ST.MARTIN'S ENGINEERING COLLEGE

(UGC Autonomous)

Dhulapally, Near Kompally,Medchal–Malkajgiri district
Secunderabad-500100,Telangana, India

Department Artificial Intelligence and Data Science

Video Lecturing

on

R S A

ALGORITHM

AGENDA

01 What is the RSA algorithm, and how does it work?

02 Introduction RSA Algorithm in Cryptography

03 Example of Asymmetric Cryptography

04 RSA Algorithm-Key Components

05 Example of RSA Algorithm

06 Idea behind RSA Algorithm

07 Advantages

08 Disadvantages

09 Applications of RSA

10 If vulnerable to quantum computing, we can ensure the security of RSA encryption

01

What is the RSA algorithm, and how does it work?

What is the RSA algorithm, and how does it work?

The **RSA algorithm** is an **asymmetric encryption technique** that utilizes a pair of keys a **public key** and a **private key** to protect data. It is based on the **mathematical complexity** of **factoring large integers**, making it **extremely challenging for unauthorized** entities to **decipher** the encrypted information.

02

Introduction of RSA Algorithm in Cryptography

Introduction of RSA Algorithm in Cryptography

The **RSA** **(Rivest-Shamir-Adleman)** algorithm is a widely used method in public-key cryptography. It employs a pair of keys one for encryption and the other for decryption ensuring secure communication. The strength of RSA lies in the complexity of factoring large prime numbers, making unauthorized access highly impractical. Developed in 1977 by Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman, this algorithm remains a cornerstone of modern cryptographic systems.

03

Example of Asymmetric Cryptography

Example of Asymmetric Cryptography

If **Sender A** wishes to transmit a message securely to **Receiver B**:

- Sender A encodes the message using Receiver B's Public Key.
- Receiver B decodes the message using their Private Key.

04

RSA Algorithm-Key Components

RSA Algorithm-Key Components

RSA Algorithm is based on **factorization** of large number and **modular arithmetic** for encrypting and decrypting data. It consists of three main stages:

1. **Key Generation:** Creating Public and Private Keys
2. **Encryption:** Sender encrypts the data using Public Key to get **cipher text.**
3. **Decryption:** Decrypting the **cipher text** using Private Key to get the original data.

Key Generation

1. Choose two large prime numbers, say **p** and **q**. These prime numbers should be kept secret.
2. Calculate the product of primes, **n = p * q**. This product is part of the public as well as the private key.
3. Calculate Euler Totient Function **Φ(n)** as **Φ(n)** = **Φ(p * q)** = **Φ(p) * Φ(q)** = **(p – 1) * (q – 1).**
4. Choose encryption exponent **e**, such that
   - $1 < e < \Phi(n)$, and
   - $\gcd(e, \Phi(n)) = 1$, that is e should be co-prime with Φ(n).
5. Calculate decryption exponent **d,** such that
   - **$(d * e) \equiv 1 \bmod \Phi(n)$**, that is d is modular multiplicative inverse of **e** mod Φ(n).

Finally, the **Public Key = (n, e)** and the **Private Key = (n, d)**.

Encryption

To encrypt a message **M**, it is first converted to numerical representation using ASCII and other encoding schemes. Now, use the public key (n, e) to encrypt the message and get the cipher text using the formula:

$$C = M^e \bmod n,$$

where **C** is the Cipher text and

e and **n** are parts of public key.

Decryption

To decrypt the cipher text C, use the private key (n, d) and get the original data using the formula:

