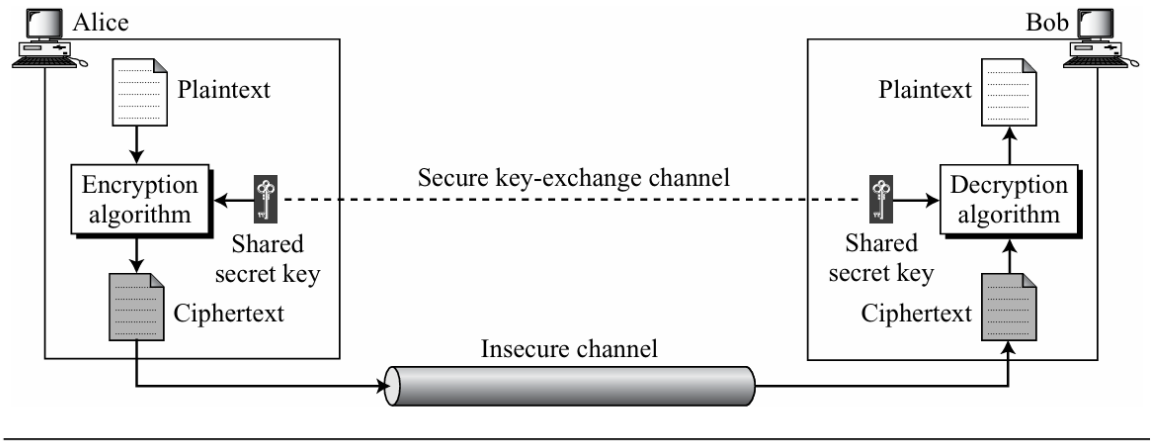

1. **Alice (Sender) Side:**

Plaintext: Alice starts with some plaintext data (the message she wants to send to Bob).

Encryption Algorithm: Alice uses an encryption algorithm that relies on a shared secret key, agreed upon via a secure key-exchange channel with Bob, to convert plaintext into ciphertext.

Ciphertext: After encryption, Alice sends the ciphertext over an insecure channel (such as the internet) to Bob.

2. **Secure Key-Exchange Channel:**

Alice and Bob must have a secure key-exchange channel to exchange the shared secret key safely. This could be done using protocols such as Diffie-Hellman or public-key cryptography. This channel ensures that no third party can intercept the key.

3. I**nsecure Channel:**

The insecure channel represents a public network where the ciphertext can be intercepted, but without the secret key, the message remains unreadable.

4. **Bob (Receiver) Side:**

Decryption Algorithm: Bob receives the ciphertext over an insecure channel and uses the shared secret key (from the secure key-exchange) with a decryption algorithm to convert it back to plaintext.

-**Plaintext:** Once the ciphertext is decrypted, Bob can read the original message.

This system ensures the confidentiality of the message even when sent over potentially insecure networks.

Symmetric Key Cryptography V/S Asymmetric Key Cryptography

1. Symmetric Key Cryptography

- **the same key** is used for both encryption and decryption.
- key **must be shared** between the sender and the receiver securely.

Encryption Scheme:

- ✓ **Sender:** The plaintext (original message) is encrypted using a secret key and an encryption algorithm.
- ✓ **Receiver:** The ciphertext (encrypted message) is decrypted using the same secret key and a decryption algorithm.

Example:

Encryption Process

1. **Choose a Key:** In this example, the key is the number of positions each letter in the plaintext will be shifted. Let's use a key of **3**.
2. **Encrypt the Message:**
   - ✓ **Plaintext:** HELLO
   - ✓ **Encryption Key:** 3

Ciphertext: KHOOR

Decryption Process

1. **Use the Same Key:** The key used for decryption is the same as the encryption key (3 in this case).
2. **Decrypt the Message:**
   - ✓ **Ciphertext:** KHOOR
   - ✓ **Decryption Key:** 3

Plaintext: HELLO

Drawbacks:

- Key distribution: Sharing the secret key securely can be challenging, especially over an insecure channel.

Use Case:

- Common algorithms: AES (Advanced Encryption Standard), DES (Data Encryption Standard).
- Used for: Fast encryption for large amounts of data (e.g., file encryption).

2. Asymmetric Key Cryptography

- uses a **pair of keys**: a public key and a private key.
- The public key is used for encryption, and the private key is used for decryption.
- Unlike symmetric key cryptography, the keys are different but mathematically related.
- In **asymmetric cryptography**, the **key pair** (public and private keys) is generated by the **receiver**, not the sender.
- In **asymmetric cryptography**, the **public key** is the key that is shared openly with anyone who wants to send encrypted data to the key owner. The public key is used for **encryption** or **verification**, while the corresponding **private key** is kept secret and is used for **decryption** or **signing**.

Example Scenario:

- **Encryption**: If Alice wants to send Bob an encrypted message, she encrypts the message using Bob's **public key**. Only Bob can decrypt the message because only he has the corresponding **private key**.

- **Digital Signatures**: If Bob wants to prove that a message came from him, he signs it using his **private key**. Anyone can verify the signature using Bob's **public key** to ensure it was Bob who signed it.

Encryption Scheme:

- **Sender:** Encrypts the message using the receiver's public key.
- **Receiver:** Decrypts the message using their private key.

Advantages:

- No need to securely share a secret key. The public key can be openly shared, while the private key is kept secret.

Drawbacks:

- Asymmetric encryption is computationally slower than symmetric encryption.

1. Ciphertext Only Attack (COA):

- **Explanation**:
  The attacker only has access to the :
  - ✓ ciphertext and
  - ✓ the encryption algorithm

but no knowledge of the :

- corresponding plaintext or
- the encryption key.

