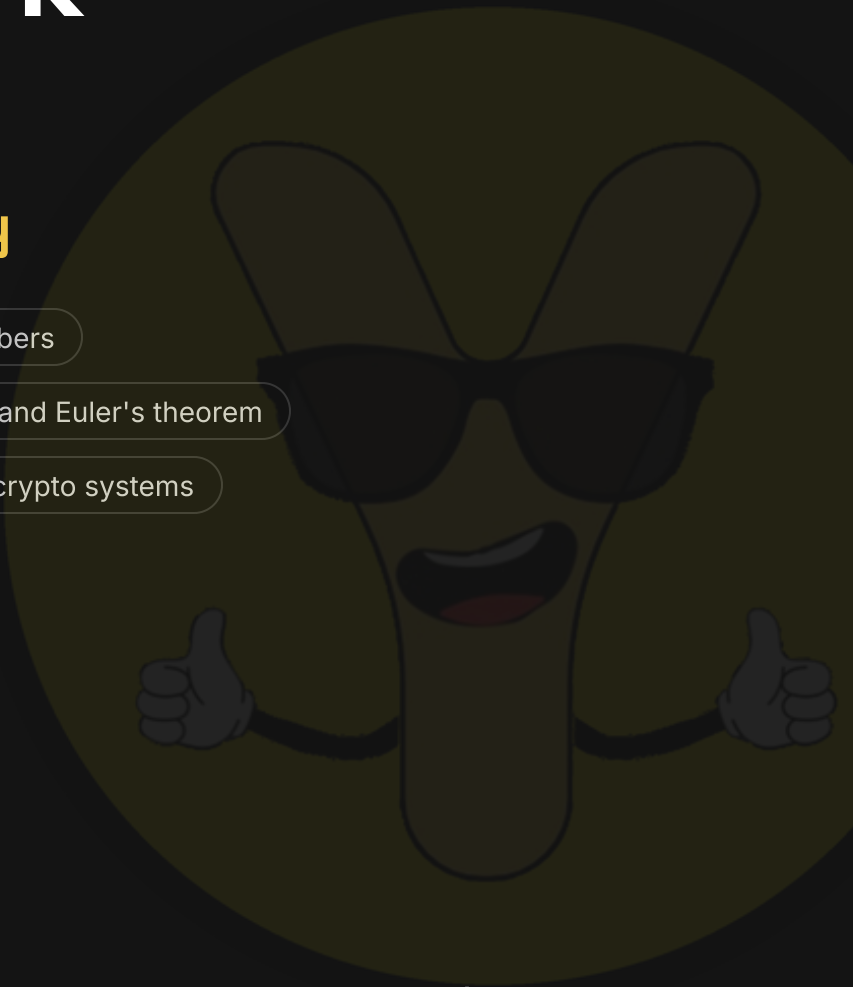
AKTU · 4TH YEAR · SEMESTER 7 · BCS072

Cryptography and Network Security

Unit 2 — Number Theory and Public Key Cryptography

- Introduction to group, field, finite field of the form GF(p)
- Modular arithmetic
- Prime and relative prime numbers
- Extended Euclidean Algorithm
- Advanced Encryption Standard (AES) encryption and decryption
- Fermat's and Euler's theorem
- Primarily testing
- Chinese Remainder theorem
- Discrete Logarithmic Problem
- Principals of public key crypto systems
- RSA algorithm
- Security of RSA



