Assignment 4

Diffie-Hellman Key Exchange with Man-in-the-Middle Attack

Code:-

```
#include <iostream>
#include <cmath>
using namespace std;

// Function to calculate (base^exponent) % modulo efficiently
long long mod_pow(long long base, long long exponent, long long modulo) {
    long long result = 1;
    while (exponent > 0) {
        if (exponent % 2 == 1) {
            result = (result * base) % modulo;
        }
        base = (base * base) % modulo;
        exponent /= 2;
    }
    return result;
}

int main() {
    // Constants (publicly known)
    const long long p = 23; // Prime modulus
    const long long g = 5;  // Primitive root modulo p

    // Alice's private key
    long long privateKeyAlice;
    cout << "Alice, enter your private key: ";
    cin >> privateKeyAlice;

    // Bob's private key
    long long privateKeyBob;
    cout << "Bob, enter your private key: ";
    cin >> privateKeyBob;
```

```
    // Alice computes her public key
    long long publicKeyAlice = mod_pow(g, privateKeyAlice, p);

    // Bob computes his public key
    long long publicKeyBob = mod_pow(g, privateKeyBob, p);

    // MITM attack (Eve)
    long long interceptedPublicKeyAlice;
    long long interceptedPublicKeyBob;

    // Eve intercepts the public keys
    interceptedPublicKeyAlice = publicKeyAlice;
    interceptedPublicKeyBob = publicKeyBob;

    // Attacker (Eve) replaces public keys with her own
    publicKeyAlice = interceptedPublicKeyBob;
    publicKeyBob = interceptedPublicKeyAlice;

    // Shared secret calculation
    long long sharedSecretAlice = mod_pow(publicKeyBob, privateKeyAlice, p);
    long long sharedSecretBob = mod_pow(publicKeyAlice, privateKeyBob, p);

    // Display shared secrets
    cout << "Shared secret computed by Alice: " << sharedSecretAlice << endl;
    cout << "Shared secret computed by Bob: " << sharedSecretBob << endl;

    if (sharedSecretAlice == sharedSecretBob) {
        cout << "Communication is secure. Messages are not compromised." << endl;
    } else {
        cout << "MITM attack successful! Eve has intercepted the messages." << endl;
        cout << "Eve's intercepted data: " << sharedSecretAlice << endl;
    }

    return 0;
}
```

Output:-

```
Alice, enter your private key: 17
Bob, enter your private key: 13
Shared secret computed by Alice: 10
Shared secret computed by Bob: 19
MITM attack successful! Eve has intercepted the messages.
Eve's intercepted data: 10
```

```
Alice, enter your private key: 5555
Bob, enter your private key: 55
Shared secret computed by Alice: 22
Shared secret computed by Bob: 22
Communication is secure. Messages are not compromised.
```

Conclusion:-

Diffie-Hellman key exchange is a secure way for two parties to establish a shared secret key over an insecure channel. However, the protocol is vulnerable to man-in-the-middle attacks. To prevent man-in-the-middle attacks, the Diffie-Hellman key exchange is typically used in conjunction with digital certificates.

Additional safety guidelines:-

- Use a large prime number, p. This will make the Diffie-Hellman key exchange protocol more difficult to crack.
- Use a base number, g, that is not a power of p. This will also make the Diffie-Hellman key exchange protocol more difficult to crack.