```
#include <bits/stdc++.h>

using namespace std;

// Function to calculate (base^exponent) % modulo efficiently
int mod_pow(int base, int exponent, int modulo) {
    int 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)
    int p = 23;  // Prime modulus
    int g = 5;   // Primitive root modulo p

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

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

```
// Eve's private key
int private_key_eve;
cout << "Eve, enter your private key: ";
cin >> private_key_eve;

// Alice computes her public key
int public_key_alice = mod_pow(g, private_key_alice, p);

// Bob computes his public key
int public_key_bob = mod_pow(g, private_key_bob, p);

// Eve computes his public key
int public_key_eve = mod_pow(g, private_key_eve, p);

// Shared secret calculation
int shared_secret_alice = mod_pow(public_key_bob, private_key_alice, p);
int shared_secret_bob = mod_pow(public_key_alice, private_key_bob, p);

// MITM attack
// int shared_secret_alice = mod_pow(public_key_eve, private_key_alice, p);
// int shared_secret_bob = mod_pow(public_key_eve, private_key_bob, p);

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

if (shared_secret_alice == shared_secret_bob) {
   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: " << shared_secret_alice << endl;
    }
}
```