RSA

The Art of Cryptography

Securing the Digital World with Prime Numbers

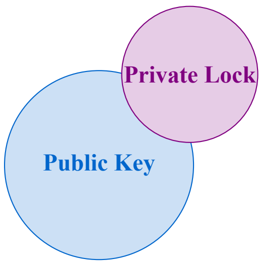
By Omar Hany, Jassmin Nasser, Bassant Mohamed,Basmala Salah

Discrete Mathematics - MATH 205 | Spring 2025

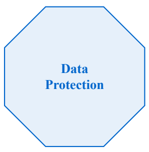
Cryptography transforms data into secrets, and RSA is its shining star, using the magic of prime numbers to lock and unlock our digital world.

1 Introduction to Cryptography

Cryptography, derived from the Greek words *kryptós* (hidden) and *gráphein* (writing), is the art and science of securing information. By transforming data into unreadable formats, it ensures **confidentiality**, **integrity**, **authenticity**, and **non-repudiation**. From ancient ciphers like the Caesar Cipher to modern systems like RSA, cryptography has evolved to meet the demands of a digital age.



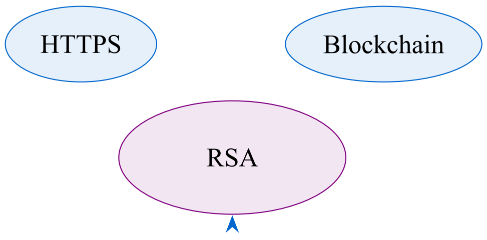
1.1 Historical Context



Cryptography dates back to ancient Egypt, where hieroglyphs concealed messages. The Caesar Cipher, used by Julius Caesar, shifted letters by a fixed number (e.g., A→D for a shift of 3). Today, RSA (Rivest-Shamir-Adleman), introduced in 1977, leverages mathematical complexity to secure digital communications.

1.2 Why RSA?

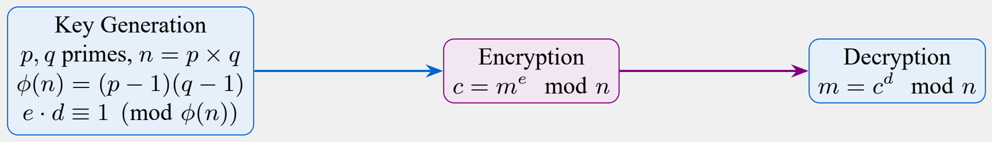
- **Scalability**: Enables secure communication over open networks like the internet.
- **Versatility**: Powers HTTPS, VPNs, digital signatures, and blockchain.
- **Strength**: Relies on the near-impossible task of factoring large prime numbers.



figureRSA's Role in Modern Applications

2 The Mathematics of RSA

RSA's strength lies in modular arithmetic and prime number theory. Below are the core steps, enriched with mathematical rigor.



figureRSA Process Flow

2.1 Step-by-Step Breakdown

1. **Key Generation**:
   - Choose two large prime numbers p and q (e.g., $p = 13$, $q = 11$).
   - Compute $n = p \times q = 143$ and $\phi(n) = (p-1)(q-1) = 120$.
   - Select e (public exponent) coprime with $\phi(n)$, e.g., $e = 7$.
   - Compute d (private exponent) such that $e \cdot d \equiv 1 \pmod{\phi(n)}$, e.g., $d = 103$.
   - Public key: $(e, n) = (7, 143)$; Private key: $(d, n) = (103, 143)$.
2. **Encryption**:

- Convert message to a number m (e.g., $m = 12$ for 'L').
- Compute ciphertext $c = m^e \mod n = 12^7 \mod 143 = 67$.

3. **Decryption**:

- Compute message $m = c^d \mod n = 67^{103} \mod 143 = 12$.
- Convert back to text (e.g., $12 \to$ 'L').

2.2 Mathematical Foundation

RSA relies on Euler's theorem: for coprime m and n, $m^{\phi(n)} \equiv 1 \pmod{n}$. The security stems from the difficulty of factoring n into p and q, a problem believed to be computationally infeasible for large numbers.