DECODED BY

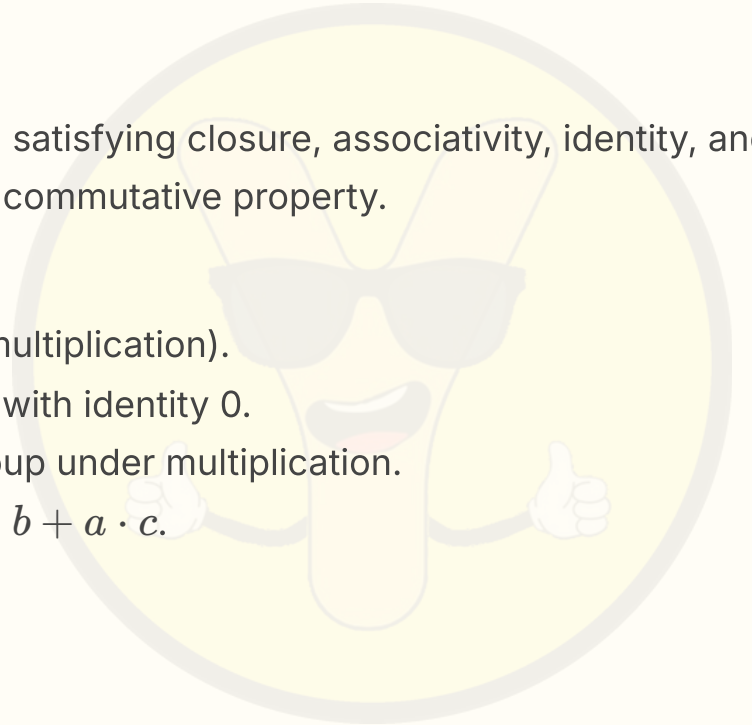
Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

Introduction to Group and Field

Fundamental algebraic structures form the mathematical foundation for modern public key cryptography.

- Group $(G, \cdot)$:
  - Set of elements with a binary operation satisfying closure, associativity, identity, and inverse.
  - Abelian group additionally satisfies the commutative property.
- Field $(F, +, \cdot)$:
  - Set with two operations (addition and multiplication).
  - Forms an abelian group under addition with identity 0.
  - Non-zero elements form an abelian group under multiplication.
  - Distributive law holds: $a \cdot (b + c) = a \cdot b + a \cdot c$.



Finite Field of the Form GF(p)

GALOIS FIELD GF(P)

A finite field containing exactly p elements, where p is a prime number, with arithmetic modulo p.

- Denoted as $\mathbb{Z}_p$ or $GF(p)$.
- All arithmetic operations (addition, subtraction, multiplication, division) result in an element within the set $\{0, 1, 2, \ldots, p-1\}$.
- Every non-zero element has a multiplicative inverse because p is prime.

Modular Arithmetic Essentials

Modular arithmetic restricts operations to a finite range of integers wrapping around a modulus.

- Congruence Modulo n:
  - $a \equiv b \pmod{n}$ means n divides $(a - b)$.
- Properties:
  - $(a + b) \pmod{n} = [(a \pmod{n}) + (b \pmod{n})] \pmod{n}$
  - $(a \cdot b) \pmod{n} = [(a \pmod{n}) \cdot (b \pmod{n})] \pmod{n}$
- Additive inverse of a is $(n - a) \pmod{n}$.
- Multiplicative inverse exists only if $\gcd(a, n) = 1$.

