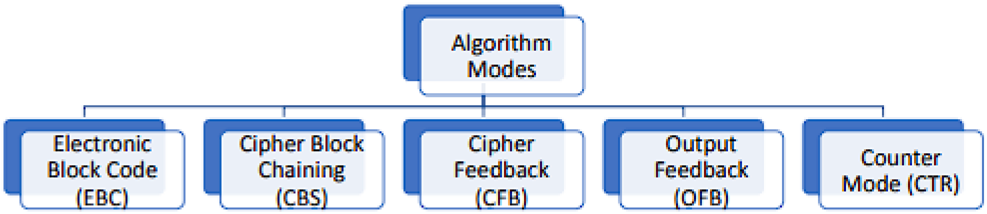
📘Unit 3 Cryptography (Milan)



Electronic Block Code (EBC)

- **Message Handling**:
  - Message is broken into independent blocks, each of which is encrypted separately.
  - Each block functions as a value, substituted much like a codebook.
- **Characteristics**:
  - Each block is encoded independently from the others.
- **Uses**:
  - Secure transmission of single values.

Benefits:

- Parallel encryption of blocks of bits is possible, making it a faster encryption method.
- Simple to implement as a block cipher.

Problems:

- Repetitions in the message may appear in the ciphertext.
- Weakness stems from the independence of encrypted message blocks.
- Primarily useful for sending small amounts of data.

Cipher Block Chaining (CBC)

- **Message Handling**:
  - The message is broken into blocks and linked together during encryption.
  - Each ciphertext block is "chained" with the next plaintext block using the previous ciphertext block.
- **Process**:
  - Uses an Initial Vector (IV) to start the encryption process.
- **Uses**:
  - Bulk data encryption, authentication.

Benefits:

- Provides stronger encryption by chaining blocks together.

Problems:

- A ciphertext block depends on all previous blocks, meaning any changes in one block affect all subsequent blocks.
- Requires an Initialization Vector (IV), which must be shared securely.
- If the IV is sent in clear text, attackers can alter the first block and adjust the IV accordingly.

Cipher Feedback (CFB)

- **Message Handling**:
  - Used when a block cipher is required to function like a stream cipher (e.g., Telnet).
- **Process**:
  1. Take a 64-bit Initial Vector (IV) and store it in a shift register.
  2. Encrypt the IV to get EIV.
  3. XOR the leftmost r bits of the plaintext with the leftmost r bits of the EIV to produce the ciphertext (Ci).
  4. Shift the IV left by r bits and insert Ci to the right.

5. Repeat the process.

- **Uses**:
  - Often used for secure transmission of bits/bytes in real-time.

Benefits:

- Allows for stream-like encryption while using a block cipher.
- Standard supports feedback in varying bit lengths (CFB-1, CFB-8, CFB-64, CFB-128, etc.).

Problems:

- Errors in encryption propagate across several blocks.
- Commonly used for streaming data, but errors propagate after the initial mistake.

Output Feedback (OFB)

- **Message Handling**:
  - Converts a block cipher into a stream cipher.
  - Each bit of ciphertext is independent of the previous ones, which avoids error propagation.
- **Process**:
  1. Encrypt a 64-bit IV to get EIV (denoted as Ti).
  2. XOR the leftmost r bits of the plaintext with the leftmost r bits of EIV to produce the ciphertext (Ci).
  3. Shift the IV left by r bits and insert the r bits of EIV (Ti) to the right.
  4. Repeat the process.

Benefits:

- Prevents error propagation in the ciphertext.

Problems:

- More vulnerable to message stream modification.
- Requires synchronization between the sender and receiver.

- Feedback modes like CFB-64 or CFB-128 are recommended, as m-bit feedback can lead to security issues.

Counter Mode (CTR)

- **Message Handling**:
  - Converts a block cipher into a stream cipher.
  - There is no feedback involved. Instead, pseudorandomness in the key stream is achieved using a counter as the IV.
- **Process**:
  - Encrypts the counter value rather than any feedback from previous blocks.
- **Uses**:
  - High-speed network encryption.

Benefits:

- Parallel encryption can be done in hardware or software.
- Data can be processed ahead of time.
- Ideal for burst high-speed network links.
- Provides random access to encrypted data blocks.

Problems:

- Ensures that key/counter values are never reused.

