MANIPAL ACADEMY
OF HIGHER EDUCATION

(Deemed-to-be-University under Section 3 of the UGC Act, 1956)

Information Security Lab

V Semester: B. Tech CCE (ICT3126)

Year: 2024

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGYMANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

CONTENT

Course Objectives

· Assess potential system vulnerabilities.
· Gain practical insight into various algorithms that can provide system, device, and network security.
· Use techniques and tools to fix security vulnerabilities

Course Outcomes

1. Examine security threats and vulnerabilities.
2. Demonstrate the various methods which can combat different security threats.
3. Model a security infrastructure appropriately to provide maximum security against any breach.

Evaluation Plan

· Split up of 60 marks for Regular Lab Evaluation

– Mid sem: 1 / 2 questions (Scenario based):20 Marks

– Record: 5 Marks: 3 Submissions

– Quiz: 15 Marks

– Project Midsem Evaluation : 10 Marks

· End Semester Lab evaluation: 40 marks (Duration 2 hrs)

– Program: 20 Marks

– Project: 20 Marks

Pre-Lab Session Instructions

1. Students should carry the Lab Manual Book and the required stationery to every labsession
2. Be on time and follow the institution's dress code
3. Must sign in the log register provided
4. Make sure to occupy the allotted seat and answer the attendance
5. Adhere to the rules and maintain the decorum

In-Lab Session Instructions

1. Follow the instructions on the allotted exercises
2. Show the program and results to the instructors on completion of experiments
3. Prescribed textbooks and class notes can be kept ready for reference if required
4. Avoid using LLMs

General Instructions for the Exercises in Lab

· Implement the given exercise individually and not in a group.

– Programs should be interactive with appropriate prompt messages, error messages if any, and descriptive messages for outputs.

– Comments should be used to state the problem.

– The statements within the program should be properly indented

Instructions to Students

- Plagiarism (copying from others) is strictly prohibited and would invite severe penalties in evaluation.

- In case a student misses a lab, he/ she must ensure that the experiment is completed before the next evaluation with the permission of the faculty concerned.

- Students missing out on the lab for genuine reasons like conferences, sports or activities assigned by the Department or Institute will have to take prior permission from the HOD to attend additional lab (with another batch) and complete it before the studentgoes on leave. The student could be awarded marks for the write-up for that day provided he submits it during the immediate next lab.

- Students who feel sick should get the HOD's permission to evaluate the lab records. However, attendance will not be given for that lab.

- Students will be evaluated only by the faculty with whom they are registered even though they carry out additional experiments in another batch.

- The presence of the student during the lab end semester exams is mandatory even ifthe student assumes he has scored enough to pass the examination

- Minimum attendance of 75

- If the student loses his book, he/she will have to rewrite all the lab details in the lab record.

- Questions for lab tests and examination are not necessarily limited to the questions in the manual, but may involve some variations and / or combinations of the questions.

  The students should NOT Bring mobile phones or any other electronic gadgets to the lab.

- The students should NOT Go out of the lab without permission.

Lab No. 1: Basic Symmetric Key Ciphers

Objectives

- To familiarize with Substitution Ciphers.
- To understand the working of transposition ciphers.

Symmetric key ciphers use the same key to encrypt and decrypt data. They are often used in combination with other algorithms into a symmetric encryption schemes.

Symmetric key cryptography schemes are categorized as stream ciphers and block ciphers. Stream ciphers work on a single bit (byte or computer word) at a time and execute some form of feedback structure so that the key is repeatedly changing.

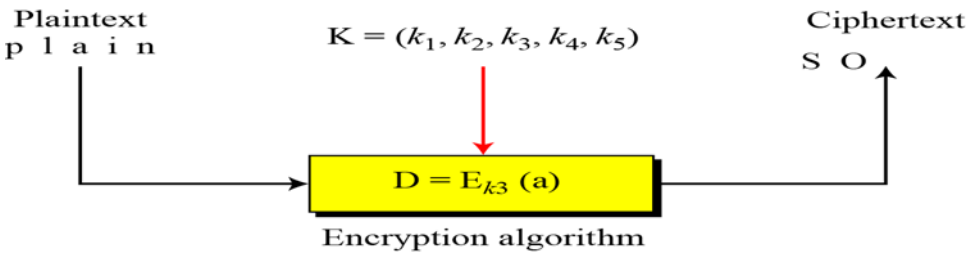


Figure 1: Stream Cipher

Block Cipher

In a block cipher, a group of plaintext symbols of size m ($m > 1$) are encrypted together creating a group of ciphertext of the same size. A single key is used to encrypt the whole block even if the key is made of multiple values as shown in Figure 2.

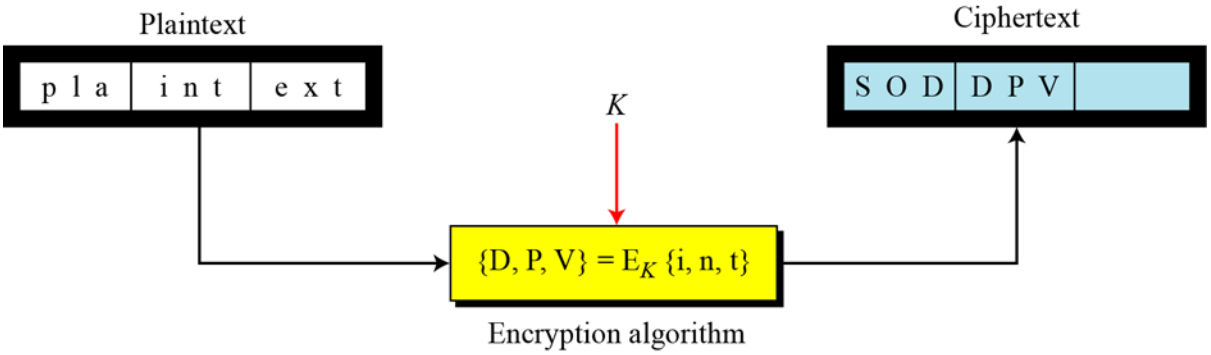


Figure 2: Block Cipher

A block cipher is so-called because the scheme encrypts one block of information at a time utilizing the same key on each block. In general, the same plaintext block will continually encrypt to the same ciphertext when using the similar key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

Block ciphers can operate in several modes such as Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB) and Output Feedback (OFB) mode.