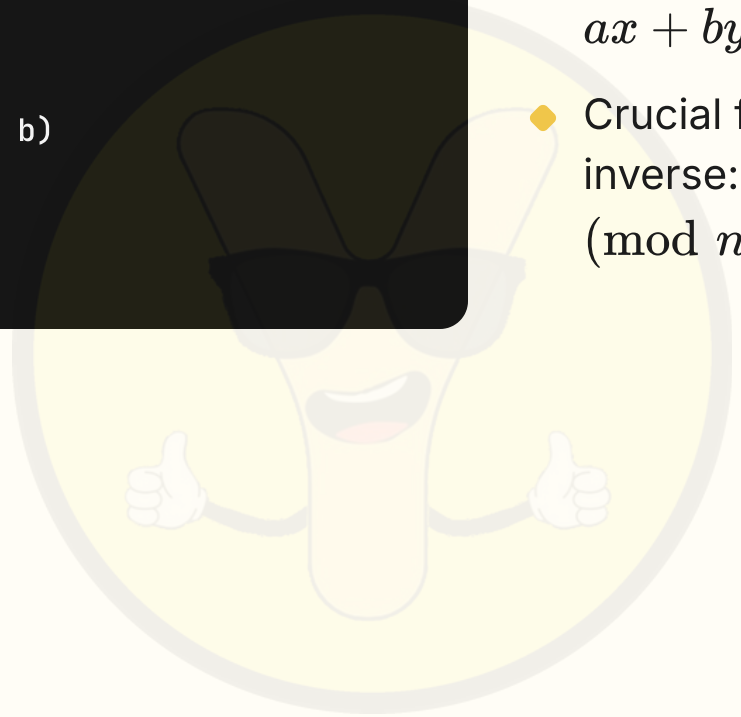
Prime and Relative Prime Numbers

Primes and coprime numbers are the building blocks of asymmetric key generation.

- Prime Number:
  - An integer $p > 1$ whose only positive divisors are 1 and p.
  - Examples: 2, 3, 5, 7, 11, 13.
- Relative Prime (Coprime):
  - Two integers a and b are relatively prime if their greatest common divisor is 1: $\gcd(a, b) = 1$.
  - Example: $\gcd(8, 15) = 1$, hence 8 and 15 are coprime.
- Fundamental Theorem of Arithmetic: Every integer greater than 1 is either a prime or a unique product of primes.

Extended Euclidean Algorithm

```
Function ExtendedGCD(a, b):
    if b == 0:
        return (a, 1, 0)
    (gcd, x1, y1) = ExtendedGCD(b, a % b)
    x = y1
    y = x1 - (a // b) * y1
    return (gcd, x, y)
```



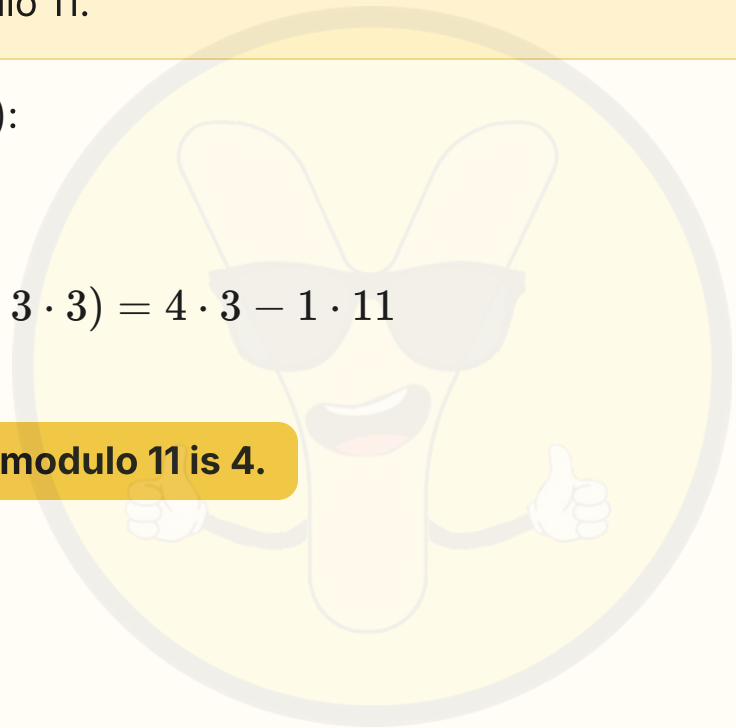
- Finds $\gcd(a, b)$ and integers x, y such that $ax + by = \gcd(a, b)$.
- Crucial for finding the modular multiplicative inverse: if $\gcd(a, n) = 1$, then $ax \equiv 1 \pmod{n}$, making x the inverse.

Extended Euclidean Algorithm Worked Example

Find the multiplicative inverse of 3 modulo 11.

1. Apply Euclidean algorithm for $\gcd(11, 3)$:
2. $11 = 3 \cdot 3 + 2$
3. $3 = 2 \cdot 1 + 1$
4. Express 1 backwards: $1 = 3 - 1 \cdot (11 - 3 \cdot 3) = 4 \cdot 3 - 1 \cdot 11$
5. Coefficients: $3(4) + 11(-1) = 1$

The modular multiplicative inverse of 3 modulo 11 is 4.



