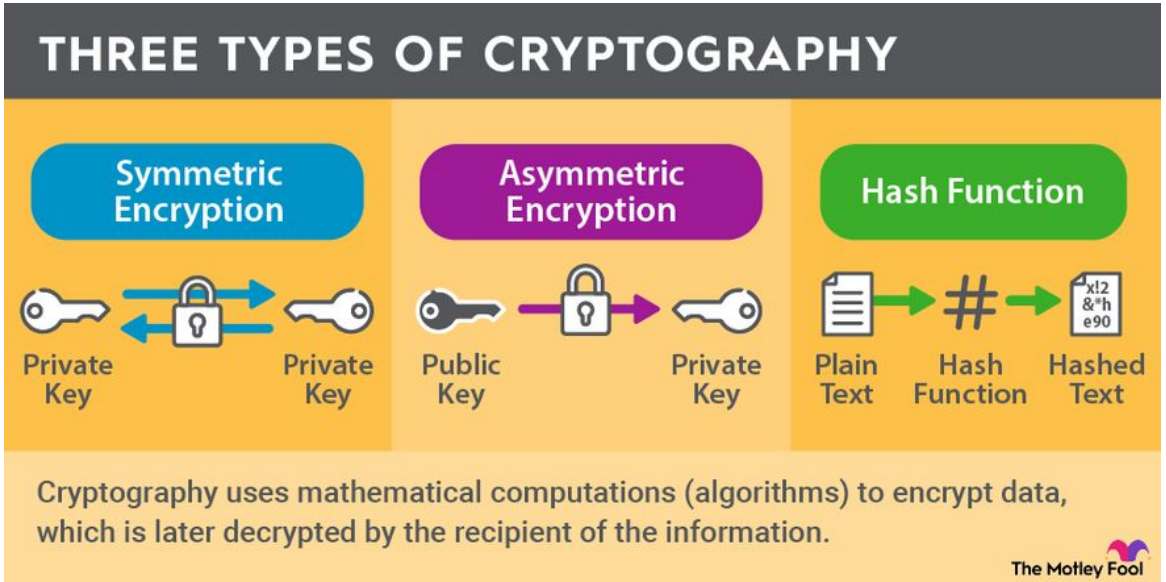
A Brief Introduction to RSA



Hash function

- Solves the problem of **data integrity**

Symmetric Encryption

- Solves the problem of **confidentiality** and **fast encryption** of data

Asymmetric Encryption

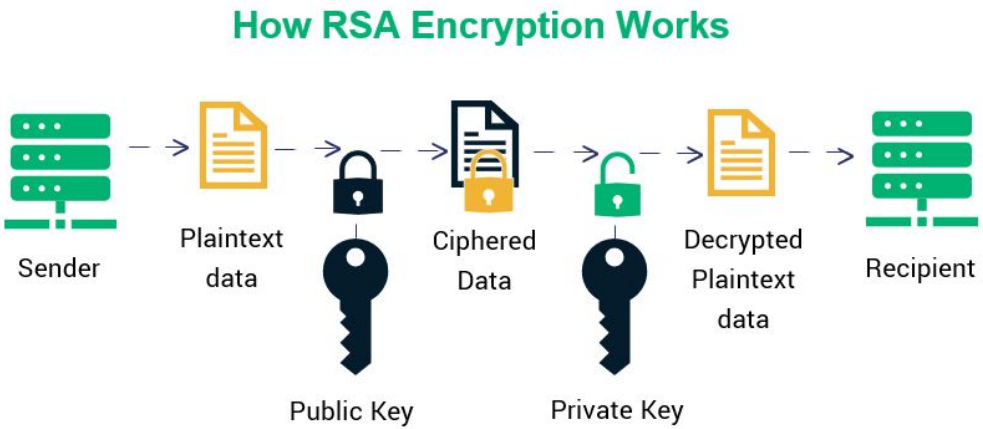
- Solves the problem of **secure key exchange** and enables **authentication** via digital signatures

Introduction to RSA

- RSA stands for **Rivest-Shamir-Adleman**.

