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 and K_2 to complete the 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

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

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

T Aborts

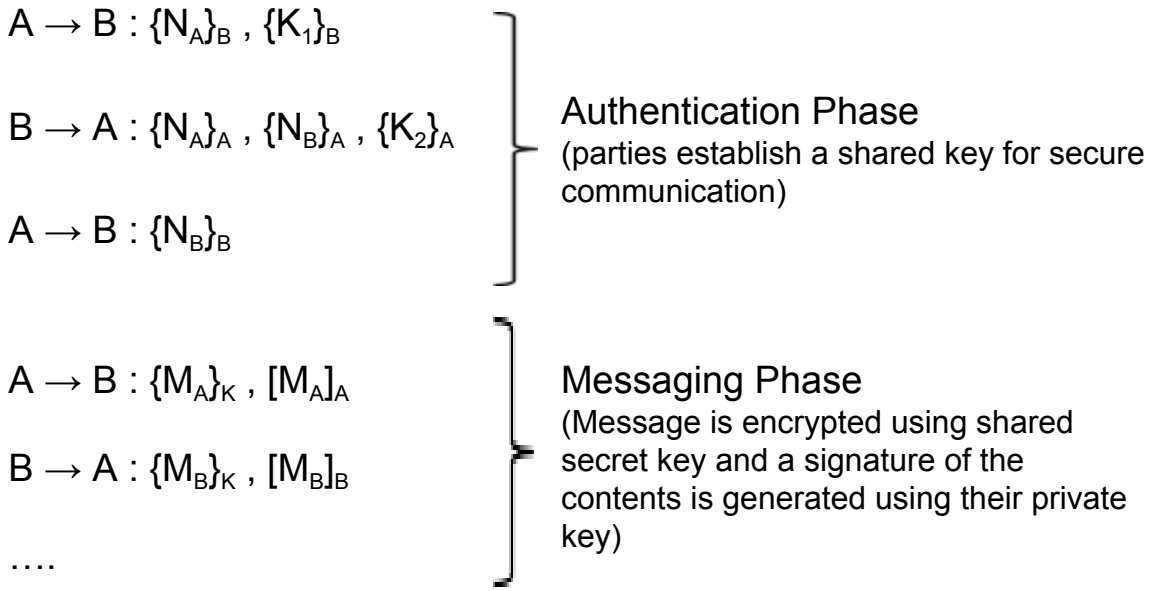
Now Trudy can use N_B, K_2 figured out from Step 2 & 3 to complete the protocol handshake and at the end of endshake, assumptions B, C fail and Alice never completes the handshake.

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

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

The problem with broken protocol is that it leaks info by sending decoded challenges as plain text. The modified protocol address this issue by encoding the responses with public key of the other party. Now A/B can decode the response to verify that B/A has indeed solved the challenge.

Task 3: Verifiable authenticity for the sender of a message



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

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