Subject Name: Computer Networks

Subject Code: CS6111

Lab Exercise: 03

Lab Questions:

Now you would have established a basic communication between the server and the client with the help of the scripts given.

Try to perform the following question using the same scripts but the content that the server and the client returns will vary. Modify the code accordingly and try to get the output.

Question 1 – Recap question (based on previous week content)

Design a client-server application using a connection-oriented protocol (TCP) where the server performs checksum verification for data integrity.

Description:

1. **Client-Server Communication:**
    - The client sends a message consisting of binary data to the server.
    - The server:
        - Receives the message.
        - Calculates a checksum for the received data.
        - Verifies the checksum to ensure data integrity.
        - Sends back a response to the client indicating whether the data integrity check passed or failed.
    - The messages sent and received should be displayed on both the client and server.

Steps:

1. **Client:**
    - Sends a message consisting of binary data and an accompanying checksum to the server.
2. **Server:**
    - Receives the message and checksum.
    - Calculates the checksum for the received data.
    - Compares the calculated checksum with the received checksum.
    - Sends a response back to the client indicating the result of the checksum verification (either "Integrity Check Passed" or "Integrity Check Failed").

3. **Client and Server:**
    - Display the messages sent and received.

Example:

- Client sends the message: "1010101111001101" with checksum "1010".
- Server calculates the checksum of the received data.
- Server compares the calculated checksum with the received checksum.
- Server sends back the response: "Integrity Check Passed" or "Integrity Check Failed" based on the comparison result.

Use this function to calculate the check sum of the binary sequence.

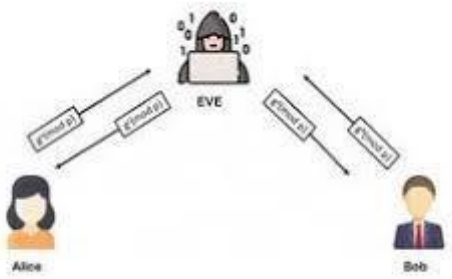
This function will take the binary sequence as input and returns the checksum as output.

```
unsigned int calculate_checksum(const char *binary_data) {
unsigned int checksum = 0;
   for (size_t i = 0; i < strlen(binary_data); ++i) {
checksum += binary_data[i] - '0';
   }
   return checksum % 16; // Assuming a 4-bit checksum
}
```

Question 2 (for this week – Execution)

Scenario:

Alice and Bob (two clients) want to communicate securely over a network. Antony (an attacker) acts as a server that relays messages between them. Antony can intercept, read, modify, or inject messages in their communication without Alice and Bob knowing.



Steps:

1. **Alice – Client 1:**
    - **Sends:** A message M1 to Bob through Eve.
    - **Content:** The message M1 contains sensitive information that Alice wants to share with Bob.
    - **Example Message:** "Hello Bob, let's meet at 5 PM."
2. **Eve (Attacker) :**
    - **Receives:** The message M1 from Alice intended for Bob.
    - **Action:**
        - Intercepts the message M1.
        - Reads the content of M1.
        - Modifies the message to create a new message M1’.
        - **Example Modification:** Changing "5 PM" to "6 PM" in the message.
    - **Sends:** The modified message M1' to Bob.
    - **Example Modified Message:** "Hello Bob, let's meet at 6 PM."
3. **Bob:**
    - **Receives:** The modified message M1' from Eve, believing it is from Alice.
    - **Content:** Bob receives the altered information and assumes it is the original message from Alice.
    - **Example:** Bob now thinks they are meeting at 6 PM, as per the message he received.

4. **Bob:**
    - **Sends:** A reply message M2 to Alice through Eve, acknowledging the meeting time.
    - **Content:** The message M2 confirms the details or asks for clarification.
    - **Example Message:** "Got it, see you at 6 PM."
5. **Eve (Attacker):**
    - **Receives:** The reply message M2 from Bob intended for Alice.
    - **Action:**
        - Intercepts the reply message M2.
        - Reads the content of M2.
        - Decides whether to modify or forward it as is.
        - **Example Decision:** Eve forwards the message unchanged.
    - **Sends:** The reply message M2 to Alice.
    - **Example Message:** "Got it, see you at 6 PM."
6. **Alice:**
    - **Receives:** The reply message M2 from Eve, believing it is from Bob.
    - **Content:** Alice is now under the impression that Bob agreed to meet at 6 PM, unaware that the original message was altered.
    - **Example:** Alice prepares to meet at 6 PM, potentially leading to confusion.

Implement this using Mediator concept where the attacker is the mediator.