

CMSC 414: HW 4

1. (text 11.3) In section 11.3.1, we discuss various ways for forming a session key. Remember that R is the challenge sent by Bob to Alice, and A is Alice's secret, which Bob also knows. Which of the following are secure for a session key?
- $A \oplus R$
 - $\{R + A\}_A$
 - $\{A\}_A$
 - $\{R\}_{R+A}$

Solution

$A \oplus R$ is not secure: eavesdropper who discovers it also discovers A .
 $\{R + A\}_A$ is secure.
 $\{R\}_{R+A}$ is secure.
 $\{A\}_A$ is not secure: it is the same for all sessions.

2. (text 11.4) Design a variant of Otway-Rees that only has one nonce generated by Alice and one nonce generated by Bob. Explain why it is still as secure.

Solution

Essentially, replace N_C by $K_A\{N_A, "A", "B"\}$

| | A (Alice) | KDC | B (Bob) |
|---|---|--|--|
| 1 | <ul style="list-style-type: none"> ▪ generate nonces N_A ▪ send [A, B, $K_A\{N_A, A, B\}$] to B | | |
| 2 | | | <ul style="list-style-type: none"> ▪ generate nonce N_B ▪ send [$K_B\{N_B, K_A\{N_A, A, B\}, A, B\}$] to KDC |
| 3 | | <ul style="list-style-type: none"> ▪ invent session key K_{AB} ▪ extract N_A, N_B • send [$K_A\{N_A, K_{AB}\}, K_B\{N_B, K_{AB}\}$] to B | |
| 4 | | | <ul style="list-style-type: none"> ▪ send $K_A\{N_A, K_{AB}\}$ to A |
| 5 | send $K_{AB}\{\text{anything recognizable}\}$ to B | | |
| | <----- A and B establish data session key (eg, $(K_{AB}+1)\{R_1 \oplus R_2\}$ -----> | | |

When the KDC extracts N_A and N_B (step 3), it ensures that B is making the request, that A made the request that B is forwarding inside B's request, and that A and B want to talk to each other.

The nonce N_B in B's ticket ensures that the ticket is freshly created by KDC.
 The nonce N_A in A's ticket ensures that the ticket is freshly created by KDC.

3. (text 11.8) Design a two-message authentication protocol that achieves both mutual authentication and establishment of a session key, assuming that Alice and Bob know each other's public keys.

Solution

Alice picks a session key K and sends a signed message containing K encrypted by Bob's public key and a timestamp ts .

Bob responds with the timestamp ts encrypted with K .

| A (Alice) | B (Bob) |
|--|---|
| <ul style="list-style-type: none"> send [$\{K\}_B, ts]_A$ to B | <ul style="list-style-type: none"> send [$K\{ts\}$] |

4. (text 11.10) Section 11.4 *Mediated Authentication (with KDC)* describes several protocols. For each of those protocols, describe which nonces have to be unpredictable (i.e., cannot be sequence numbers).

Solution

Protocol 11-18 (Needham-Schroeder): N_1 must be unpredictable (otherwise Trudy can impersonate the KDC by giving Alice an old ticket to Bob with a key that Trudy had stolen earlier).

Protocol 11-19 (Expanded Needham-Schroeder) and protocol 11-21 (Kerberos): N_1 must be unpredictable.

Protocol 11-20 (Otway-Rees): N_C must be unpredictable (explained in text).

5. (text 16.1) Which of the following properties does protocol 16-2 have: perfect forward secrecy (PFS); escrow foilage against passive attacks; escrow foilage against active attacks; identity hiding; perfect forward secrecy for identity hiding. Assume private encryption keys are escrowed and private signature keys are not escrowed.

Solution

PFS; escrow foilage against passive attacks; escrow foilage against active attacks (unless signature key is escrowed); no identity hiding.

-
6. (text 16.1) Repeat problem 5 for a modified form of protocol 16-2 in which the first two messages are encrypted with the other end's public key rather than signed by the transmitter's private signature key. So in message 1, Alice sends {"Alice", $g^a \bmod p$ } encrypted with Bob's public key, and Bob in message 2 sends {"Bob", $g^b \bmod p$ } encrypted with Alice's public key.

Solution

PFS; escrow foilage against passive attacks; no escrow foilage against active attacks; identity hiding; no PFS for identity hiding.

-
7. (text 16.1) Repeat problem 5 for protocol 16-4.

Solution

PFS; escrow foilage against passive attacks; escrow foilage against active attacks (unless signature key is escrowed); identity hiding; PFS for identity hiding; active attacker can discover Alice's identity.

-
8. (text 16.1) Repeat problem 5 for protocol 16-9, where Alice and Bob share a secret key S .

Solution

PFS; escrow foilage against passive attacks; no escrow foilage against active attacks; no identity hiding.

-
9. (text 16.1) Repeat problem 5 for the protocol where each side sends a nonce encrypted with the other's public encryption key, and the resulting session key is the \oplus of the two nonces.

Solution

no PFS; no escrow foilage; no identity hiding.

-
10. (text 16.4) Referring to section 16.6 *Endpoint Identifier Hiding*, modify protocol 16-4 to hide the initiator's identity rather than the target's identity.

Solution

Have Bob append everything he would have said in message 4 to message 2.

-
11. (text 16.5) Referring to section 16.6 *Endpoint Identifier Hiding*, show a protocol that hides both identifiers from an active attacker, assuming that Alice (the initiator) already knows Bob's public key.

Solution

Have Alice send her name, her public encryption key certificate, and her Diffie-Hellman value encrypted with Bob's public key.

Bob replies encrypted with Alice's public key.

-
12. (text 16.7) Devise a protocol based on a pre-shared secret key that hides identities and gives PFS for identity hiding. Make two variants, one in which an active attacker can learn only the initiator's identity, and one in which an active attacker can learn only the target's identity.

Solution

Variant where attacker only learn's initiator identity

Messages 1 and 2: Diffie-Hellman exchange.

Msg 3: initiator sends its id and proof of knowledge of shared key, encrypted with the Diffie-Hellman key.

Msg 4: target sends its id and proof of knowledge of the shared key encrypted with the Diffie-Hellman key.

Variant where attacker only learn's target's identity

Msg 1: Diffie-Hellman number.

Msg 2: (a) Diffie-Hellman number and (b) target's id and proof of knowledge of shared key, encrypted with Diffie-Hellman key.

Msg 3: initiator sends its id and proof of knowledge of shared key, encrypted with the Diffie-Hellman key.