- It is one of the first widely used **public key cryptographic** systems.

- RSA is used to secure sensitive data, particularly in the context of communication, digital signatures, and encryption.



RSA Applications

- **Data Privacy**: RSA ensures that sensitive information can be transmitted securely, even over insecure channels.

- **Authentication**: RSA-based digital signatures allow verification of the identity of the sender.

- Widespread Use:

    - **SSL/TLS protocols**

    - **email encryption** (PGP)

    - **digital certificates**

RSA core idea

Two keys: **Public and Private**

Public key: Used for **encryption**

Private key: Used for **decryption**

Core Idea:

- What's encrypted with the public key can only be decrypted with the private key.
- The public key can be shared openly, while the private key must remain secret.

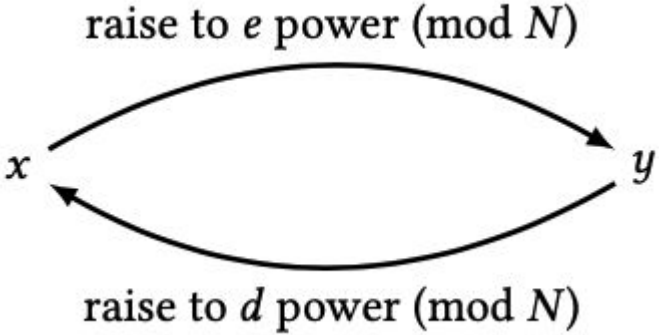
Modulo %

- 1 mod 3 = 1
- 2 mod 3 = 2
- 3 mod 3 = 0
- 4 mod 3 = 1
- 5 mod 3 = 2
- 6 mod 3 = 0

Encryption and decryption

Public Key: (e, n) = (17, 3233)

Private Key: (d, n) = (2753, 3233)



Encryption

- Message x = 65
- By using public key, (sage: 65 ^ 17 % 3233)
- Ciphertext $y = x^e \bmod N = 65^{17} \bmod 3233 = 2790$

Decryption

- Ciphertext y = 2790
- By using private key, (sage: 2790 ^ 2753 % 3233)
- Decrypted Message $x = y^d \bmod N = 2790^{2753} \bmod 3233 = 65$
- We've successfully retrieve the original message x = 65 !!!

Multiplicative Inverses

The multiplicative inverse of x mod n is the integer y that satisfies $x * y \equiv_n 1$

Ex:

- $2 * 8 \equiv_{15} 1$
- $3 * 8 \equiv_{15} 9$

Consider mod 15, and x = 4, what will y be?

$4 * y \equiv_{15} 1$

$\Rightarrow$ ans: y = 4, because $4 * 4 \equiv_{15} 1$

Multiplicative group modulo

$Z_N^* = \{x \in Z_N \mid x \text{ has a multiplicative inverse mod } n\}$

- How do we determine which number has a multiplicative inverse mod n
- Theorem: x has a multiplicative inverse mod n **if and only if** gcd(x, N)=1

Example:

$Z_{15} = \{0, 1, \ldots, 14\}$

To find Z_{15}^*, we exclude any of the numbers that share a common factor with 15

$\Rightarrow Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

$(1 * 1 \equiv_{15} 1)$, $(2 * 8 \equiv_{15} 1)$, (~~3~~), $(4 * 4 \equiv_{15} 1)$, (~~5~~) , ...

Sage: 1 ^ -1 % 15, 5 ^ -1 % 15

Euler's totient function

$\phi(N) = |Z_N^*|$

- Number of element in the Z_n^* group

$n = p * q$, $\phi(N) = (p-1)(q-1)$

Ex: $p = 3, q = 5 \Rightarrow n = 15$

$\Rightarrow \phi(15) = (2)(4) \Rightarrow 8$

Euler's Theorem

If $x \in Z_N^*$, then $x^{\phi(N)} \equiv_N 1$

Ex: if N = 15, then $\phi(N) = 8$

By euler's theorem: for all $x \in Z_{15}^*$, $x^8 \equiv_{15} 1$

Sage: 1 ^ 8 % 15, 13 ^ 8 % 15

$Z_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

Euler's Theorem

If $x \in Z_N^*$, then $x^{\phi(N)} \equiv_N 1$

Ex: if N = 15, then $\phi(N) = 8$

By euler's theorem: for all $x \in Z_{15}^*$, $x^8 \equiv_{15} 1$

How is this useful?