Fermat’s and Euler’s Theorems

These theorems simplify large modular exponentiation calculations in cryptography.

- Fermat's Little Theorem:
  - If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod{p}$.
  - Alternative form: $a^p \equiv a \pmod{p}$ for any integer a.
- Euler's Totient Function $\phi(n)$:
  - Counts positive integers up to n that are relatively prime to n.
  - For prime p, $\phi(p) = p - 1$. For $n = p \cdot q$, $\phi(n) = (p-1)(q-1)$.
- Euler's Theorem:
  - If $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \pmod{n}$.

Primality Testing

Determining whether a large number is prime is vital for generating keys.

- Deterministic Tests:
  - Trial division: Check divisibility up to $\sqrt{n}$. Too slow for large numbers.
  - AKS primality test: Runs in polynomial time, but impractical for crypto-scale keys.
- Probabilistic Tests:
  - Fermat primality test: Tests if $a^{n-1} \equiv 1 \pmod{n}$. Fails for Carmichael numbers.
  - Miller-Rabin test: Widely used probabilistic test based on square roots of unity modulo n.
  - Error probability drops exponentially with the number of rounds k.

Chinese Remainder Theorem (CRT)

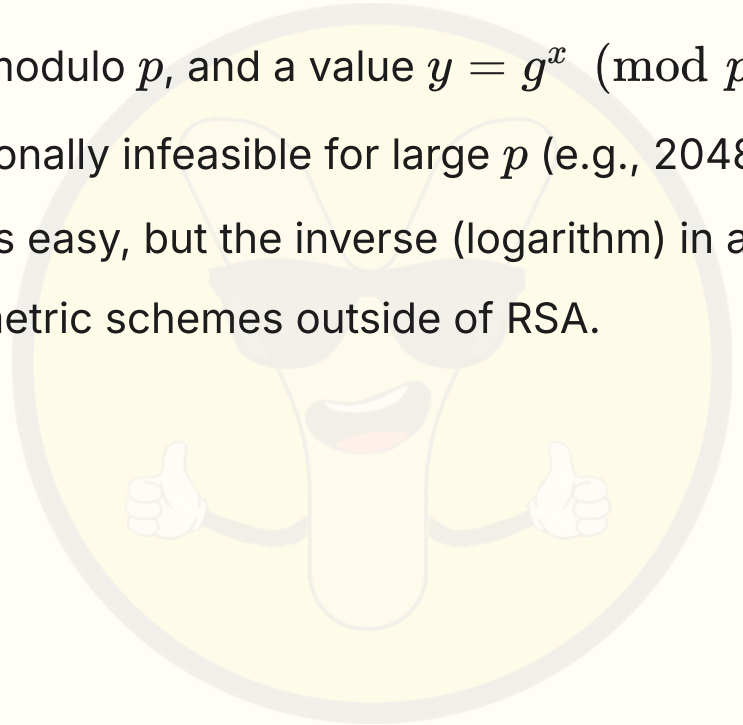
CRT is used to speed up modular exponentiation in RSA decryption.

- Helps solve a system of simultaneous congruences with moduli that are pairwise coprime.
- Let $n = p \cdot q$ where $\gcd(p, q) = 1$.
- Any integer $x \in \mathbb{Z}_n$ can be uniquely represented by the pair (x_p, x_q) where $x_p = x \pmod{p}$ and $x_q = x \pmod{q}$.
- Computations modulo p and q independently reduce computational overhead by a factor of 4 during RSA decryption.

Discrete Logarithm Problem (DLP)

DLP serves as the hard mathematical problem underlying Diffie-Hellman and DSA.

