Protocols

Questions and answers

1. Explain the two major concepts of authentication and its purpose in basic words?
   - Data origin authentication: ensure integrity
   - Entity authentication: identify involved partys
2. If you are presented a protcol what are the criteria you would judge it with?
   - Integrity (Authenticity of data origin)
   - Freshness (eg. nonce)
   - Liveness (not old, eg. time-stamp)
   - Protocol must embed identities (e.g.: message is for Bob.. so no other package is not valid for other recievers)
3. Referring to the following agreement. Does it fullfil all of the above criteria
   - Alice => Bob: {Session-Key}Public-KeyBob
   - KeyMAC = HMAC(Session-Key || 'MAC')
   - KeyENC = HMAC(Session-Key || 'ENC')
   - Answer
     - Authenticity of data origin: yes via KeyMAC
     - Freshness: I belive no as Alice is not bound to generate a new Session-Key everytime, at least in this example (in reality some kind of nonce would be involved and then "Freshness" would be ok)
     - Liveness: not fullfiled
     - Protocol: yes halfway, the message is for Bob as only he can decipher session key. However, Alice identity is not secured.
4. Referring to the Needham-Schroeder protocol. How can Bob tell the first message he receives of Alice {Session-Key, Alice}KeyBob is a fresh one?
   - He cannot. It could be from a previously established connection. However, Bob will try to ensure that this is a first time message by sending an encrypted Nonce to Alice and as only Alice can decipher it, only she could respond correctly.

Notes and varia

mainly from video (a bit redundant with answers to questions)

Authentication

- Data origin authentication: integrity (usually via MACS)
- Entity authentication (identify parties)
  - Uniliteral (only one party)
  - Mutual

Authentication requirements

- Authenticity of data origin
- Freshness (eg. nonce)
- Liveness (not old, eg. time-stamp)

- Protocol must embed identities (e.g.: message is for Bob.. so no other package is not valid for other recievers)

key aggreement -> how to get both partys to have the same session key (also see diagrams in video)

- public/private key aggrement
    - Alice will generate a session key and encrypt it with public key from Bob (from bobs certificate).
        - Attention: session key is not directly used. (as this would provide an attacker with info)
        - for integrity: HMAC(Session-Key || MAC)
        - for encryption: HMAC(Session-Key || ENC)
- Diffi-Hellmann -> secret is calculated on each side...
- Needham-Schroeder Protocol
    - based on trusted third party
    - vulnerarble to replay attacks
    - https://de.wikipedia.org/wiki/Needham-Schroeder-Protokoll
- Kerberos
    - based on trusted third party