Substitution Ciphers

A substitution cipher replaces one symbol with another. Substitution ciphers can be categorized as either monoalphabetic ciphers or polyalphabetic ciphers

In monoalphabetic substitution, the relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one. The simplest monoalphabetic cipher is the additive cipher, shown in Figure 3. This cipher is sometimes called a shift cipher and sometimes a Caesar cipher.

When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26}.

In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26}; the key is an integer in Z_{26}*.

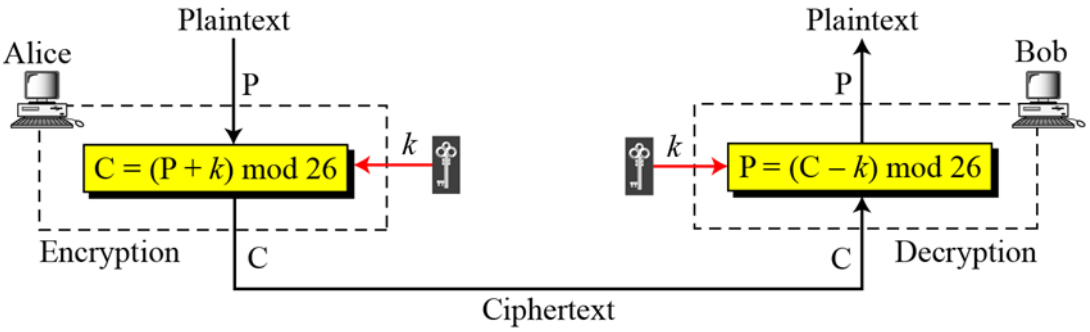


Figure 3. Additive Cipher

In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

Auto Key Cipher

$P = P_1P_2P_3 \ldots$ $C = C_1C_2C_3 \ldots$ $k = (k_1, P_1, P_2, \ldots)$

Encryption: $C_i = (P_i + k_i) \bmod 26$ Decryption: $P_i = (C_i - k_i) \bmod 26$

Transposition Cipher

A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.

- Keyless Transposition Ciphers
- Keyed Transposition Ciphers
- Combining Two Approaches

Lab Exercises

1) Encrypt the message "I am learning information security" using one of the following ciphers. Ignore the space between words. Decrypt the message to get the original plaintext:
   a) Additive cipher with key = 20
   b) Multiplicative cipher with key = 15
   c) Affine cipher with key = (15, 20)

2) Encrypt the message "the house is being sold tonight" using one of the following ciphers. Ignore the space between words. Decrypt the message to get the original plaintext:
   a) Vigenere cipher with key: "dollars"
   b) Autokey cipher with key = 7

3) Use the Playfair cipher to encipher the message "The key is hidden under the door pad". The secret key can be made by filling the first and part of the second row with the word "GUIDANCE" and filling the rest of the matrix with the rest of the alphabet.

4) Use a Hill cipher to encipher the message "We live in an insecure world". Use the following key:
   K = [03 03
   02 07]

5) John is reading a mystery book involving cryptography. In one part of the book, the author gives a ciphertext "CIW" and two paragraphs later the author tells the reader that this is a shift cipher and the plaintext is "yes". In the next chapter, the hero found a tablet in a cave with "XVIEWYWI" engraved on it. John immediately found the actual meaning of the ciphertext. Identify the type of attack and plaintext.

6) Use a brute-force attack to decipher the following message. Assume that you know it is an affine cipher and that the plaintext "ab" is enciphered to "GL":
   XPALASXYFGFUKPXUSOGEUTKCDGEXANMGNVS

Additional Exercises

1. Use a brute-force attack to decipher the following message enciphered by Alice using an additive cipher. Suppose that Alice always uses a key that is close to her birthday, which is on the 13th of the month:
NCJAEZRCLAS/LYODEPRLYZRCLASJLCPEHZDTOPDZOLN&BY

2. Eve secretly gets access to Alice's computer and using her cipher types "abcdefghi". The screen shows "CABDEHFGL". If Eve knows that Alice is using a keyed transposition cipher, answer the following questions:

a) What type of attack is Eve launching?
b) What is the size of the permutation key?
c) Use the Vigenere cipher with keyword "HEALTH" to encipher the message "Life is full of surprises".

Lab No. 2: Advanced Symmetric Key Ciphers

Objectives

- To understand the working of DES and AES algorithms.
- To implement AES and DES algorithm.
- Analyze the performance of AES and DES algorithms

DES:

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST)

The encryption process is made of two permutations (P-boxes, initial and final permutations, and sixteen Feistel rounds as shown in Figure 4