- Given a prime p, a primitive root g modulo p, and a value $y = g^x \pmod{p}$.
- Finding x given g, y, p is computationally infeasible for large p (e.g., 2048 bits).
- One-way function: Exponentiation is easy, but the inverse (logarithm) in a finite field is extremely hard.
- Forms the security basis for asymmetric schemes outside of RSA.

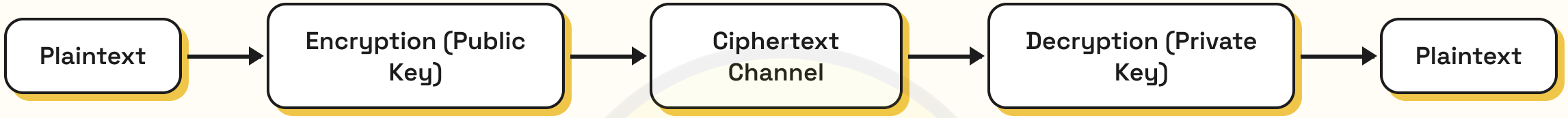


Principles of Public Key Cryptosystems

Proposed by Diffie and Hellman in 1976 to solve the key distribution problem.

- Uses two related keys: a public key for encryption and a private key for decryption.
- Derived from a mathematical trapdoor one-way function.
- Computationally easy to calculate $Y = f(X)$ with the public key.
- Computationally infeasible to calculate $X = f^{-1}(Y)$ without the private key.
- Eliminates the need for a secure prior channel for key exchange.

Public Key Cryptosystem Flow



Asymmetric encryption decouples the encryption key from the decryption key.

RSA Algorithm: Key Generation

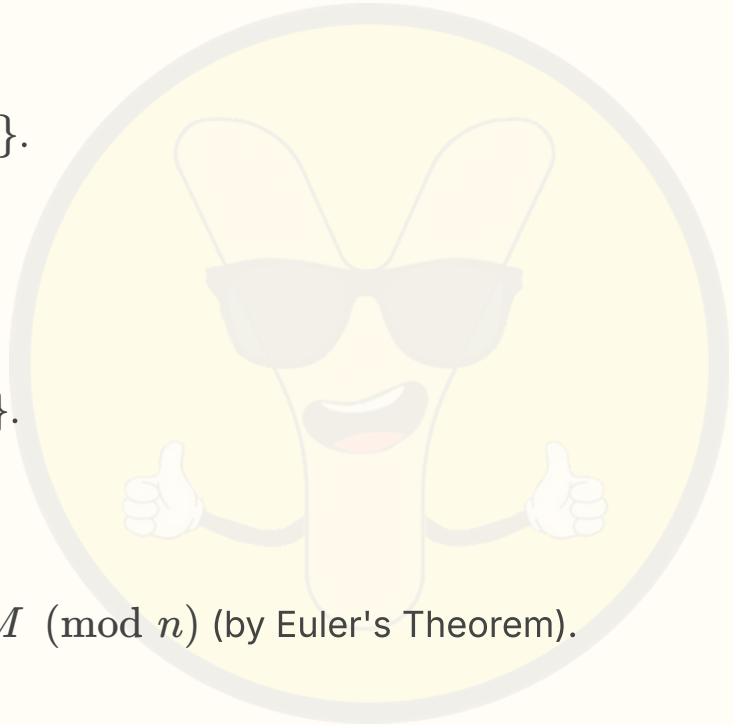
Rivest, Shamir, and Adleman developed the first practical public key cryptosystem.

- Step 1: Select two distinct large random primes p and q.
- Step 2: Compute modulus $n = p \cdot q$.
- Step 3: Compute Euler's totient $\phi(n) = (p-1)(q-1)$.
- Step 4: Choose integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.
- Step 5: Compute private exponent $d \equiv e^{-1} \pmod{\phi(n)}$.
- Public Key: $PU = \{e, n\}$. Private Key: $PR = \{d, p, q\}$.

RSA Algorithm: Encryption and Decryption

Mathematical operations using modular exponentiation.

