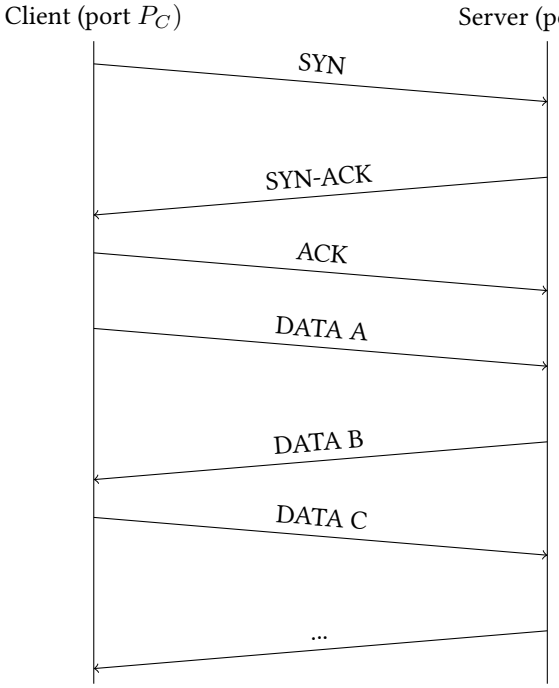
Question 1 ***Attack on TCP*** ()

Suppose that a client connects to a server, and then performs the following TCP handshake and initial data transfer:



1. Client sends initial SYN with sequence number A (usually random).
2. Server sends SYN-ACK with sequence number B (also usually random) and ACK $A + 1$.
3. Client sends ACK with sequence number $A + 1$ and ACK $B + 1$.
4. Client sends DATA A of length L_A with sequence number $A + 1$ and ACK $B + 1$.
5. Server sends DATA B of length L_B with sequence number $B + 1$ and ACK $A + 1 + L_A$.
6. Client sends DATA C of length L_C with sequence number $A + 1 + L_A$ and ACK $B + 1 + L_B$.
7. Data exchange continues until both sides are done sending data.

Q1.1 Assume that the next transmission in this connection will be DATA D from the server to the client. What will this packet look like?

Sequence number:	_____	ACK:	_____
Source port:	P_S	Destination port:	P_C
Length:	L_D	Flags:	ACK

Q1.2 You should be familiar with the concept and capabilities of a *man-in-the-middle* as an attacker who **can observe** and **can modify** traffic. There are two other types of relevant attackers in this scenario:

1. *On-path* attacker: **can observe** traffic but **cannot modify** it.
2. *Off-path* attacker: **cannot observe** traffic and **cannot modify** it.

Carol is an *on-path* attacker. Can Carol do anything malicious to the connection? If so, what can she do?

Q1.3 David is an *off-path* attacker. Can David do anything malicious to the connection? If so, what can he do?

Q1.4 The client starts getting responses from the server that don't make any sense. Inferring that David is attempting to hijack the connection, the client then immediately sends the server a **RST** packet, which terminates the ongoing connection. David wants to impersonate the client by establishing a new connection. How would he go about doing this?

Question 2 ***TLS protocol details*** ()

Depicted below is a typical instance of a TLS handshake.