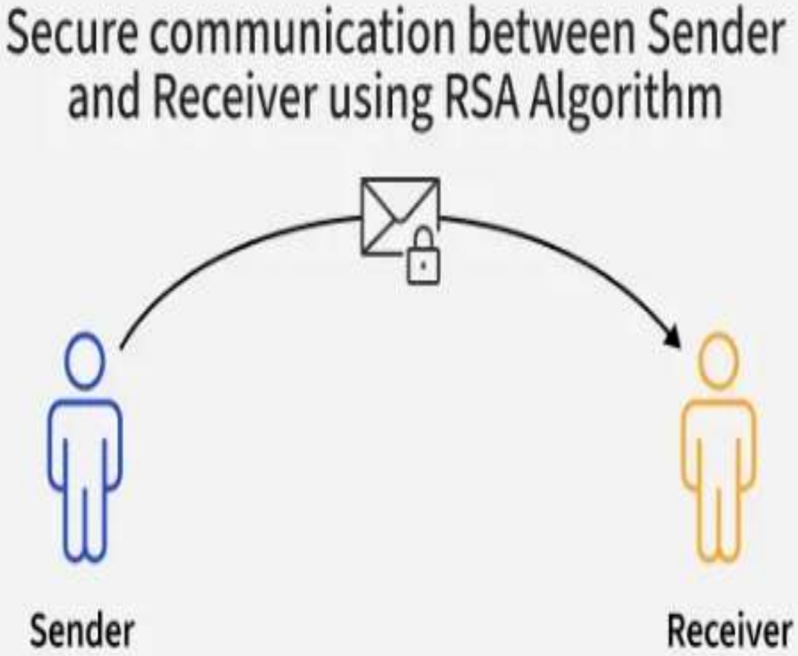
$M = C^d \bmod n,$

where **M** is the message and

d and **n** are parts of private key.

05

Example of RSA Algorithm



RSA Encryption and Decryption Example

01
Step

Key Generation

Choose two prime numbers: $p = 3, q = 11$

Calculate $n = p * q = 33$

Calculate Euler's Totient Function: $\Phi(33) = \Phi(3) * \Phi(11) = 2 * 10 = 20$

Choose $e = 7$, which is co-prime with 20

Calculate d as the multiplicative inverse of e (7), so $d = 3$

Public Key = $(n, e) = (33, 7)$ Private Key = $(n, d) = (33, 3)$

RSA Encryption and Decryption Example

02 Step | Sharing of Public Key

Public Key = $(n, e) = (33, 7)$ Private Key = $(n, d) = (33, 3)$

The Public Key is shared with the Sender and the Private Key is kept secret with the Receiver .

Sender

$(n, e) = (33, 7)$

Receiver

$(n, d) = (33, 3)$

RSA Encryption and Decryption Example

03 Step

Encryption - Message Conversion

Sender's Message(M) = "AC"

$\downarrow$ Numeric Conversion (A - Z => 1 - 26)

13

Numeric Representation of "AC"

04 Step

Encryption Formula:

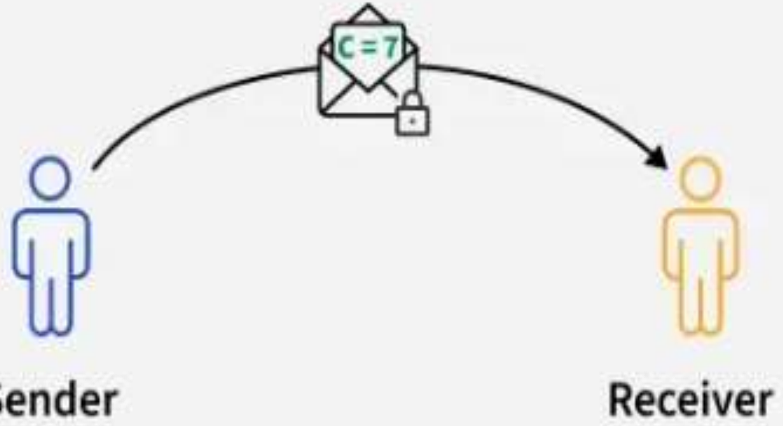
Encrypt the message using the Public Key (33, 7)

$$\text{Cipher Text } C = M^e \bmod n$$

$C = 13^7 \bmod 33$

$C = 62748517 \bmod 33$

$C = 7$



RSA Encryption and Decryption Example

05 Step

Decryption Formula:

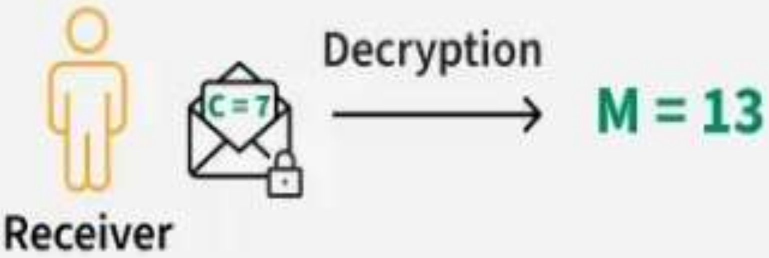
Decrypt the Cipher text using the Private Key (33, 3)

$$\text{Decrypted Text } M = C^d \bmod n$$

$$M = 7^3 \bmod 33$$

$$M = 343 \bmod 33$$

$$M = 13$$



The receiver uses decrypted text M = 13 to get the original message = "AC".

RSA Encryption and Decryption Example

06

Idea behind RSA Algorithm

Idea behind RSA Algorithm

- The RSA algorithm is based on the concept that factorizing a large integer is computationally difficult.
- The **Public Key** consists of two values: **(n, e)**, where both **n** and **e** are publicly known.
- The **Private Key** consists of **(n, d)**, where only the receiver knows the value of **d**.
- Since only the receiver has **d**, they are the only one capable of decrypting the message.
- To find **d** using **n** and **e**, we use the relationship: $(d \times e) \equiv 1 \mod \Phi(n)$
- If we can compute **Φ(n)**, we can determine **d**.
- **Φ(n)** is calculated as: $\Phi(n) = (p - 1) \times (q - 1)$

- To find **Φ(n)**, the values of **p** and **q** are needed.
- Although **n** is publicly known and equals **p × q**, finding **p** and **q** is challenging because RSA uses very large prime numbers for **p** and **q**.
- As a result, **n** becomes extremely large, making factorizing **n** computationally infeasible.
- The strength of RSA encryption relies on the size of **p** and **q**.
- Typically, RSA keys are **1024** or **2048** bits long.
- Experts suggest that **1024-bit keys** might be breakable in the near future.
- However, as of now, breaking RSA encryption remains practically impossible.

Note: *If someone gets to know the value of **p** and **q**, then he can calculate the value of **d** and decrypt the message.*

07

Advantages

Advantages

- **Security:** RSA is highly secure and widely used for safe data transmission.
- **Public-key cryptography:** It uses two keys — a public key for encryption and a private key for decryption.
- **Key exchange:** RSA enables secure key exchange without transmitting the secret key directly.
- **Digital signatures:** It allows message signing with a private key and verification with a public key.
- **Applications:** RSA is extensively used in online banking, e-commerce, and secure communications.

08

Disdvantages

Limitations of RSA Algorithm:

1. **Slow processing speed:** Inefficient for encrypting large data volumes.
2. **Large key size:** Requires significant computational resources and storage.
3. **Side-channel vulnerabilities:** Susceptible to attacks exploiting power consumption, electromagnetic leaks, and timing analysis.
4. **Limited applicability:** Unsuitable for applications needing constant encryption/decryption of large data.
5. **Complexity:** Involves advanced mathematics, making it hard to grasp and implement.
6. **Key management:** Secure handling of the private key can be challenging.

09

Applications of RSA

RSA finds applications in:

- ***Secure Data Transmission:*** *Encrypting data for confidentiality.*
- ***Key Exchange:*** *Establishing shared secret keys.*
- ***Digital Signatures:*** *Verifying the authenticity and integrity of messages.*

10

If vulnerable to quantum computing, we can ensure the security of RSA encryption

Yes, RSA is vulnerable to quantum computing attacks, as quantum computers could efficiently factor large numbers, undermining the algorithm's security. However, ongoing research aims to develop quantum-resistant cryptographic algorithms.

To enhance RSA security:

Use Strong Keys: Employ sufficiently large key sizes (e.g., 2048 bits or higher).

Protect Private Keys: Store private keys securely and avoid sharing them.

Implement Side-Channel Attack Mitigations: Use techniques to reduce information leakage.

Stay Informed: Keep updated on advancements in cryptography and vulnerabilities.

THANKS TO ALL

Suggestions and Feedback