- Encryption:
  - Sender obtains public key $PU = \{e, n\}$.
  - Plaintext $M < n$.
  - Ciphertext $C = M^e \pmod{n}$.
- Decryption:
  - Receiver uses private key $PR = \{d, n\}$.
  - Plaintext $M = C^d \pmod{n}$.
- Correctness Proof:
  - $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{1+k\phi(n)} \equiv M \pmod{n}$ (by Euler's Theorem).



RSA Worked Example

Let $p = 3$, $q = 11$, $e = 7$, $M = 5$. Perform encryption and decryption.

1. Modulus $n = 3 \times 11 = 33$
2. Totient $\phi(n) = (3 - 1)(11 - 1) = 20$
3. Find d: $7d \equiv 1 \pmod{20} \implies d = 3$ (since $7 \times 3 = 21 \equiv 1$)
4. Encryption: $C = 5^7 \pmod{33} = 78125 \pmod{33} = 14$
5. Decryption: $M = 14^3 \pmod{33} = 2744 \pmod{33} = 5$

Ciphertext is 14 and decrypted plaintext matches original message 5.

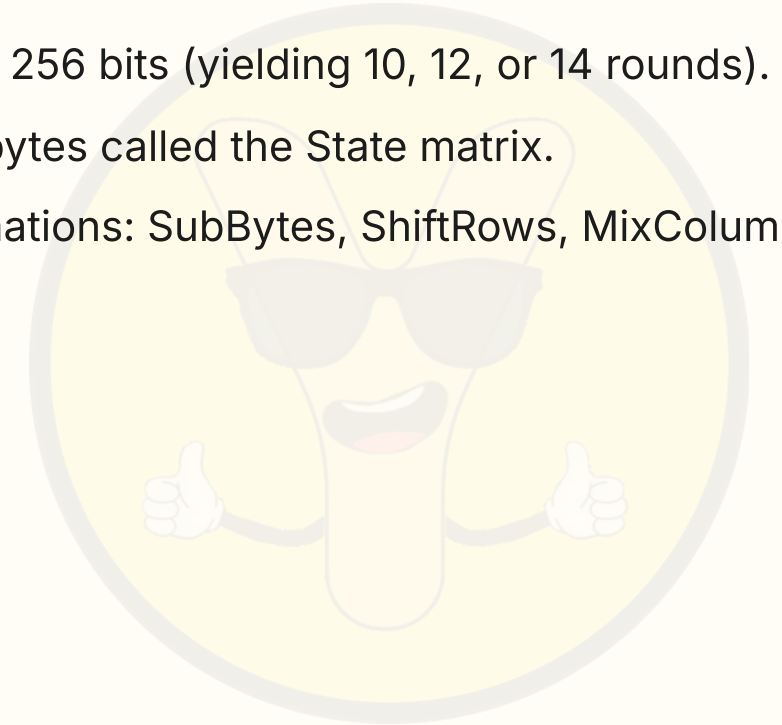
AES vs RSA Comparison

Feature	AES (Symmetric)	RSA (Asymmetric)
Key Type	Shared single secret key	Key pair (Public and Private)
Speed	Very fast in software and hardware	Slower, computationally intensive
Primary Use	Bulk data encryption	Key exchange and digital signatures
Mathematical Basis	Substitution-Permutation Network	Integer Factorization Problem

Advanced Encryption Standard (AES) Architecture

AES is a symmetric block cipher replacing DES, operating on 128-bit blocks.

- Key sizes supported: 128, 192, and 256 bits (yielding 10, 12, or 14 rounds).
- Data organized as a 4×4 array of bytes called the State matrix.
- Every round applies four transformations: SubBytes, ShiftRows, MixColumns, and AddRoundKey.
- Final round omits MixColumns.