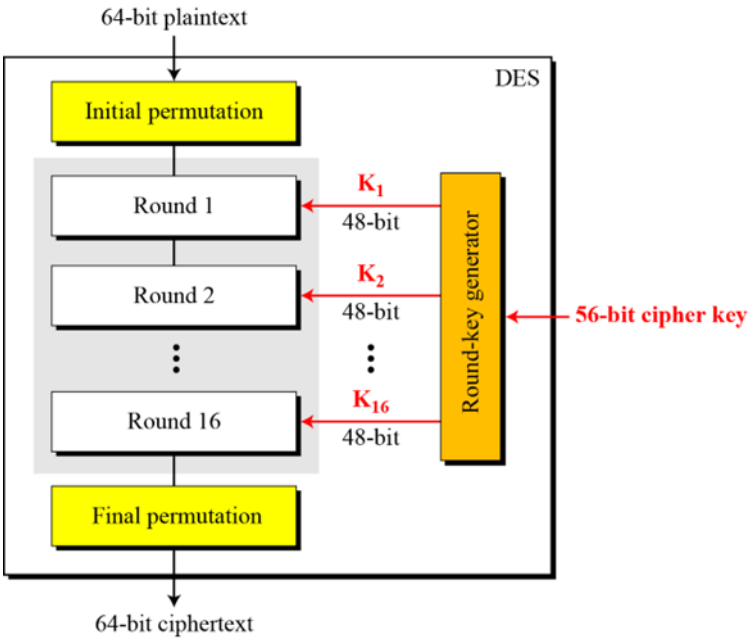


Figure 4: General structure of DES

Initial Permutation (IP): The 64-bit plaintext block is permuted according to a fixed table, shuffling the bits to create a new order.

Key Schedule Generation: The 56-bit key is divided into two 28-bit halves. Each half is then rotated and permuted according to a predefined schedule to produce sixteen 48-bit round keys, one for each round of encryption.

16 Rounds of Encryption: The 64-bit block is split into two 32-bit halves, called Left (L) and Right (R).

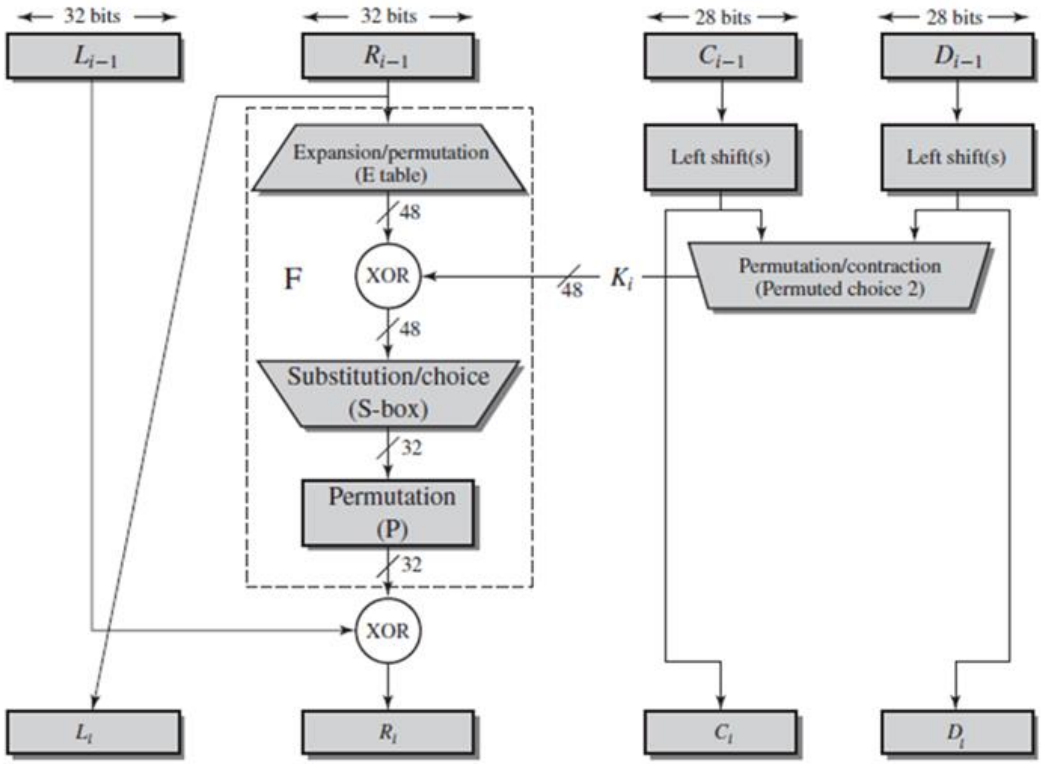


Figure 5: Single Round of DES Algorithm

Single round of DES is shown in Figure 5. For each of the 16 rounds, a new right half is generated by expanding the previous right half to 48 bits using the Expansion (E) function.

The expanded right half is XORed with the round key. That result is passed through a series of substitution boxes (S-boxes), which reduce the 48-bit output back to 32 bits. The S-box output is then permuted using the Permutation (P) function. The new right half is XORed with the previous left half. The previous right half becomes the new left half.

- After 16 rounds, the left and right halves are recombined and permuted using the final permutation (FP).

Final Permutation (FP): The combined left and right halves are permuted according to a fixed table to produce the final 64-bit ciphertext block.

The Advanced Encryption Standard (AES):

AES is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001. AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.

Main components of AES are as follows as shown in Figure 6.

Initial Round: **AddRoundKey**: Each byte of the state is combined with a round key using the XOR operation.

