Lab Manual – Encryption & Decryption (Weak vs Strong)

Objective

- To understand how encryption and decryption work in practice.
- To compare **weak ciphers** (Caesar Cipher, XOR Cipher) with a **strong modern cipher** (AES).
- To observe how output changes with each method and why AES is secure.

Background

- **Caesar Cipher**: A substitution cipher where each letter is shifted by a fixed number. Very weak, can be brute-forced easily.
- **XOR Cipher**: Uses bitwise XOR with a key. Slightly stronger than Caesar, but still weak if key is small or reused.
- **AES (Advanced Encryption Standard)**: A modern symmetric cipher widely used in banking, HTTPS, and data security. Secure against brute-force with proper keys.

Lab Setup

- You will implement **three encryption schemes**:
  1. Caesar Cipher
  2. XOR Cipher
  3. AES (with PyCryptodome library)
- For each, you will:

1. Encrypt a given plaintext.
2. Decrypt it back to the original message.
3. Observe the differences in ciphertext output.

Lab Tasks

Task 1: Caesar Cipher

1. Use a shift value (e.g., 3).
2. Encrypt the text "HELLO WORLD".
3. Decrypt it back to the original message.
4. Try brute-forcing all possible 26 shifts and notice how easily it can be cracked.

Task 2: XOR Cipher

1. Choose a simple numeric key (e.g., 7).
2. Encrypt the text "HELLO WORLD".
3. Decrypt using the same function/key and verify the result.
4. Try changing the key and note how the output changes.
5. Observe that the ciphertext looks random but can be easily broken if the key is small.

Task 3: AES Cipher

1. Generate a random AES key (16 bytes for AES-128).
2. Encrypt the text "HELLO WORLD (AES secure!)".
3. Observe that the ciphertext is unreadable (random hex output).
4. Decrypt using the same key and IV to retrieve the original message.
5. Compare AES ciphertext to Caesar/XOR – notice that it looks much stronger.

Observations

- **Caesar Cipher**: Easy to encrypt/decrypt but trivial to crack (only 26 possible keys).
- **XOR Cipher**: Stronger than Caesar but still weak if small/repeated key is used.
- **AES**: Produces random-looking ciphertext. Decryption is only possible with the correct key + IV.

Comparison Table

Cipher	**Key Size**	**Security Level**	**Example Weakness**
Caesar Cipher	1–25 shift	Very Weak	Brute force in seconds
XOR Cipher	Small integer	Weak	Key reuse makes it breakable
AES Cipher	16/24/32 bytes	Strong	Secure if key kept secret

Feature	**Caesar Cipher**	**XOR Cipher**	**AES Cipher**
Era	Ancient (Roman times)	Simple 20th century idea	Modern (2001, NIST standard)
Type	Substitution cipher	Bitwise operation cipher	Block cipher
Key Size	1 number (0–25 shifts)	1 byte → N bytes	128/192/256 bits
Security	Very weak (easily broken)	Weak unless using one-time pad	Very strong (industry standard)
Speed	Very fast	Very fast	Fast (optimized in hardware)
Practical Use	Educational only	Obfuscation, simple protection	Real-world encryption everywhere