Perform Encryption And Decryption Using The RSA Algorithm For The Following: A. P = 3; Q = 11, E = 7;

Perform Encryption And Decryption Using The RSA Algorithm For The Following: A. P = 3; Q = 11, E = 7;

The RSA algorithm is one of the most widely used public-key cryptographic systems, enabling secure communication over insecure channels. It leverages the mathematical properties of prime numbers and modular arithmetic to facilitate encryption and decryption. In this article, we will walk through the process of performing RSA encryption and decryption using the specific parameters: P = 3, Q = 11, and E = 7. This example provides a practical understanding of how RSA works step-by-step, from key generation to encrypting and decrypting a message.

Understanding the RSA Algorithm

RSA (Rivest-Shamir-Adleman) is an asymmetric cryptographic algorithm that involves a pair of keys: a public key for encryption and a private key for decryption. The core idea is that messages encrypted with the public key can only be decrypted with the corresponding private key, ensuring confidentiality.

Key Components of RSA

- **Prime Numbers (P and Q):** Two large prime numbers used to generate the keys.
- **Modulus (N):** The product of P and Q, used as the modulus for both encryption and decryption.
- **Euler's Totient Function (ϕ(N)):** Calculated as (P − 1) (Q − 1), used in key generation.
- **Public Exponent (E):** An integer typically chosen to be coprime with ϕ(N); used for encryption.
- **Private Exponent (D):** The modular multiplicative inverse of E modulo ϕ(N); used for decryption.

Step 1: Generate the RSA Keys

Using the provided parameters, we will generate the necessary keys for encryption and decryption.

Calculating N (the modulus)

Given P = 3 and Q = 11:

N = P Q = 3 11 = 33

This value is used in both the public and private keys.

Calculating Euler's Totient Function ϕ(N)

ϕ(N) = (P − 1) (Q − 1) = (3 − 1) (11 − 1) = 2 10 = 20

This value is essential for determining the private key.

Choosing Public Exponent E

Given E = 7, we verify that E is coprime with ϕ(N):

gcd(7, 20) = 1

Since they are coprime, E = 7 is suitable.

Calculating the Private Exponent D

D is the modular inverse of E modulo ϕ(N):

Find D such that:

E D ≡ 1 (mod 20)

To find D, we solve:

7 D ≡ 1 (mod 20)

Using the Extended Euclidean Algorithm:

- 20 = 2 7 + 6
- 7 = 1 6 + 1
- 6 = 6 1 + 0

Back substitution:

1 = 7 − 1 6

But 6 = 20 − 2 7, so:

1 = 7 − 1 (20 − 2 7) = 7 − 1 20 + 2 7 = (1 + 2) 7 − 1 20 = 3 7 − 1 20

Thus,

3 7 ≡ 1 (mod 20)

This means D = 3.

Therefore, the private key exponent D = 3.

Step 2: Public and Private Keys

- Public Key (e, N): (7, 33)
- Private Key (d, N): (3, 33)

These keys will be used for encryption and decryption.

Step 3: Encrypting a Message

Suppose the message to be encrypted is 'M'. For simplicity, consider the message as a numerical value within the range of N (0 to 32). Let's encrypt the message M = 4.

The encryption process:

C = M^E mod N

Calculating:

C = 4^7 mod 33

Let's compute 4^7 step-by-step:

- 4^1 = 4
- 4^2 = 16
- 4^3 = 16 4 = 64 ≡ 64 mod 33 = 64 − 331= 64 − 33=31
- 4^4 = 31 4 = 124 ≡ 124 − 333=124 − 99=25
- 4^5 = 25 4=100 ≡ 100 − 333=100 − 99=1
- 4^6 = 1 4=4
- 4^7 = 4 4=16

Alternatively, to be precise, using modular exponentiation:

4^7 mod 33:

- 4^1 = 4
- 4^2 = 16
- 4^3 = 16 4=64 ≡ 64 − 331=31
- 4^4 = 31 4=124 ≡ 124 − 333=124 − 99=25
- 4^5 = 25 4=100 ≡ 100 − 333=100 − 99=1
- 4^6 = 1 4=4
- 4^7=4 4=16

So, C = 16.

