CS161 Summer 2026 — Introduction to Computer Security — Discussion 8

Q1 *ARP* **(4 points)**

Recall that ARP, the Address Resolution Protocol, translates Layer 3 IP addresses into Layer 2 MAC addresses.

For this part, imagine that Alice has successfully obtained a configuration from the network's router and now she wants to communicate with Bob, whose computer is on the same LAN network.

Alice knows Bob's IP address but wants to learn his MAC address. You want to convince Alice that your MAC address (and not Bob's) corresponds to Bob's IP address, causing messages intended for Bob to be sent to you instead. You, Alice, and Bob are part of the same LAN network and, apart from the router, there are no other machines on the network.

Assume that:

- Your computer has IP address 10.10.10.66 and your MAC address is 66:66:66:66:66:66
- Alice's IP address is 10.10.10.77 and her MAC address is 77:77:77:77:77:77
- Bob's IP address is 10.10.10.88 and his MAC address is 88:88:88:88:88:88
- The network router's IP address is 10.10.10.5 and its MAC address is 99:99:99:99:99:99

Q1.1 (1 point) Alice broadcasts to everyone else on the LAN: "What is the MAC address of 10.10.10.88?" Recall that 10.10.10.88 corresponds to Bob's IP address.

What values for the IP and MAC address could you include in your response to Alice to cause her messages intended for Bob to be sent to you instead?

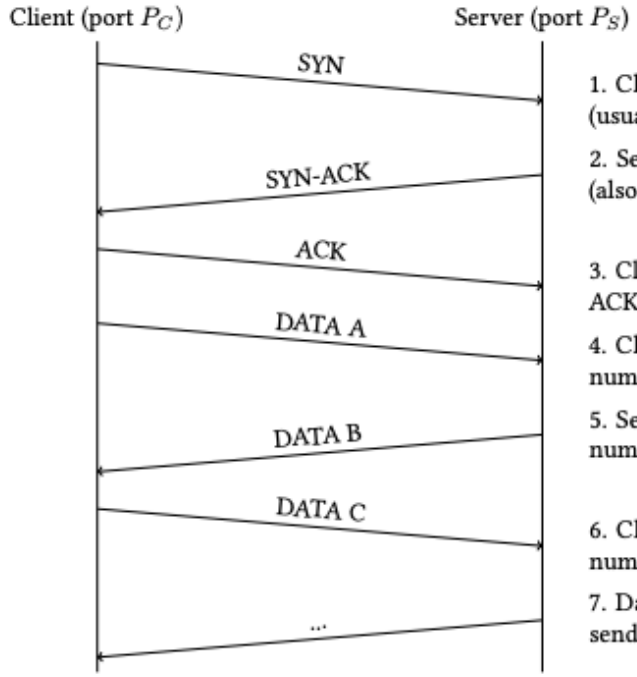
Q1.2 (1 point) How would your spoofed response to Alice change if Bob was outside the LAN that you and Alice are both part of?

Q1.3 (1 point) Which defenses exist against such an ARP-spoofing attack?

Q1.4 (1 point) You decide to use a network switch to prevent further ARP spoof attacks.

Explain how a network switch prevents such attacks in the following two cases: (1) the switch knows Bob's IP to MAC address mapping and (2) the switch does **not** know Bob's IP to MAC address mapping.

Q2 *Attack on TCP* **(4 points)**



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

Q2.1 (1 point) 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

Q2.2 (1 point) 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?

Q2.3 (1 point) David is an *off-path* attacker. Can David do anything malicious to the connection? If so, what can he do?

Q2.4 (1 point) 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?

Q3 *TLS Protocol Details* **(4 points)**

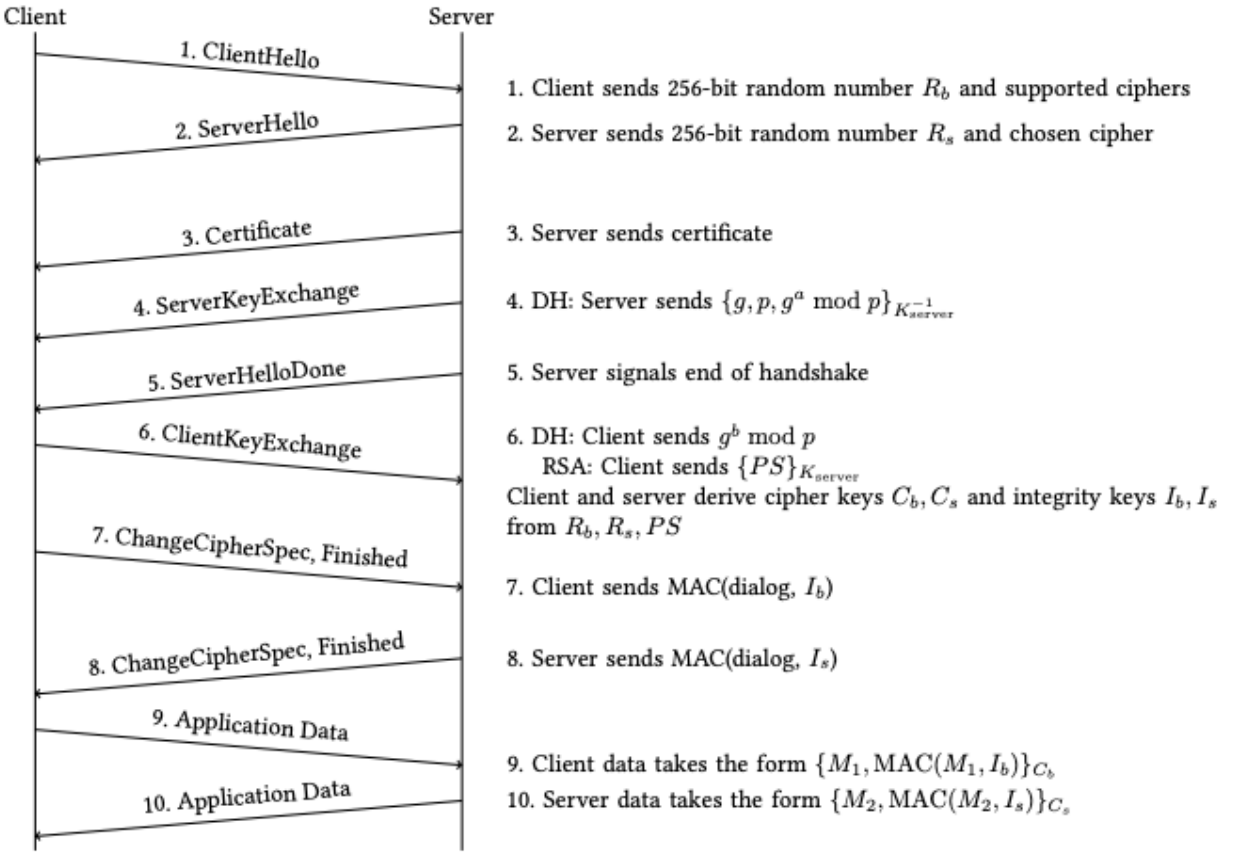


Figure 1: TLS 1.2 Key Exchange

Q3.1 (1 point) What is the purpose of the *client random* and *server random* fields?

Q3.2 (1 point)

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.)

Q3.3 (1 point) 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?

Q3.4 (1 point)

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?