Problem 1. [15 points]

Part a. [7 points]

Does *Inv* A_1 hold, where

A_1 : ((j in hst.keys) and j > 0 and hst[j] = [A,p]) $\Rightarrow$ hst[j-1] = [B,1,p]

Solution

No.

Here is a counter-example evolution.

1. Protocol goes through steps Initial, A.1, B.1, Z.1, B.2, starting with A sending msg [A,B,1,enc(kA,[A,B,xA])].
   State: A.nL = xA; A.key = kA; A.t at A.2; B.t at B.3; B.kAB = p; hst = [[B,1,p]];
   [B,A,eA] in channel where eA = enc(kA,[xA,p]).
2. Attacker intercepts the final message, [B,A,eA], in step 1.
   Attacker sends [A,B,2,grbg].
   B.t receives this message, executes B.3 unsuccessfully, returns to B.1.
3. Attacker replays msg 1, [A,B,1,enc(kA,[A,B,xA])]. Protocol goes through steps B.1, Z.1, B.2.
   State: A.nL = xA; A.key = kA; A.t at A.2; B.t at B.3; B.kAB = q and q $\neq$ p; hst = [[B,1,p],[B,1,q]];
   [B,A,fA] in channel where fA = enc(kA,[xA,q].
4. Attacker replaces msg [B,A,fA] with msg [B,A,eA] (obtained in step 2).
5. A.t receives msg 4, executes A.2 successfully.
   State: hst = [[B,1,p],[B,1,q],[A,p]] and q $\neq$ p.
   A_1 false.

Part b. [8 points]

Does *Inv* A_2 hold, where

A_2 : ((j in hst.keys) and j > 0 and hst[j] = [B,2,p]) $\Rightarrow$ hst[j-1] = [A,p]

Solution

No.

Here is a counter-example evolution.

1. Protocol goes through steps Initial, A.1, B.1, Z.1, B.2, starting with A sending msg [A,B,1,enc(kA,[A,B,xA])].
   State: A.nL = xA; A.key = kA; A.t at A.2; B.t at B.3; B.kAB = p; hst = [[B,1,p]];
   [B,A,eA] in channel where eA = enc(kA,[xA,p]).
2. Attacker intercepts the final message, [B,A,eA], in step 1.
   Attacker sends [B,A,grbg] (prelude to doing getPwdA).
   A.t receives this message, executes A.2 unsuccessfully, returns to A.1.
3. Attacker executes getPwdA; obtains kA.
   Attacker decrypts eA using kA to get p.
   Attacker sends [A,B,2,enc(p,'HELLO')].
4. B.t receives msg 3, executes B.3 successfully.
   State: hst = [[B,1,p],[B,2,p]].
   A_2 false.

Problem 2. [15 points]

Part a. [7 points]

Does *Inv* A_1 hold, where

A_1 : `((j in hst.keys) and j > 0 and hst[j] = [A,p])` $\Rightarrow$ `hst[j-1] = [B,1,p]`

Solution

No.

The evolution in problem 1a also works here.

Part b. [8 points]

Does *Inv* A_2 hold, where

A_2 : `((j in hst.keys) and j > 0 and hst[j] = [B,2,p])` $\Rightarrow$ `hst[j-1] = [A,p]`

Solution

No.

The evolution in problem 1b also works here.

Problem 1a: Attempt to prove *Inv* A_2 holds

First prove that master keys are not exposed and that the keys at the users and the kdc are equal.

- *Inv* ψ(A.key) holds.

  (Holds initially. The only A.key expressions sent by the users and kdc are: enc(A.key, [A,B,xA]) where xA is random; and enc(A.key, [xA,kAB]) where kAB is random.)
- *Inv* A.key = Z.keyA holds.

  (Holds initially. Preserved by getPwdA.)
- *Inv* ψ(B.key) and *Inv* B.key = Z.keyB hold.

  (Proof similar to that of *Inv* ψ(A.key) and *Inv* A.key = Z.keyA.)

Now to attempt to prove *Inv* A_2.

1. Suppose B appends [B,2,p] to hst at time b_0.
   So B.t is at B.3 and receives [A,B,2,enc(p,'HELLO')] where p = B.kAB.
2. So B's previous step is B.2, say at time b_1.
   B receives [Z,B,enc(keyB,[xB,p]),.], where xB = B.nL, and appends [B,1,p] to hst.
3. So B's previous step is B.1, say at time b_2.
   B receives [A,B,1,f], sets B.nL to random value xB, and sends [B,Z,enc(B.key,[A,B,xB,f])].
4. Because xB is random and *Inv* ψ(B.key) holds, Z generated entry enc(keyB,[xB,p]) in msg 2 at some time z_0 during $[b_2, b_1]$. So Z sends [Z,B,enc(keyB,[xB,p]),.], where xB = B.nL at z_0.

   So at z_0, Z receives [B,Z,enc(B.key,[A,B,xB,enc(A.key,[A,B ,xA]))] for some xA. Entry 2 of this message has to be generated by B (because *Inv* ψ(B.key)) holds). For this value xB, B generates such an entry only once.

   Hence in step 3, f equals enc(A.key,[A,B ,xA]).
5. Hence at some time a_0 before b_2, A generated f and set A.nL to the random value xA. (Attacker could not have generated this entry because *Inv* ψ(A.key) holds.)
6. At time z_0, ψ(p) holds (because attacker does not have B.key).

   If ψ(p) continues to hold at b_0, then attacker could not have generated entry enc(p,'HELLO') in step 1 message. Hence it was generated by A at some time a_1 during $[z_0, b_0]$, at which point A adds [A,p] to hst. After that A has not updated hst. So A_2 holds.

   If ψ(p) does not hold at b_0, then attacker can generate the message in step 1. So we have to show that this is not possible. Attacker can obtain p only by obtaining the A.key after time z_0. Attacker can get A.key after time z_0 using getPwdA, but for that it has to move A.t to A.1....