Encrypted message: 16

Step 4: Decrypting the Ciphertext

To retrieve the original message, use the private key exponent D = 3:

M = C^D mod N

Calculating:

M = 16^3 mod 33

Compute:

- 16^1=16
- 16^2=1616=256 ≡ 256 - 733=256 - 231=25
- 16^3=2516=400 ≡ 400 - 1233=400 - 396=4

Thus, the decrypted message is 4, which matches the original message.

Summary of the RSA Process

Step	Description	Result
1	Generate N = P Q	33
2	Calculate ϕ(N)	20
3	Choose E = 7	Valid (coprime with 20)
4	Compute D (private exponent)	3
5	Public key	(7, 33)
6	Private key	(3, 33)
7	Encrypt message M=4	Ciphertext C=16
8	Decrypt ciphertext C=16	Original message M=4

Practical Implications and Applications of RSA

RSA's strength lies in the difficulty of factoring large composite numbers, making it a reliable method for secure communication. While this example uses small primes for simplicity, real-world RSA implementations use very large primes—often hundreds or thousands of bits long—to ensure security.

Applications include:

- Secure email communication
- Digital signatures
- Secure web browsing (SSL/TLS)
- Cryptographic protocols

Conclusion

This detailed walkthrough demonstrates the fundamental process of performing encryption and decryption using the RSA algorithm with specific parameters: P = 3, Q = 11, and E = 7. By calculating the necessary components, generating keys, and executing modular exponentiation for encryption and decryption, we observe the core principles that make RSA a cornerstone of modern cryptography. Whether for securing sensitive data or authenticating digital signatures, understanding the RSA process is essential for anyone interested in cybersecurity.

Remember, while small primes simplify the illustration, real-world RSA relies on large primes to ensure security. Nonetheless, this example provides a clear foundation for understanding how RSA encrypts and decrypts messages, emphasizing the importance of key generation, modular arithmetic, and the mathematical elegance behind public-key cryptography.

Frequently Asked Questions

What are the steps to perform RSA encryption and decryption with P=3, Q=11, and E=7?

First, compute N = P Q = 3 11 = 33. Then, calculate ψ(N) = (P−1)(Q−1) = 2 10 = 20. Find the public key exponent E=7. Next, determine the private key D such that (D E) mod ψ(N) = 1; here, D=3. To encrypt a message M, compute ciphertext C = M^E mod N; to decrypt, compute M = C^D mod N.

How do you calculate the private key D in RSA with given P=3, Q=11, and E=7?

Calculate ψ(N) = (P−1)(Q−1) = 20. Find D such that (D E) mod ψ(N) = 1. Using the extended Euclidean algorithm, D=3 because (3 7) mod 20 = 21 mod 20 = 1.

What is the public key in this RSA setup?

The public key is composed of the modulus N=33 and the exponent E=7, represented as (N=33, E=7).

How do you encrypt a message, say M=2, using the RSA parameters P=3, Q=11, E=7?

Calculate ciphertext C = M^E mod N = 2^7 mod 33. 2^7=128; 128 mod 33=128−333=128−99=29. So, the encrypted message is C=29.

How is decryption performed to retrieve the original message from ciphertext C=29?

Compute M = C^D mod N = 29^3 mod 33. 29^3=292929=24389. Now, 24389 mod 33: 33739=24407; since 24407−24389=18, the remainder is 18. However, since 18 does not match original message 2, check calculations: better to compute stepwise: 29^2=841; 841 mod 33=841−2533=841−825=16; then 29^3=2916=464; 464

mod 33=464−1433=464−462=2. Therefore, the original message M=2.

Why is RSA considered secure with the given parameters P=3, Q=11, and E=7?

RSA's security relies on the difficulty of factoring large composite numbers. With small parameters like P=3 and Q=11, RSA is not secure for real-world use, but for educational purposes, it demonstrates the encryption-decryption process effectively.