Main Rounds (Repeated for 9, 11, or 13 times depending on key size):

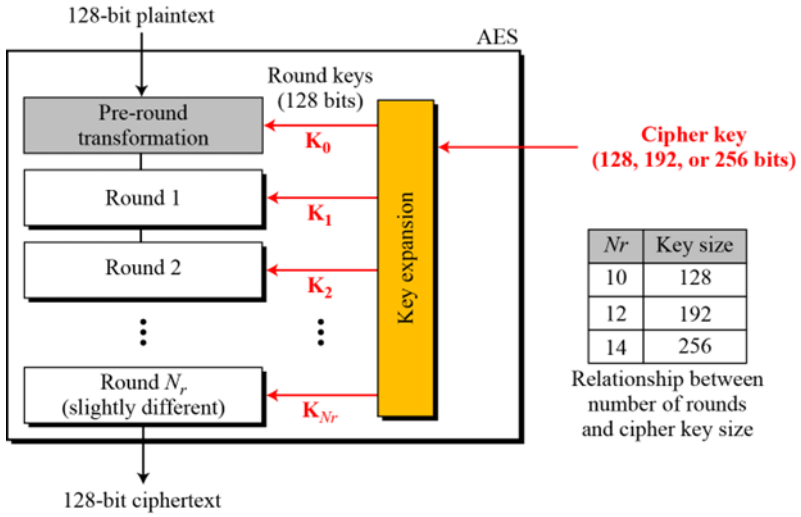
SubBytes: A non-linear substitution step where each byte is replaced with another byte using an S-box (substitution box).

ShiftRows: A transposition step where each row of the state is shifted cyclically by a certain number of bytes.

MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.

AddRoundKey: Each byte of the state is combined with a round key using the XOR operation.

Final Round: It has SubBytes, ShiftRows and AddRoundKey steps



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Figure 6: General design of AES encryption cipher

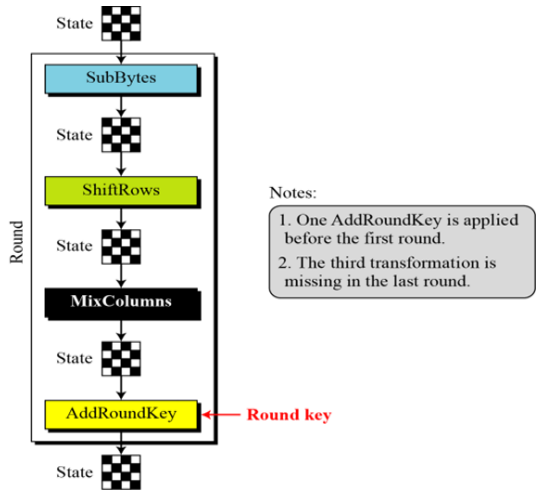


Figure 7: Structure of each round at the encryption site

To provide security, AES uses four types of transformations: substitution, permutation, mixing, and key-adding.

Detailed description about each type of transformation is as follows:

Substitution

The first transformation, SubBytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

Transformation Using the GF(2^8) Field

AES also defines the transformation algebraically using the GF(28) field with the irreducible polynomials ($x^8 + x^4 + x^3 + x + 1$)

The SubBytes and InvSubBytes transformations are inverses of each other.

Another transformation found in a round is shifting, which permutes the bytes as shown in Figure 8. In the encryption, the transformation is called ShiftRows.

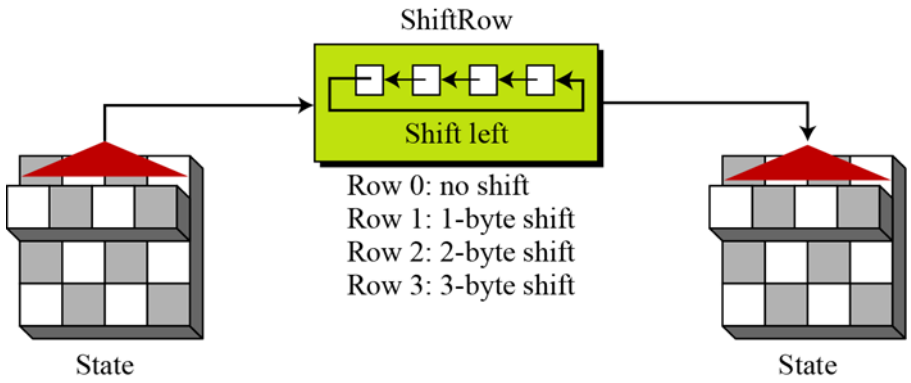


Figure 8: ShiftRows transformation

InvShiftRows

In the decryption, the transformation is called InvShiftRows and the shifting is to the right.

MixColumns

The MixColumns transformation operates at the column level; it transforms each column of the state to a new column. The MixColumns and InvMixColumns transformations are inverses of each other.

Key Adding

AddRoundKey proceeds one column at a time. AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.

Exercises:

1. Encrypt the message "Confidential Data" using DES with the following key: "A1B2C3D4". Then decrypt the ciphertext to verify the original message.

2. Encrypt the message "Sensitive Information" using AES-128 with the following key: "0123456789ABCDEF0123456789ABCDEF". Then decrypt the ciphertext to verify the original message.

3. Compare the encryption and decryption times for DES and AES-256 for the message "Performance Testing of Encryption Algorithms". Use a standard implementation and report your findings.

4. Encrypt the message "Classified Text" using Triple DES with the key "1234567890ABCDEF1234567890ABCDEF1234567890ABCDEF". Then decrypt the ciphertext to verify the original message.

5. Encrypt the message "Top Secret Data" using AES-192 with the key "FEDCBA9876543210FEDCBA9876543210". Show all the steps involved in the encryption process (key expansion, initial round, main rounds, final round).

Additional Exercises:

1. Using DES and AES(128, 192, and 256 bits key).encrypt the five different messages using same key.

   a. Consider different modes of operation
   b. Plot the graph which shows execution time taken by each technique.
   c. Compare time taken by different modes of operation

2. Encrypt the following block of data using DES with the key "A1B2C3D4E5F60708". The data to be encrypted is: Mathematica

Block1:
5468697320697320612063 6f6e666964656e7469616c206d657373616765

Block2:
416e64207468697320697320746865207365636f6e6420626c6f636b

   a. Provide the ciphertext for each block.
   b. Decrypt the ciphertext to retrieve the original plaintext blocks.