- Pick e and d such that $ed \equiv_{\phi(N)} 1$
- $\Rightarrow ed = t(\phi(N)) + 1$

$x^{e(d)} = x^{ed} = x^{t\phi(N)+1} = (x^{\phi(N)})^t x \equiv_N (1)^t x = x$

Where's the public key and private key from?

1. Choose two prime numbers
   - p = 61, q = 53

2. Compute N
   - N = p * q = 61 * 53 = 3233

3. Compute Euler's Totient function ϕ(N)
   - ϕ(N) = (p−1)×(q−1) = (61 - 1) * (53 * 1) = 60 * 52 = 3120

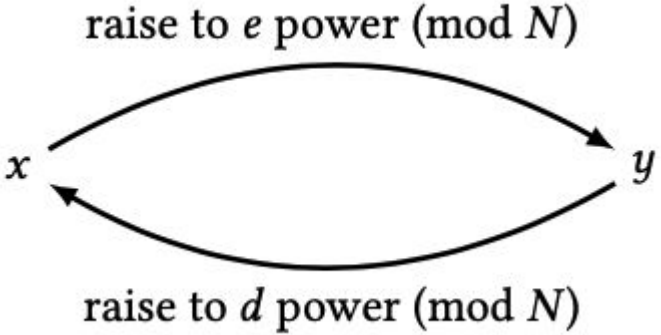
4. Choose e
   - such that 1 < e < ϕ(N) and e is coprime with ϕ(N)
   - Let's say e = 17
   - e becomes part of the public key(e, N), will be used for encryption

5. Compute d
   - such that e * d ≡ 1 (mod ϕ(N))
   - sage: 1/Mod(17, 3120)
   - d = 2753
   - d becomes part of the private key (d, N), will be used for decryption

Encryption and decryption

Public Key: (e, N) = (17, 3233)

Private Key: (d, N) = (2753, 3233)



Encryption

- Message x = 65
- By using public key, (sage: 65 ^ 17 % 3233)
- Ciphertext $y = x^e \bmod N = 65^{17} \bmod 3233 = 2790$

Decryption

- Ciphertext y = 2790
- By using private key, (sage: 2790 ^ 2753 % 3233)
- Decrypted Message $x = y^d \bmod N = 2790^{2753} \bmod 3233 = 65$
- We've successfully retrieve the original message x = 65 !!!

Sage demo

```
sage: p = random_prime(10^5)
sage: q = random_prime(10^5)
sage: N = p*q
sage: phi = (p-1)*(q-1)
sage: e = random_prime(phi)
sage: d = e ^ -1 % phi
sage: x = 31415926
sage: y = x^e % N (alternatively, y = Mod(x, N) ^ e)
sage: y^d % N
```

Security Considerations

Key size

- the numbers involved in RSA need to be quite large to resist **factoring attacks**
- Current best practices suggest to use 2048- or 4096-bit RSA moduli, meaning that p and q are each **1024 or 2048 bits**

Hacking demo:

- **e and N are known to others**, (17, 3233)

Sage: factor(3233)
Output: 53 * 61
Sage: phi = 52 * 60
Output: 3120

- Recall **e * d ≡ 1 (mod ϕ(n)) ⇒ d = 1 / Mod(e, ϕ(n))**

Sage: q = 1 / Mod(17, 3120)
Output: 2753
=== We found the private key q !!! ===

References

Books

- Serious Cryptography
- The Joy of Cryptography
- A Graduate Course in Applied Cryptography

Tools

- https://www.sagemath.org/
- https://sagecell.sagemath.org/