RSA Algorithm

● Ron Rivest, Adi Shamir and Len Aldeman have developed this algorithm (Rivest-Shamir-Adleman).
● It is a block cipher which converts plain text into cipher text and vice versa at receiver side.
● The algorithm works as follow

1. Select two prime numbers p and q where $p \neq q$.

2. Calculate n = p * q.
3. Calculate Φ(n) = (p-1) * (q-1).
4. Select e such that, e is relatively prime to Φ(n)
   i.e. (e, Φ(n)) = 1 and 1 < e < Φ(n)
5. Calculate d = e mod Φ(n) or ed = 1 mod Φ(n).
6. Public key = {e, n}, private key = {d, n}.
7. Find out cipher text using the formula,
   C = P mod n where, P < n and
   C = Cipher text, P = Plain text, e = Encryption key and
   n=block size.d
8. P = C mod n. Plain text P can be obtain using the given formula.
9. where, d = decryption key.

Digital Signatures

1. Symmetric-Key Signatures

Concept

- A centralized authority, referred to as **Big Brother (BB)**, manages trust among users.

Key Distribution

- Each user selects a secret key and delivers it to BB in person.
- Only the user (e.g., Alice) and BB possess the user's secret key (**KA**).

Signing Process

1. When Alice wishes to send a signed plaintext message (**P**) to Bob, she generates the signature using the formula:

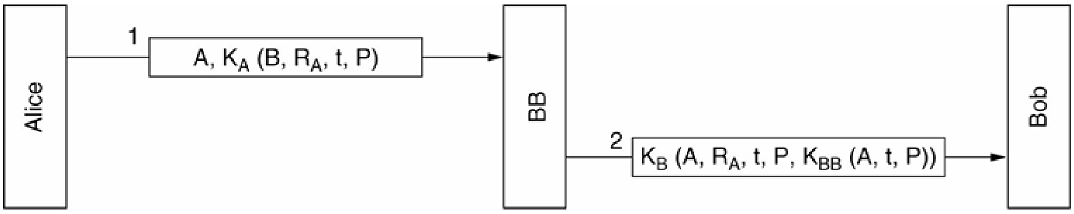
   `KA(B, RA, t, P)`

   where:

   - B: Bob's identity

- RA: A random number chosen by Alice
- t: A timestamp to ensure message freshness

2. Alice sends the encrypted message to BB.
3. BB decrypts the message and forwards it to Bob, including:
   `KBB(A, t, P)`
4. Bob receives the plaintext message and processes Alice's request.



2. Public-Key Signatures

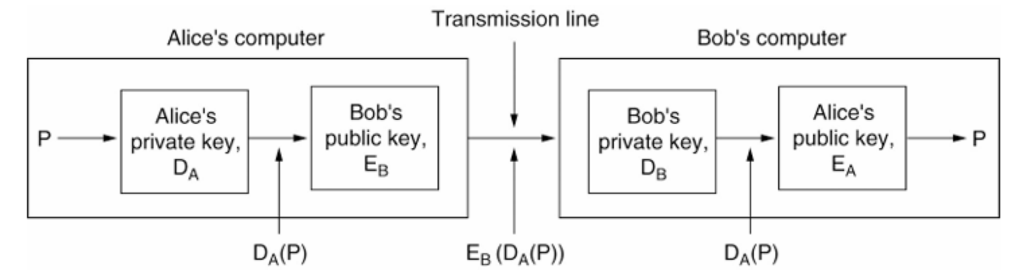
Contribution of Public-Key Cryptography

- Public-key systems enhance the signing process by utilizing a method where:

  This holds true for public-key algorithms such as RSA.
  `E(D(P)) = P`

Signing Process

1. Alice can send a signed plaintext message (**P**) to Bob by transmitting:
   `EB(DA(P))`
   where:
   - DA: Alice's private key
   - EB: Bob's public key
2. Upon receiving the message, Bob uses his private key to decrypt it, yielding:
   `DA(P)`
3. Bob then retrieves the original plaintext by applying Alice's public key encryption.



3. Message Digest

Definition

- A one-way hash function that transforms an input of arbitrary length into a fixed-length bit string, referred to as a **message digest (MD)**.

Properties of Hash Function (MD)

1. It is easy to compute (MD(P)) given (P).
2. It is nearly impossible to determine (P) from (MD(P)).
3. For a given (P), it is infeasible to find another input (P') such that (MD(P') = MD(P)).
4. A change in even one bit of the input results in a significantly different output.

4. Message Digest Algorithms

SHA-1 AND SHA-2

- Developed in 1993 by NIST.
- Processes data in 512-bit blocks and produces a 160-bit message digest.
- Mangling of bits ensures each output bit is influenced by all input bits.

MD5

- Created by Ronald Rivest in 1992.
- **Process:**
    1. The input message is padded to 448 bits (modulo 512).
    2. The original message length is appended as a 64-bit integer.
    3. The total input length is made a multiple of 512 bits.
    4. The algorithm processes 512-bit blocks, mixing them with a running 128-bit buffer.
    5. Mixing incorporates a table derived from the sine function to prevent backdoor vulnerabilities.
    6. The contents of the final 128-bit buffer constitute the message digest.