3. Using AES-256, encrypt the message "Encryption Strength" with the key "0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF". Then decrypt the ciphertext to verify the original message.
Encrypt the message "Secure Communication" using DES in Cipher Block Chaining (CBC) mode with the key "A1B2C3D4" and an initialization vector (IV) of "12345678". Provide the ciphertext and then decrypt it to retrieve the original message.

4. Encrypt the message "Cryptography Lab Exercise" using AES in Counter (CTR) mode with the key "0123456789ABCDEF0123456789ABCDEF" and a nonce of "0000000000000000". Provide the ciphertext and then decrypt it to retrieve the original message.

Lab No. 3: Asymmetric Key Ciphers

Objectives

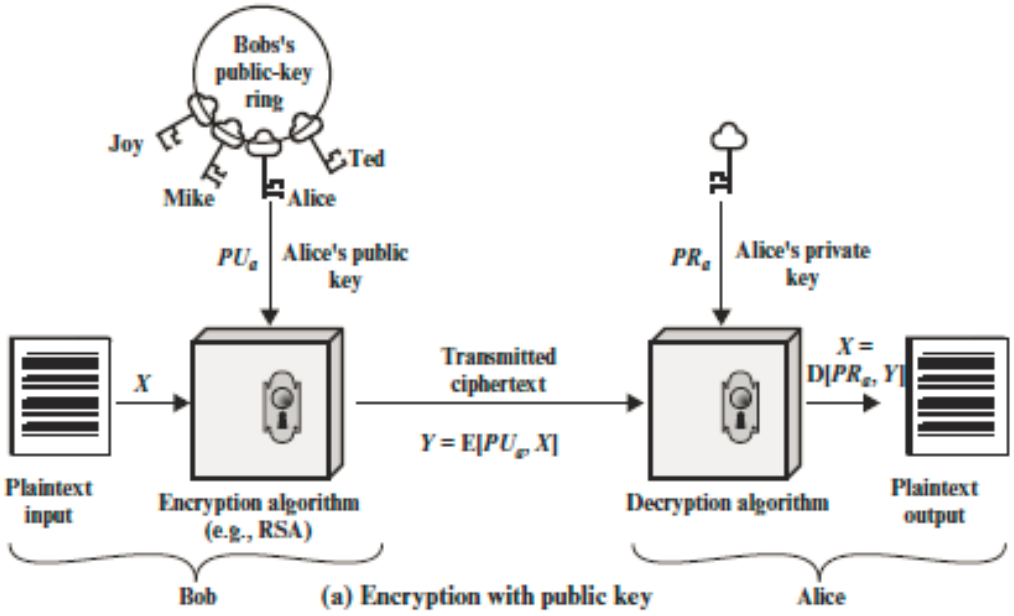
- To demonstrate the ability to generate, use, and understand the public and private keys in various asymmetric encryption schemes.

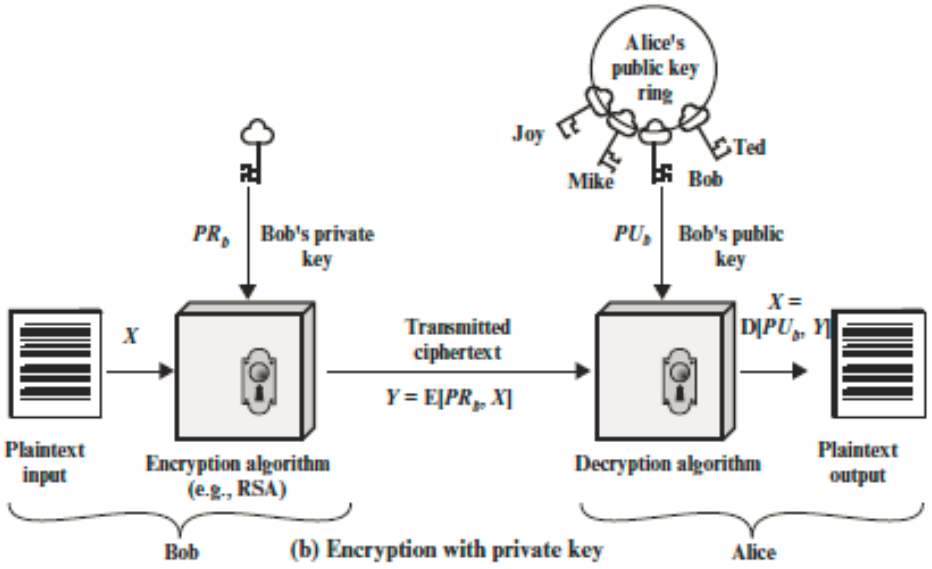
- To understand the performance implications of different asymmetric encryption algorithms

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristics:

- It is computationally infeasible to determine the decryption key given only knowledge of cryptographic algorithms and encryption keys.

- Either of the two related keys can be used for encryption, with the other used for decryption





THE RSA ALGORITHM

The RSA scheme is a cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n. A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 21024. We examine RSA in this section in some detail, beginning with an explanation of the algorithm. Then we examine some of the computational and cryptanalytical implications of

RSA. RSA uses the mathematical properties of prime numbers and modular arithmetic.

- Public keys are used for encryption, and private keys are used for decryption.

- The security of RSA relies on the difficulty of factoring large composite numbers into their prime factors.

RSA is widely used for secure data transmission, digital signatures, and key exchange mechanisms due to its robustness and security features

Key Generation

- Generate Two Large Prime Numbers: Choose two distinct large prime numbers, p and q. These primes should be large enough to ensure the security of the RSA algorithm.

- Compute the Modulus:

Calculate $n=p\times q$; n is used as the modulus for both the public and private keys.

- Compute Euler's Totient Function: Calculate $\phi(n)=(p-1)\times(q-1)$.

$\phi(n)$ represents the number of integers less than n that are relatively prime to n.