What are common pitfalls to avoid when performing RSA encryption and decryption manually?

Common pitfalls include miscalculating modular exponentiation, incorrect computation of the private key D, and errors in modular reduction steps. Always verify calculations step-by-step and use proper algorithms like the extended Euclidean algorithm for finding D.

Additional Resources

RSA Encryption and Decryption: A Step-by-Step Analysis Using Sample Values

In the realm of modern cryptography, the RSA algorithm stands as a cornerstone of secure digital communication. Its ability to encrypt and decrypt messages using a pair of keys—the public key for encryption and the private key for decryption—makes it indispensable for safeguarding sensitive information. This article delves into the practical application of RSA encryption and decryption, illustrating the process with specific parameters: P = 3, Q = 11, and E = 7. We will explore each step in detail, providing clarity on the underlying mathematics and the logic that makes RSA a robust cryptographic method.

Understanding the Foundations of RSA

What is RSA Encryption?

RSA (Rivest–Shamir–Adleman) encryption is an asymmetric cryptographic algorithm that allows secure communication without the need for a shared secret key beforehand. It relies on the mathematical difficulty of factoring large composite numbers into their prime factors, which underpins its security. The core idea involves generating a pair of keys:

- Public Key (e, n): Used for encrypting messages.
- Private Key (d, n): Used for decrypting messages.

Anyone can use the public key to encrypt data, but only the holder of the private key can decrypt it, ensuring confidentiality.

Key Generation Process Overview

The process involves several key steps:

1. Choose two distinct prime numbers, P and Q.
2. Compute n = P Q. (This is the modulus for both keys.)
3. Calculate Euler's Totient function, ψ(n) = (P − 1) (Q − 1).
4. Select an encryption exponent E such that 1 < E < ψ(n), and E is coprime with ψ(n).
5. Find the decryption exponent D, which is the modular multiplicative inverse of E modulo ψ(n).

In our specific scenario, P = 3, Q = 11, and E = 7. Let's analyze how these values fit into the RSA framework.

Step-by-Step Construction of Keys Using Given Values

1. Calculate n = P Q

- P = 3
- Q = 11

Thus,

n = 3 11 = 33

The modulus n = 33 will be used in both the public and private keys.

2. Compute Euler's Totient Function, ψ(n)

ψ(n) = (P − 1) (Q − 1) = (3 − 1) (11 − 1) = 2 10 = 20

This value signifies the count of integers less than n that are coprime to n.

3. Verify the Chosen E (E = 7) Is Suitable

E must satisfy two conditions:

- 1 < E < ψ(n)
- E is coprime with ψ(n)

Check if E = 7 is coprime with 20:

- gcd(7, 20) = 1 (since 7 and 20 share no common divisors other than 1)

Since the gcd is 1, E = 7 is suitable.

4. Compute the Decryption Exponent D

D is the modular inverse of E modulo $\varphi(n)$:

$D \equiv E^{-1} \bmod \varphi(n)$

which means:

$D\,E \equiv 1 \bmod 20$

To find D:

- Use the Extended Euclidean Algorithm to compute D such that:

$7\,D \equiv 1 \bmod 20$

Let's find D:

- Try D = 3:

$7 \cdot 3 = 21 \equiv 1 \bmod 20$

Since 21 mod 20 = 1, D = 3 satisfies the condition.

Therefore, D = 3.

Constructing the Public and Private Keys

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

These keys form the core of RSA encryption and decryption processes.

Encrypting a Message Using RSA

Suppose the message to encrypt is represented numerically as M. For simplicity, let's assume M = 5. The encryption process involves computing:

$C \equiv M^E \bmod n$

where:

- C = ciphertext
- M = message
- E = public exponent (7)
- n = modulus (33)

Calculating:

$C = 5^7 \bmod 33$

Let's perform modular exponentiation efficiently.

