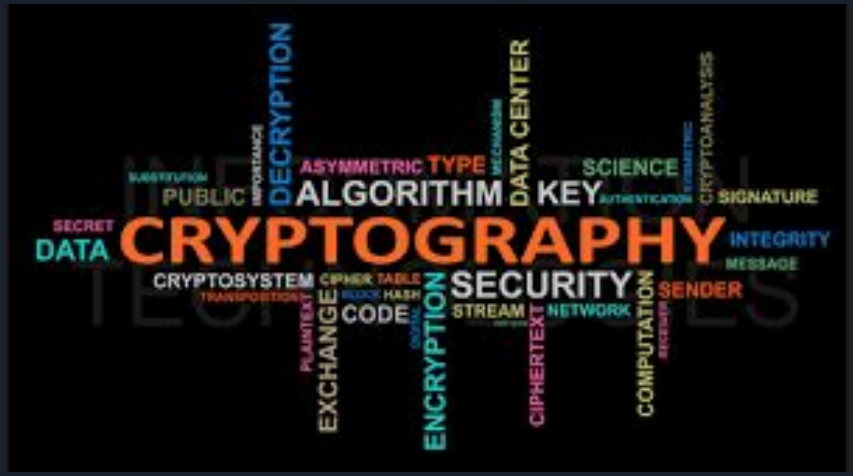
Cryptology Versus Cryptography

- **Cryptography** is the study of conversion of plain text(readable format) to ciphertext(non-readable format) i.e. encryption. It is also called the **study of encryption**
- **Cryptology** is the study of the conversion of plaintext to ciphertext and vice versa. It is also called the **study of encryption and decryption**



RSA Overview

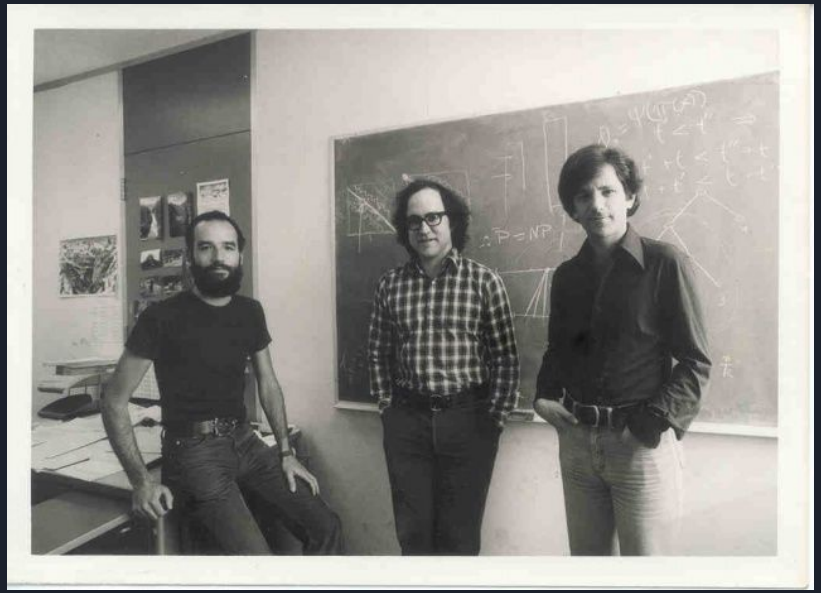
- Used by modern Computers to encrypt and decrypt messages.
- It is Asymmetric
- It is also a public key cryptology