- Choose the Public Exponent: Select an integer ee such that $1<e<\phi(n)$ and $\gcd(e,\phi(n))=1$.

The public exponent e is typically chosen as a small prime number like 3 or 65537 for efficiency.

- Compute the Private Exponent: Calculate the private exponent d such that $d\times e\equiv 1 (\mod \phi(n))$.

d is the modular multiplicative inverse of ee modulo $\phi(n)$.

- Public and Private Keys: The public key consists of (n,e). The private key consists of (n,d).

Encryption

- Convert the Message: Convert the plaintext message M into an integer m such that $0 \leq m < n$. This can be done using a suitable padding scheme to ensure the message is within the valid range.

- Encrypt the Message: Compute the ciphertext cc using the public key: $c = m^e \mod n$

Decryption

- Decrypt the Ciphertext: Compute the plaintext message m using the private key: $m = c^d \mod n$

- Convert the Integer Back to Message: Convert the integer m back to the original plaintext message M.

ELGAMAL Encryption Algorithm

The ElGamal encryption algorithm is an asymmetric key encryption algorithm based on the Diffie-Hellman key exchange. It was designed by Taher ElGamal in 1985 and provides both encryption and digital signature functionalities.

Key Components

Public Parameters:

- A large prime number p.
- A generator g of the multiplicative group of integers modulo p.

Private Key:

- A random integer x such that $1 \leq x \leq p-2$.

Public Key:

- Compute $y = g^x \bmod p$.
- The public key is the tuple (p,g,y).

Key Generation

- Choose a large prime number p.
- Choose a generator g for the multiplicative group of integers modulo p.
- Select a private key x where $1 \leq x \leq p-2$
- Compute the public key component y as $y = g^x \bmod p$.

Public key: (p,g,y)

Private key: x

Encryption

- Convert the plaintext message M into an integer m such that $0 \leq m < p$.
- Choose a random integer k such that $1 \leq k \leq p-2$.
- Compute the ciphertext components:
  - $c_1 = g^k \bmod p$
  - $c_2 = m \cdot y^k \bmod p$

Ciphertext: (c_1, c_2)

Decryption

Use the private key *xxx* to compute:

- $s = c_1^x \bmod p$

Compute the plaintext message m as:

- $m = c_2 \cdot s^{-1} \bmod p$
- Here, s^{-1} is the modular inverse of *s* modulo p.

Plaintext: M

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is an asymmetric encryption technique based on the algebraic structure of elliptic curves over finite fields. It offers significant advantages over traditional RSA encryption, such as smaller key sizes for

equivalent security levels and faster computational speeds.

Key Components

Elliptic Curve Definition:

An elliptic curve over a finite field Fp is defined by an equation of the form $y^2=x^3+ax+b \bmod p$, where a and b are constants defining the curve's shape, and p is a prime number.

Base Point G (Generator):

- A specific point on the curve used as the base for generating keys and performing operations.

Private Key d:

- A random integer d chosen as the private key.

Public Key Q:

- The public key is computed as $Q=d \cdot G$, where .(dot) denotes elliptic curve point multiplication.

Key Generation

Curve Parameters:

- Choose a suitable elliptic curve (e.g., secp256k1) defined over a finite field Fp .
- Select curve parameters a, b, and a prime modulus p.

Base Point G Selection:

- Choose a base point G on the elliptic curve.

Private Key Generation:

- Generate a random integer *d* such that $1 \leq d <$ order of G.

Public Key Computation:

- Compute the public key Q as $Q = d \cdot G$.

Public key: Q

Private key: *d*

Encryption and Decryption

ECC is primarily used for key exchange rather than direct encryption of messages. The Diffie-Hellman key exchange and elliptic curve DSA (ECDSA) are commonly used protocols based on ECC for secure communication and digital signatures

Lab Exercises:

1. Using RSA, encrypt the message "Asymmetric Encryption" with the public key (n, e). Then decrypt the ciphertext with the private key (n, d) to verify the original message.

2. Using ECC (Elliptic Curve Cryptography), encrypt the message "Secure Transactions" with the public key. Then decrypt the ciphertext with the private key to verify the original message.

3. Given an ElGamal encryption scheme with a public key (p, g, h) and a private key x, encrypt the message "Confidential Data". Then decrypt the ciphertext to retrieve the original message.

4. Design and implement a secure file transfer system using RSA (2048-bit) and ECC (secp256r1 curve) public key algorithms. Generate and exchange keys, then encrypt and decrypt files of varying sizes (e.g., 1 MB, 10 MB) using both algorithms. Measure and compare the performance in terms of key generation time, encryption/decryption speed, and computational overhead. Evaluate the security and efficiency of each algorithm in the context of file transfer, considering factors such as key size, storage requirements, and resistance to known attacks. Document your findings, including performance metrics and a summary of the strengths and weaknesses of RSA and ECC for secure file transfer.

5. As part of a project to enhance the security of communication in a peer-to-peer file sharing system, you are tasked with implementing a secure key exchange mechanism using the Diffie-Hellman algorithm. Each peer must establish a shared secret key with another peer over an insecure channel. Implement the Diffie-Hellman key exchange protocol, enabling peers to generate their public and private keys and securely compute the shared secret key. Measure the time taken for key generation and key exchange processes.

Additional Exercises:

1. With the ElGamal public key (p = 7919, g = 2, h = 6465) and the private key x = 2999, encrypt the message "Asymmetric Algorithms". Decrypt the resulting ciphertext to verify the original message.

2. Using ECC (Elliptic Curve Cryptography), encrypt the message "Secure Transactions" with the public key. Then decrypt the ciphertext with the private key to verify the original message.

