Task 1: Consider the following interaction with Trudy (T) - god on the wire, in picture.

Goal: Trudy is attempting to impersonate as Alice to Bob

Step 1: Trudy initiates the protocol as Alice - Bob

$T \rightarrow B : \{N_T\}_B , \{K_T\}_B$

$B \rightarrow T : N_T , \{N_B\}_A , \{K_2\}_A$

$T \rightarrow B : N_B$

Problem: Trudy would need N_B to complete the handshake and K_2 to completely derive the shared key established during handshake.

Step 2: Finding out N_B

$T \rightarrow A : \{N_B\}_A , \{K_T\}_A$

$A \rightarrow T : \mathbf{N_B} , \{N_A\}_B , \{K_2\}_B$

T Aborts

Step 3: Finding out K_2 - not essential to breaking the protocol but allows future communication using the shared key established.

$T \rightarrow A : \{K_2\}_A , \{K_T\}_A$

$A \rightarrow T : \mathbf{K_2} , \{N_A\}_B , \{K_2\}_B$

T Aborts

Now Trudy can use N_B, K_2 figured out from Step 2 & 3 to complete the protocol handshake and at the end of endshake, assumptions B, C fail and Alice never completes the handshake.

Task 2: Modified protocol - to fix the bug highlighted above.

The problem with broken protocol is that it leaks info by sending solved responses (for a challenge from the other party) in plain text at different phases - Phase 2 & 3 of the 3-phase handshake.

Modified 3-phase protocol:

$A \rightarrow B : \{N_A\}_B , \{K_1\}_B$
$B \rightarrow A : \{N_A\}_A , \{N_B\}_A , \{K_2\}_A$
$A \rightarrow B : N_B$

The modified protocol addresses this issue by **encrypting** the responses **with public key** of the other party **in 2nd phase**, while still using a **plain response** in the **3rd phase**.

These modifications ensures that:

- A genuine handshake initiator, say Alice, can decode the response to verify that Bob has indeed solved the challenge correctly in Phase 2. Thus Phase 2 verifies Bob's (non-initiator) authenticity.
- If the initiator were an intruder, say Trudy, they would never be able to complete Phase 3 of the handshake since unlike in broken protocol, Phase 2 doesn't leak plain responses needed for Phase 3.
- This suggests that the patch is indeed a fix and not another form of the broken protocol.

Task 3: Verifiable authenticity for the sender of a message

Notation: $[M_A]_A$ - Message from A signed with private key of A.

$A \rightarrow B : \{N_A\}_B , \{K_1\}_B$

$B \rightarrow A : \{N_A\}_A , \{N_B\}_A , \{K_2\}_A$

$A \rightarrow B : N_B$

Authentication Phase
(parties establish a shared key for secure communication)

$A \rightarrow B : \{M_A\}_K , [M_A]_A$

$B \rightarrow A : \{M_B\}_K , [M_B]_B$

….

Messaging Phase
(Message is encrypted using shared secret key and a signature of the contents is generated using their private key)

Authenticity of the message can be verified at any point by the receiver through:

- Decoding the message contents (say $\{M_A\}_K$ -> M_A)
- Using Sender's public key on the decoded message and verify that it indeed complies with the signature sent along (here $[M_A]_A$)