AES Encryption Transformations

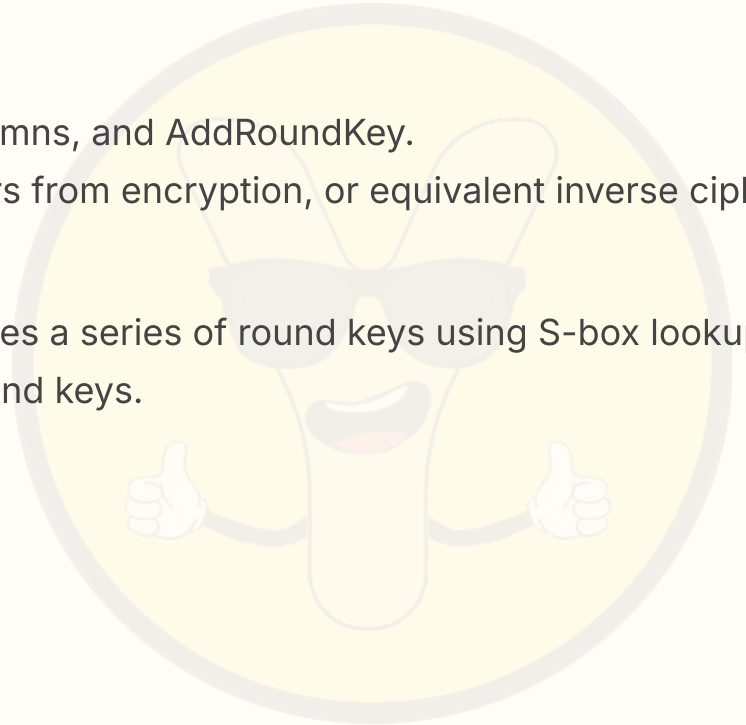
Each round mixes confusion and diffusion properties.

- SubBytes:
  - Non-linear substitution layer using an S-box for byte-by-byte substitution (confusion).
- ShiftRows:
  - Cyclic byte shift across rows of the state matrix (diffusion). Row 0 shifted by 0, Row 1 by 1, etc.
- MixColumns:
  - Matrix multiplication operating on columns, mixing bytes within each column.
- AddRoundKey:
  - Bitwise XOR of the current state with a 128-bit round key derived from the key schedule.

AES Decryption and Key Expansion

Decryption uses inverse transformations in reverse order.

- Inverse Transformations:
  - InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey.
  - Order of operations in decryption differs from encryption, or equivalent inverse cipher structure is used.
- Key Expansion Schedule:
  - Takes the initial cipher key and generates a series of round keys using S-box lookups, word rotations, and RCON constants.
  - Ensures avalanche effect across all round keys.



Security of RSA

The strength of RSA relies entirely on the difficulty of integer factorization.

- Integer Factorization Problem (IFP):
  - Given n, finding prime factors p and q is hard for large n (recommended size: 2048 to 4096 bits).
- Attacks on RSA:
  - Common modulus attack, small decryption exponent attack (Wiener's attack), and timing attacks.
  - Chosen Ciphertext Attack (CCA): Padding schemes like OAEP (Optimal Asymmetric Encryption Padding) protect against structural vulnerabilities.
- Quantum Threat: Shor's algorithm running on a quantum computer can factor n in polynomial time, rendering standard RSA insecure.

Exam corner

- **Q1** State and explain Fermat's Little Theorem and Euler's Theorem with suitable examples.
- **Q2** Explain the Extended Euclidean Algorithm. Find the modular multiplicative inverse of 5 modulo 23 using this algorithm.
- **Q3** Discuss the steps involved in RSA key generation, encryption, and decryption algorithms.
- **Q4** Given $p = 5$, $q = 11$, $e = 3$, and $M = 9$, calculate the RSA ciphertext and decrypted plaintext.
- **Q5** What is the Chinese Remainder Theorem? How does it optimize RSA decryption performance?
- **Q6** Compare and contrast AES and RSA encryption schemes across at least four distinct parameters.
- **Q7** Explain the round transformations in the Advanced Encryption Standard (AES) encryption process.
- **Q8** What is the Discrete Logarithm Problem? Why is it considered the foundation of modern asymmetric cryptography?

Unit 2 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 3 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

