Answers to Exercise 1

1- What properties should a nonce satisfy (at the generation time)?

☐ **a) freshness**

☐ b) known to all participants

☐ **c) secret**

☐ d) easy to compute

2- Which of the following can be used to make replay attacks in authentication protocols harder?

☐ **a) Nonce**

☐ **b) Monotonically increasing sequence number**

☐ **c) Time stamp**

☐ **d) Random number used no more than once**

3- Which notation are we using for symmetric encryption?

☐ a) $\{M\}_{\mathrm{inv}K}$

☐ b) $\{M\}_K$

☐ **c) $\{|M|\}_K$**

☐ d) none of the mentioned

4- On which assumption is the security of the Diffie-Hellmann Key Exchange based?

☐ **a) Computing discrete logarithms**

☐ b) Computing prime factorization

☐ c) Computing cubic roots

☐ d) Exponentiation

Answers to Exercise 2

1. *Anonymity:* An attacker can easily tell who has voted from the first message.

2. *Confidentiality:* Confidentiality should be provided as long as the server private key $\mathrm{inv}(K_S)$ remains secure.

3. *Authentication:* It is reasonable for S to assume that the answer Ans_Q came from A, since only A should have been able to read and return the correct N_S. However, A cannot be sure that she is answering the right question. Nothing authenticates S to A, so the intruder can pose as S and make A think she has voted when in fact her answers never got to the real server.

4. *Multiple Votes:* There is no obvious way for a malicious user to vote more than once on the same question without compromising the private keys of someone else.

5. *Availability:* Availability generally cannot be guaranteed in this setting. If the intruder is able to block all messages, then he can easily mount a Denial of Service (DoS) attack.

6. *Integrity:* Integrity often follows from authentication, so the situation is similar: S is assured that Ans_Q has not been tampered with, but A cannot say anything about what she receives from S.

Answers to Exercise 3

One attack works and can, e.g., be carried out as follows:

E picks a very large number Seq (e.g., $Seq = 2^{32} - 3$). This minimizes the risk that Seq was already used in a previous communication between A and B.

1. $E \rightarrow B: A, Seq$
2. $B \rightarrow E: \{|Seq + 1, A|\}_{\mathrm{sk}(A,B)}$
   a. $E \rightarrow A: B, Seq + 1$
   b. $A \rightarrow E: \{|Seq + 2, B|\}_{\mathrm{sk}(A,B)}$
3. $E \rightarrow B: \{|Seq + 2, B|\}_{\mathrm{sk}(A,B)}$

Now, B believes to talk to A while in fact she talks to E. Note that E has not learned the symmetric $sk_{A,B}$ key shared between A and B. Thus, the attacker cannot complete the second protocol run, as she cannot create the message required in the last step of this run:

3'. $E \rightarrow A: \{|Seq + 3, A|\}_{\mathrm{sk}(A,B)}$

This type of attack can be prevented by making the messages in step 2 and step 3 syntactically different, e.g., by changing the second step to:

2- $B \rightarrow A: \{|Seq + 1, A, B|\}_{\mathrm{sk}(A,B)}$

Note that we, alternatively, also could have added A to the third message – as long as we only change one message and not both.