The goal is to analyze the ciphertext and figure out the plaintext or key using statistical methods or patterns in the ciphertext.

- **Example**:
  - Suppose Alice sends a message encrypted using a simple substitution cipher (where each letter in the plaintext is substituted with another letter). Bob receives the encrypted message: `WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ`.
  - The attacker intercepts this ciphertext. Knowing it's a substitution cipher, the attacker might analyze letter frequencies (since in English, 'E' is the most common letter) and compare them with the ciphertext to deduce possible substitutions and crack the code.
  - **Outcome**: The attacker might deduce that the plaintext is: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" by recognizing letter patterns in the ciphertext.

1a. Brute Force Attack:

- **Explanation**: The attacker tries all possible keys until the correct one is found.
- **Example**: Alice encrypts the message "HELLO" using a Caesar cipher with a shift of 3, producing the ciphertext `KHOOR`.
  - The attacker intercepts `KHOOR` but doesn't know the shift value.
  - Since there are only 25 possible shifts (one for each letter), the attacker tries each one systematically.
  - They start with a shift of 1 (resulting in `JGNNQ`), shift 2 (`IFMMP`), and so on, until they try shift 3 and get `HELLO`.
  - **Outcome**: The attacker successfully decrypts the message using brute force after trying all possible shifts.

2a. Statistical Attack:

- **Explanation**: The attacker uses letter frequency and statistical properties of language to crack the cipher.
- **Example**: Alice encrypts a message using a simple substitution cipher, and the ciphertext UHZSVHSNZ is intercepted by the attacker.
  - The attacker knows that "E" is the most common letter in English, so they look for the most frequent letter in the ciphertext, which is "S".
  - The attacker assumes that "S" corresponds to "E" and starts replacing it in the ciphertext.
  - They also analyze other common letters like "T" and "A", matching them with their ciphertext equivalents.
  - **Outcome**: After mapping letter frequencies, the attacker reveals the plaintext: "HELLOTHERE".

3a. Pattern Attack:

- **Explanation**: The attacker finds repeating patterns in the ciphertext to deduce the key.
- **Example**: Alice uses a Vigenère cipher to encrypt the message "ATTACKATDAWN", resulting in the ciphertext LXFOPVEFRNHR.
  - The attacker notices repeating segments in the ciphertext and guesses that the key repeats every few letters.
  - Using the **Kasiski method**, the attacker finds the key length to be 3.
  - The ciphertext is split into three parts, and frequency analysis is performed on each part separately.
  - **Outcome**: The attacker deduces that the key is "KEY", and uses it to decrypt the ciphertext, revealing "ATTACKATDAWN".

Known Plaintext Attack (KPA):

- **Explanation**:
  - ✓ The attacker knows:
  - ✓ the encryption algorithm
  - ✓ some portions of both plaintext and ciphertext pairs and
  - ✓ uses this to reverse-engineer the encryption key to decrypt other ciphertexts.
- **Example**:
  - ❖ Alice sends Bob a message, and the attacker knows that part of the plaintext in this message is "SECRET" and intercepts the ciphertext: `84D59F....`
  - ❖ With the pair "SECRET" → `84D59F`, the attacker analyzes how the encryption algorithm transformed this plaintext into ciphertext. By repeating this process with other known plaintext-ciphertext pairs, the attacker could deduce the encryption key.
  - ❖ **Outcome**: Once the attacker has the key, they can use it to decrypt other ciphertexts sent between Alice and Bob without needing any further plaintext-ciphertext pairs.

Chosen Plaintext Attack (CPA):

- **Explanation**:
  - ✓ In a CPA, the attacker can choose:
  - ✓ plaintexts to be encrypted and
  - ✓ analyze the corresponding ciphertexts.
  - ✓ This is more powerful because the attacker can design specific plaintexts to gain more information about the encryption algorithm.
- **Example**:
  - ❖ Imagine an attacker is trying to break an encryption system where Alice and Bob communicate, and the attacker has the ability to submit plaintexts to the system to get their ciphertexts.
  - ❖ The attacker submits a plaintext like "AAAAAA" or "123456" to the system and gets back the ciphertext: `AB12CD....`

- They can then submit multiple variations of the plaintext, like "AAAAAB", "123457", and compare the resulting ciphertexts to look for patterns or weaknesses in how the encryption algorithm works.
- **Outcome**: By carefully choosing which plaintexts to encrypt, the attacker might deduce the encryption method or the secret key.

Chosen Ciphertext Attack (CCA):

- **Explanation**:
  In a CCA, the attacker can choose
  - ✓ ciphertexts to be decrypted and obtain the corresponding plaintexts.
  - ✓ This type of attack is especially effective if the attacker can access a system that will decrypt ciphertexts for them.
- **Example**:
  - The attacker sends a ciphertext like `93F7AB...` to Bob's decryption system, which then returns the decrypted plaintext "HELLO".
  - By doing this repeatedly with various ciphertexts, the attacker gains more information about the encryption key or the algorithm's behavior.
  - For example, the attacker may craft a ciphertext that is slightly different from one they already know the plaintext for, and observe how small changes in the ciphertext result in changes to the plaintext.
  - **Outcome**: The attacker eventually learns enough to break the encryption and decrypt future messages or recover the key.

In monoalphabetic substitution, a character (or a symbol) in the plaintext is always changed to the same character (or symbol) in the ciphertext regardless of its position in the text. For example, if the algorithm says that letter A in the plaintext is changed to letter D, every letter A is changed to letter D. In other words, the relationship between letters in the plaintext and the ciphertext is one-to-one.