3. Encrypt the message "Cryptographic Protocols" using the RSA public key (n, e) where n = 323 and e = 5. Decrypt the ciphertext with the private key (n, d) where

d = 173 to confirm the original message

4. You are tasked with implementing a secure communication system for a healthcare organization to exchange sensitive patient information securely between doctors and hospitals. Implement the ElGamal encryption scheme to encrypt patient records and medical data, ensuring confidentiality during transmission. Generate public and private keys using the secp256r1 curve and use ElGamal encryption to encrypt patient data with the recipient's public key and decrypt it with the recipient's private key. Measure the performance of encryption and decryption processes for data of varying sizes.

5. You are conducting a study to evaluate the performance and security of RSA and ElGamal encryption algorithms in securing communication for a government agency. Implement both RSA (using 2048-bit keys) and ElGamal (using the secp256r1 curve) encryption schemes to encrypt and decrypt sensitive messages exchanged between agencies. Measure the time taken for key generation, encryption, and decryption processes for messages of various sizes (e.g., 1 KB, 10 KB). Compare the computational efficiency and overhead of RSA and ElGamal algorithms. Perform the same for ECC with RSA and ElGamal.

Lab No. 4: Advanced Asymmetric Key Ciphers

Objectives :

- Implement and compare the performance of multiple asymmetric encryption algorithms (e.g., RSA, ElGamal, Rabin) in a controlled environment, measuring factors such as encryption/decryption speed and key generation time.

- Design and develop a modular key management system capable of handling various cryptographic protocols, with emphasis on scalability, security, and ease of integration.

- Create a flexible framework for testing different access control mechanisms in cryptographic systems, allowing for easy implementation and evaluation of various policies and revocation strategies

Asymmetric Encryption Algorithms: Asymmetric encryption, also known as public-key cryptography, uses a pair of mathematically related keys: a public key for encryption and a private key for decryption. This approach allows secure communication without the need to share secret keys. Common asymmetric algorithms include RSA, ElGamal, and Rabin. Each has its own mathematical foundations and security properties, making comparative analysis valuable for understanding their strengths and weaknesses in different scenarios.

Key Management Systems: Key management is a critical aspect of cryptographic systems, encompassing the generation, exchange, storage, use, and replacement of cryptographic keys. A robust key management system ensures the security and integrity of encrypted communications. It must handle tasks such as key generation, distribution, storage, rotation, and revocation. The challenges in key management increase with the scale and complexity of

the system, especially in distributed environments.

Access Control in Cryptographic Systems: Access control in cryptography involves mechanisms to restrict access to encrypted data or cryptographic operations. This includes managing who can encrypt or decrypt data, as well as controlling access to keys. Implementing flexible access control policies is crucial for maintaining security in various scenarios, from simple user authentication to complex, attribute-based access control systems. The ability to revoke access and update policies dynamically is also an important consideration in modern cryptographic systems.

Mathematical Foundations

RSA Algorithm:

1. Key Generation:

   - Choose two large prime numbers p and q

- Compute n = p * q

- Compute φ(n) = (p-1) * (q-1)

- Choose e such that 1 < e < φ(n) and gcd(e, φ(n)) = 1

- Compute d such that d * e ≡ 1 (mod φ(n))

- Public key: (n, e), Private key: (n, d)

2.Encryption:

For plaintext m, ciphertext c = m^e mod n

3. Decryption:

m = c^d mod n

ElGamal Algorithm:

1. Key Generation:

- Choose a large prime p and a generator g of the multiplicative group of integers modulo p

- Choose a random integer x, 1 < x < p-1

- Compute y = g^x mod p

- Public key: (p, g, y), Private key: x

2. Encryption:

- Choose a random k, 1 < k < p-1

- Compute c1 = g^k mod p

- Compute s = y^k mod p

- For plaintext m, compute c2 = m * s mod p

- Ciphertext: (c1, c2)

3. Decryption:

- Compute s = c1^x mod p

- Compute m = c2 * s^(-1) mod p

Rabin Algorithm:

1. Key Generation:

- Choose two large primes p and q, where p ≡ q ≡ 3 (mod 4)

- Compute n = p * q

- Public key: n, Private key: (p, q)

2. Encryption:

For plaintext m, ciphertext c = m^2 mod n

3.Decryption:

- Compute mp = c^((p+1)/4) mod p

- Compute mq = c^((q+1)/4) mod q

- Use Chinese Remainder Theorem to find four square roots:

r1 = (yp * p * mq + yq * q * mp) mod n

r2 = n - r1

r3 = (yp * p * mq - yq * q * mp) mod n

r4 = n - r3

where yp * p + yq * q = 1

- One of these roots is the original message m

Each of these algorithms has its own unique mathematical properties that contribute to its security and performance characteristics. RSA relies on the difficulty of factoring large numbers, ElGamal is based on the discrete logarithm problem, and Rabin's security is tied to the difficulty of finding square roots modulo a composite number.

Key Management Systems:

1. Key Entropy:

Entropy H of a key K with n possible values, each with probability p(i):

H(K) = -∑(i=1 to n) p(i) * log2(p(i))

2. Key Derivation Function (KDF):

DK = KDF(Key, Salt, Iterations)

Where DK is the derived key, Key is the original key or password, Salt is a random value,

and Iterations is the number of times the function is applied.

3. Key Rotation:

For a system with N keys and a rotation period of T:

Rotation Rate = N / T

4. Key Expiry:

If a key K is created at time t0 with lifetime L:

Expiry Time = t0 + L

5. Diffie-Hellman Key Exchange:

Public parameters: prime p, generator g

Alice computes: A = g^a mod p

Bob computes: B = g^b mod p

Shared secret: K = A^b mod p = B^a mod p = g^(ab) mod p

Access Control in Cryptographic Systems:

1. Role-Based Access Control (RBAC):

Access(User, Object) = ∃Role : HasRole(User, Role) ∧ CanAccess(Role, Object)

2. Attribute-Based Access Control (ABAC):

Access(User, Object, Environment) = f(UserAttributes, ObjectAttributes, EnvironmentAttributes)

Where f is a policy function evaluating to true or false.