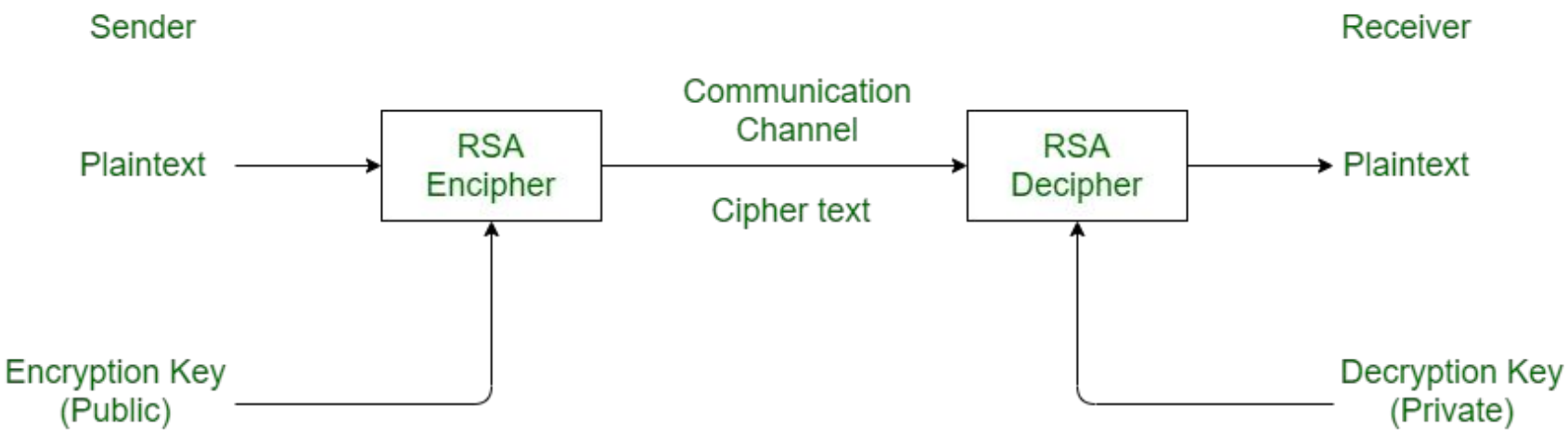
The History of RSA Encryption

- RSA or Rivest-Shamir-Adleman introduced in 1977 by MIT researchers Ron Rivest, Adi Shamir, and Leonard Adleman.

- It was innovated in 1976 byt Whitfield Diffie, Martin Hellman, and Ralph Merkle



RSA Visual Overview



RSA Algorithm

1. Select two prime numbers, prime1 and prime2
2. Find modulus of both keys with **n =** prime1 * prime2
3. Calculate **totient by (prime1 -1)(prime2-1)**
    i. Totient Function or Euler's totient is the number of positive integers that are relatively prime, 1 is counted as relatively prime
4. Choose **encrypt** such that **encrypt > 1** and coprime to **totient** which means **gcd (encrypt, totient)** must be equal to **1**, **encrypt** is the public key
5. Choose **d** such that it satisfies the equation **de = 1 + k (totient)**, **d** is the private key not known to everyone.
6. Cipher text is calculated using the equation **c = m^encrypt mod n** where **m** is the message.
7. With the help of **c** and **d** we decrypt message using equation **m = c^d mod n** where **d** is the private key.

Code Implementation in C++

Finding the Greatest Common Divisor

```
//to find gcd or greatest common divisor
int gcd(int first, int second)
{
    int temp;
    while(1)
    {
        temp = first%second;
        if(temp==0)
        return second;
        first = second;
        second = temp;
    }
}
```

Pick two random prime numbers, public and private key

```
int main()
{
    //2 random prime numbers
    double prime1 = 3;
    double prime2 = 7;
    double n=prime1*prime2;
    double count;
    double totient = (prime1-1)*(prime2-1);

    //public key
    double encrypt=2;

    //for checking co-prime which satisfies encrypt>1
    while(encrypt<totient){
    count = gcd(encrypt,totient);
    if(count==1)
        break;
    else
        encrypt++;
    }

    //private key
    double decrypt;

    //arbitrary value
    double arbitrary = 2;
```

Code Implementation in C++

Choosing Decrypt

```
// choosing decrypt such that it satisfies
// decrypt*encrypt = 1 + arbitrary * totient
decrypt = (1 + (arbitrary*totient))/encrypt;
double msg = 12;
double c = pow(msg,encrypt);
double m = pow(c,decrypt);
c=fmod(c,n);
m=fmod(m,n);

cout<<"Message data = "<<msg;
cout<<"\n"<<"prime1 = "<<prime1;
cout<<"\n"<<"prime2 = "<<prime2;
cout<<"\n"<<"n = pq is "<<n;
cout<<"\n"<<"totient = "<<totient;
cout<<"\n"<<"encrypt = "<<encrypt;
cout<<"\n"<<"decrypt = "<<decrypt;
cout<<"\n"<<"Encrypted data = "<<c;
cout<<"\n"<<"Original Message = "<<m;

return 0;
```

Code Output

```
Message data = 12
prime1 = 3
prime2 = 7
n = pq is 21
totient = 12
encrypt = 5
decrypt = 5
Encrypted data = 3
Original Message = 12
```

Thank you!