

- P18. Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair (K_B^+, K_B^-) , and Alice has Bob's certificate. But Alice does not have a public, private key pair. Alice and Bob (and the entire world) share the same hash function $H(\cdot)$.
- In this situation, is it possible to design a scheme so that Bob can verify that Alice created the message? If so, show how with a block diagram for Alice and Bob.
 - Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how with a block diagram for Alice and Bob.
- P19. Consider the Wireshark output below for a portion of an SSL session.
- Is Wireshark packet 112 sent by the client or server?
 - What is the server's IP address and port number?
 - Assuming no loss and no retransmissions, what will be the sequence number of the next TCP segment sent by the client?
 - How many SSL records does Wireshark packet 112 contain?

The screenshot shows the Wireshark interface with the following details for packet 112:

- Frame 112 (258 bytes on wire, 258 bytes captured)**
- Ethernet II, Src: IBM10:60:99 (00:09:6b:10:60:99), Dst: All-MSRP-routers_00 (00:00:0c:07:ac:00)**
- Internet Protocol, Src: 128.238.38.162 (128.238.38.162), Dst: 216.75.194.220 (216.75.194.220)**
- Transmission Control Protocol, Src Port: 2271 (2271), Dst Port: https (443), Seq: 79, Ack: 2785, Len: 204**
- Secure socket Layer**
 - SSL3 Record Layer: Handshake Protocol: Client Key Exchange**
 - Content Type: Handshake (22)
 - Version: SSL 3.0 (0x0300)
 - Length: 132
 - Handshake Protocol: Client Key Exchange**
 - Handshake Type: Client Key Exchange (16)
 - Length: 128
 - SSL3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec**
 - Content Type: Change Cipher Spec (20)
 - Version: SSL 3.0 (0x0300)
 - Length: 1
 - Change Cipher Spec Message
 - SSL3 Record Layer: Handshake Protocol: Encrypted Handshake Message**
 - Content Type: Handshake (22)
 - Version: SSL 3.0 (0x0300)
 - Length: 56
 - Handshake Protocol: Encrypted Handshake Message

The hex dump at the bottom shows the raw bytes of the packet, with the first few bytes being `fd 1f c2 d9 00 00 16 03`.