3 RSA in Code

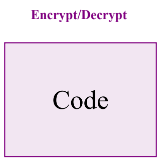
Below is a Python implementation of RSA, demonstrating key generation, encryption, and decryption.

```python
import random
from sympy import isprime, mod_inverse, gcd

def generate_prime(start=2, end=100):
    while True:
        num = random.randint(start, end)
        if isprime(num):
            return num

def generate_keys():
    p = generate_prime()
    q = generate_prime()
    while q == p:
        q = generate_prime()

    n = p * q
    phi = (p - 1) * (q - 1)

    e = 2
    while gcd(e, phi) != 1:
        e = random.randrange(2, phi)
    d = mod_inverse(e, phi)

    return ((e, n), (d, n))

def encrypt(message, public_key):
    e, n = public_key
    encrypted = []
    for char in message:
        num = ord(char)
        enc = pow(num, e, n)
        encrypted.append(enc)
    return encrypted

def decrypt(ciphertext, private_key):
    d, n = private_key
    decrypted = ''
    for num in ciphertext:
        dec = pow(num, d, n)
        char = chr(dec)
        decrypted += char
    return decrypted

# Example usage
public, private = generate_keys()
print("Public␣Key:", public)
print("Private␣Key:", private)

message = "Hello,␣RSA!"
encrypted = encrypt(message, public)
print("Encrypted:", encrypted)

decrypted = decrypt(encrypted, private)
print("Decrypted:", decrypted)
```

Example Output:

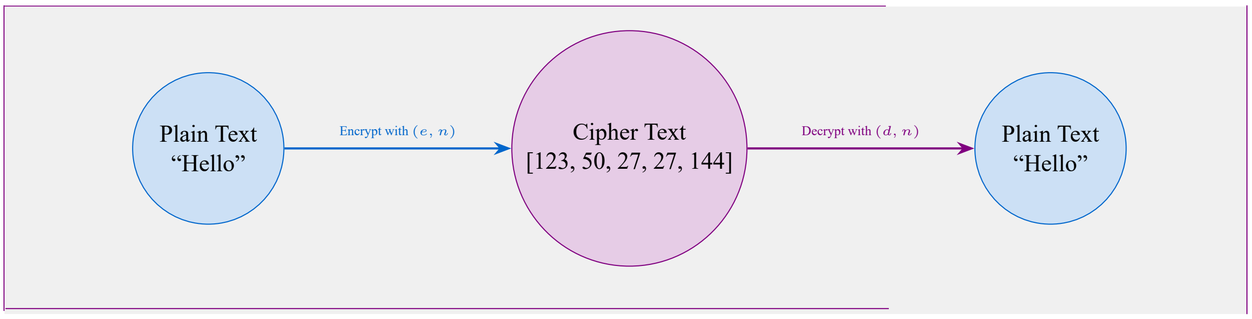
- Public Key: (109, 187)
- Private Key: (69, 187)
- Message: Hello
- Encrypted: [123, 50, 27, 27, 144]
- Decrypted: Hello

This code showcases RSA's practical application, converting text to numbers, encrypting with the public key, and decrypting with the private key.



4 Visualizing RSA

RSA transforms plain text into ciphertext and back, as illustrated below:



figureRSA Encryption and Decryption Flow

4.1 Real-World Impact

- **HTTPS**: Secures 90% of online transactions, protecting banking and e-commerce.
- **VPNs**: Ensures private communication over public networks.
- **Blockchain**: Validates transactions in cryptocurrencies like Bitcoin.
- **Fun Fact**: RSA key sizes (e.g., 2048 bits) make factoring infeasible with current technology.

5 Conclusion and Future Outlook

RSA is a triumph of mathematics, blending prime numbers and modular arithmetic to safeguard our digital world. Its elegance lies in its simplicity, yet its strength is unparalleled, securing everything from online banking to blockchain. As quantum computing emerges, RSA may face challenges, but post-quantum cryptography is poised to carry its legacy forward.

figureRSA: A Beacon of Digital Trust

References

[1] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[2] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). Handbook of applied cryptography. CRC press.

[3] Singh, S. (1999). The code book: the science of secrecy from ancient Egypt to quantum cryptography. Anchor.

In a world where data is currency, RSA stands as a timeless guardian, ensuring privacy and trust. Its legacy inspires the next generation of cryptographers to innovate and secure our future.