3. Bell-LaPadula Model:

- Simple Security Property: S(Subject) ≥ C(Object) for read access

- *-Property: S(Subject) ≤ C(Object) for write access

Where S is the security level of the subject and C is the classification of the object.

4. Mandatory Access Control (MAC):

AccessGranted = (SubjectClearance ≥ ObjectClassification) ∧ PolicyRulesatisfied

5. Discretionary Access Control (DAC):

AccessMatrix[Subject, Object] = {Rights}

Where Rights could be Read, Write, Execute, etc.

6. Time-based Access Control:

$$Access(User, Object, Time) = (Time \geq StartTime) \land (Time \leq EndTime) \land OtherConditions$$

7. Probabilistic Access Control:

$$P(Access\ Granted \mid Conditions) = f(UserTrustLevel, ObjectSensitivity, EnvironmentRisk)$$

Where P is probability and f is a function mapping conditions to a probability.

These equations and mathematical concepts form the basis for implementing and analyzing key management and access control systems. They allow for quantitative assessment of security properties, guide the design of secure systems, and provide a framework for evaluating the effectiveness of different approaches in cryptographic access control.

Lab exercises

Question 1

SecureCorp is a large enterprise with multiple subsidiaries and business units located across different geographical regions. As part of their digital transformation initiative, the IT team at SecureCorp has been tasked with building a secure and scalable communication system to enable seamless collaboration and information sharing between their various subsystems.

The enterprise system consists of the following key subsystems:

1. Finance System (System A): Responsible for all financial record-keeping, accounting, and

reporting.

2. HR System (System B): Manages employee data, payroll, and personnel-related processes.

3. Supply Chain Management (System C): Coordinates the flow of goods, services, and information across the organization's supply chain.

These subsystems need to communicate securely and exchange critical documents, such as financial reports, employee contracts, and procurement orders, to ensure the enterprise's overall efficiency.

The IT team at SecureCorp has identified the following requirements for the secure communication and document signing solution:

1. Secure Communication: The subsystems must be able to establish secure communication channels using a combination of RSA encryption and Diffie-Hellman key exchange.

2. Key Management: SecureCorp requires a robust key management system to generate, distribute, and revoke keys as needed to maintain the security of the enterprise system.

3. Scalability: The solution must be designed to accommodate the addition of new subsystems in the future as SecureCorp continues to grow and expand its operations.

Implement a Python program which incorporates the requirements.

Question 2:

HealthCare Inc., a leading healthcare provider, has implemented a secure patient data management system using the Rabin cryptosystem. The system allows authorized healthcare

professionals to securely access and manage patient records across multiple hospitals and clinics within the organization. Implement a Python-based centralized key management service that can:

- Key Generation: Generate public and private key pairs for each hospital and clinic using the Rabin cryptosystem. The key size should be configurable (e.g., 1024 bits).

- Key Distribution: Provide a secure API for hospitals and clinics to request and receive their public and private key pairs.

- Key Revocation: Implement a process to revoke and update the keys of a hospital or clinic when necessary (e.g., when a facility is closed or compromised).

- Key Renewal: Automatically renew the keys of all hospitals and clinics at regular intervals (e.g., every 12 months) to maintain the security of the patient data management system.

- Secure Storage: Securely store the private keys of all hospitals and clinics, ensuring that they are not accessible to unauthorized parties.

- Auditing and Logging: Maintain detailed logs of all key management operations, such as key generation, distribution, revocation, and renewal, to enable auditing and compliance reporting.

- Regulatory Compliance: Ensure that the key management service and its operations are compliant with relevant data privacy regulations (e.g., HIPAA).

- Perform a trade-off analysis to compare the workings of Rabin and RSA.

Additional Questions

Question 1

DigiRights Inc. is a leading provider of digital content, including e-books, movies, and music. The company has implemented a secure digital rights management (DRM) system using the ElGamal cryptosystem to protect its valuable digital assets. Implement a Python-based centralized key management and access control service that can:

- Key Generation: Generate a master public-private key pair using the ElGamal cryptosystem. The key size should be configurable (e.g., 2048 bits).

- Content Encryption: Provide an API for content creators to upload their digital content and have it encrypted using the master public key.

- Key Distribution: Manage the distribution of the master private key to authorized customers, allowing them to decrypt the content.

- Access Control: Implement flexible access control mechanisms, such as:

    - Granting limited-time access to customers for specific content

    - Revoking access to customers for specific content

    - Allowing content creators to manage access to their own content

- Key Revocation: Implement a process to revoke the master private key in case of a security breach or other emergency.

- Key Renewal: Automatically renew the master public-private key pair at regular intervals (e.g., every 24 months) to maintain the security of the DRM system.

- Secure Storage: Securely store the master private key, ensuring that it is not accessible to unauthorized parties.

- Auditing and Logging: Maintain detailed logs of all key management and access control operations to enable auditing and troubleshooting.

Question 2

Suppose that XYZ Logistics has decided to use the RSA cryptosystem to secure their sensitive communications. However, the security team at XYZ Logistics has discovered that one of their employees, Eve, has obtained a partial copy of the RSA private key and is attempting to recover the full private key to decrypt the company's communications.

Eve's attack involves exploiting a vulnerability in the RSA key generation process, where the prime factors (p and q) used to generate the modulus (n) are not sufficiently large or random.

Develop a Python script that can demonstrate the attack on the vulnerable RSA cryptosystem

and discuss the steps to mitigate the attack.