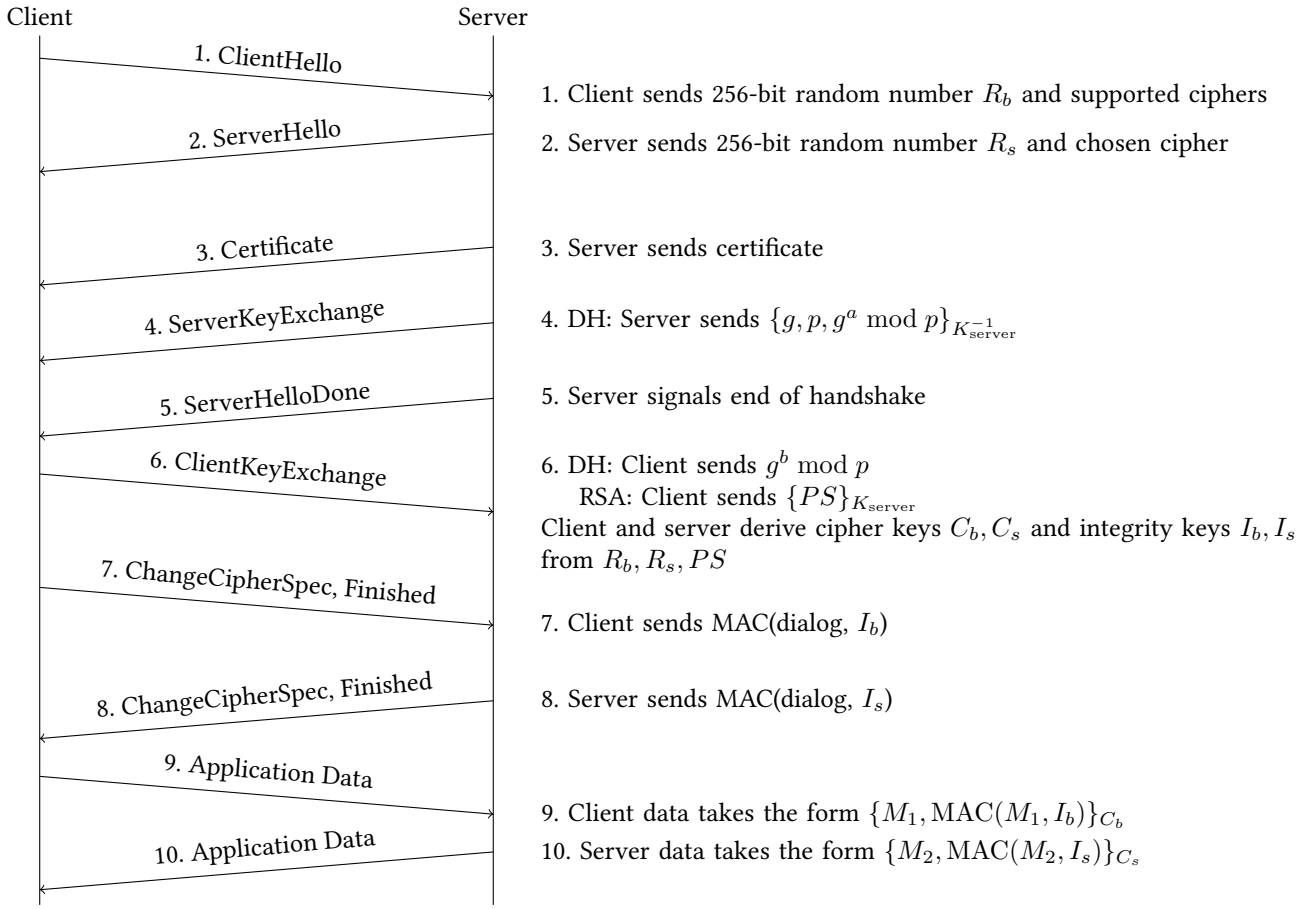


Figure 1: TLS 1.2 Key Exchange

Q2.1 What is the purpose of the *client random* and *server random* fields?

Q2.2 ClientHello and ServerHello are not encrypted or authenticated. Explain why a man-in-the-middle cannot exploit this. (Consider both the Diffie-Hellman and RSA case.)

Q2.3 Note that in the TLS protocol presented above, there are two cipher keys C_b and C_s. One key is used only by the client, and the other is used only by the server. Likewise, there are two integrity keys I_b and I_s. Alice proposes that both the server and the client should simply share one cipher key C and one integrity key I. Why might this be a bad idea?

Q2.4 The protocol given above is a simplified form of what actually happens. After step 8 (ChangeCipherSpec), the protocol as described above is still vulnerable. What is the vulnerability and how could you fix this?