Performing Modular Exponentiation

Calculating 5^7 mod 33:

- 5^1 ≡ 5 mod 33
- 5^2 ≡ 5 5 = 25 mod 33
- 5^3 ≡ 25 5 = 125 mod 33

125 mod 33:

125 ÷ 33 = 3 33 = 99, remainder 125 − 99 = 26

So, 5^3 ≡ 26 mod 33

- 5^4 ≡ 26 5 = 130 mod 33

130 ÷ 33 = 3 33 = 99, remainder 130 − 99 = 31

So, 5^4 ≡ 31 mod 33

- 5^5 ≡ 31 5 = 155 mod 33

155 ÷ 33 = 4 33 = 132, remainder 155 − 132 = 23

So, 5^5 ≡ 23 mod 33

- 5^6 ≡ 23 5 = 115 mod 33

115 ÷ 33 = 3 33 = 99, remainder 115 − 99 = 16

So, 5^6 ≡ 16 mod 33

- 5^7 ≡ 16 5 = 80 mod 33

80 ÷ 33 = 2 33 = 66, remainder 80 − 66 = 14

Thus,

C ≡ 14 mod 33

Encrypted message: 14

Decrypting the Ciphertext to Retrieve the Original Message

Decryption involves raising the ciphertext C to the power of D modulo n:

M ≡ C^D mod n

Given:

- C = 14
- D = 3
- n = 33

Calculate:

M = 14^3 mod 33

Again, perform modular exponentiation:

- 14^1 ≡ 14 mod 33
- 14^2 ≡ 14 14 = 196 mod 33

196 ÷ 33 = 5 33 = 165, remainder 196 − 165 = 31

So, 14^2 ≡ 31 mod 33

- 14^3 ≡ 31 14 = 434 mod 33

434 ÷ 33 = 13 33 = 429, remainder 434 − 429 = 5

Thus,

M ≡ 5 mod 33

The decrypted message is 5, which matches the original message, confirming the correctness of the RSA process with the chosen parameters.

Implications and Security Considerations

The example above demonstrates the fundamental mechanics of RSA encryption and decryption with small, manageable numbers. While educational, these small primes (3 and 11) are not suitable for real-world security, where primes are typically hundreds or thousands of bits long to prevent factorization attacks.

The security of RSA hinges on the difficulty of factoring large composite numbers:

- Prime selection: Larger primes increase security exponentially.
- Key size: Modern standards recommend key sizes of 2048 bits or more.
- Mathematical complexity: Factoring a large number like n becomes computationally infeasible with current technology.

This example, however, serves as a valuable educational tool, illustrating the mathematical steps involved and providing insight into how RSA fundamentally works.

Conclusion

The process of performing encryption and decryption using the RSA algorithm is rooted in prime number arithmetic and modular inverses. By selecting appropriate primes and exponents, generating a key pair, and applying modular exponentiation, users can securely encode and decode messages.

In this detailed walkthrough with P = 3, Q = 11, and E = 7, we validated the entire process:

- Calculated n and ψ(n)
- Confirmed E's suitability
- Determined the decryption key D
- Encrypted a sample message
- Decrypted the ciphertext to retrieve the original message

This example encapsulates the core principles of RSA and underscores its elegance and robustness as a cryptographic method. As technology advances, RSA remains a fundamental component of digital security, securing communications, digital signatures, and more. Understanding its inner workings, even through simplified examples, is essential for appreciating the sophistication and importance of cryptography in our digital age.

Perform Encryption And Decryption Using The Rsa Algorithm For The Following A P 3 Q 11 E 7

Perform Encryption And Decryption Using The Rsa Algorithm For The Following A P 3 Q 11 E 7

Related Articles

- Read Friar Laurences Dialogue From Act V, Scene Iii Of Romeo And Juliet. But He Which Bore My Letter,
- Modify Array Elements Using Other Elements. ACTIVITY Write A Loop That Sets Each Array Element To The
- Question 1 (2 Points)Rank In Order, From Smallest To Largest, The Gravitational Potential Energies Of

Back to Home