Rapport TP AVISPA

Francesco **Furfaro**, Aurélien **Monnet-Paquet**, M2 CySec.

Exercice 1 :

1. ./avispa ../tp/NSPK_1.hlpsl --ofmc

```
% OFMC
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/m/monnetpa/Documents/avispa-1.1//testsuite/results/NSPK_1.if
GOAL
  as_specified
```

2.
```
GOAL
  secrecy_of_nb
ATTACK TRACE
i -> (a,6): start
(a,6) -> i: {Na(1).a}_ki
i -> (b,3): {Na(1).a}_kb
(b,3) -> i: {Na(1).Nb(2)}_ka
i -> (a,6): {Na(1).Nb(2)}_ka
(a,6) -> i: {Nb(2)}_ki
i -> (i,17): Nb(2)
i -> (i,17): Nb(2)
```

Interprétation :
L'attaquant (i) envoie un signal start à Alice, elle chiffre donc un nonce Na et son identité avec la clef publique de l'attaquant. L'attaquant envoie ensuite le Na et a à Bob, chiffré avec la clef publique de Bob. Bob envoie à l'attaquant Na et Nb chiffré avec la clef publique d'Alice puis l'attaquant le transmet à Alice. Alice lui envoie finalement Nb chiffré avec la clé de l'attaquant, il récupère ainsi Nb, il a ainsi atteint son but.

3. ./avispa ../tp/NSPK_3.hlpsl --ofmc

```
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/Documents/avispa-1.1//testsuite/results/NSPK_3.if
GOAL
  authentication_on_bob_alice_na
BACKEND
  OFMC
ATTACK TRACE
i -> (a,6): start
(a,6) -> i: {Na(1).a}_ki
i -> (b,3): {Na(1).a}_kb
(b,3) -> i: {Na(1).Nb(2)}_ka
i -> (a,6): {Na(1).Nb(2)}_ka
(a,6) -> i: {Nb(2)}_ki
i -> (b,3): {Nb(2)}_kb
```

Interprétation :
Dans ce cas, on vois que l'attaquant trouve bien une solution pour s'authentifier à la place d'Alice. L'attaquant sert simplement de relais pour l'authentification, il contact Alice pour avoir le nonce et son identité a puis l'envoie à Bob etc.

Comparaison avec le précédent :
Le but des deux attaques n'est pas le même, dans le premier l'attaquant cherche un secret partagé (nb) alors que dans le deuxième il cherche à s'authentifier à la place d'Alice.

4.
Une contre mesure et d'ajouter comme goal le secret nb, et l'identité de l'émetteur quand il reçoit une demande d'authentification. Dans ce cas, Bob ajoute son identité dans le message pour Alice.

Alice - Bob

1. A -> B: {Na,A}_Kb
2. B -> A: {B,Na,Nb}_Ka
3. A -> B: {Nb}_Kb

secrecy_of na, nb

Exercice 2 :

1. ./avispa Ex2_A.hlpsl --ofmc

```
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/Documents/avispa-1.1//testsuite/results/Ex2_A.if
GOAL
  secrecy_of_nanb
BACKEND
  OFMC
ATTACK TRACE
i -> (a,3): start
(a,3) -> i: a.{Na(1)}_kb
i -> (b,10): i.{Na(1)}_kb
(b,10) -> i: b.{Na(1).Nb(2)}_ki
i -> (a,3): b.{Na(1).x253}_ka
(a,3) -> i: {zero.Msg(3)}_(Na(1).x253)
i -> (i,17): Na(1).x253
i -> (i,17): Na(1).x253
```

Analyse et interprétation du résultat :
Ici, Alice envoie un nonce à l'attaquant chiffré avec la clef publique de Bob, il remplace l'identité d'Alice par la sienne, et envoie le message à Bob. Bob répond à l'attaquant avec la clé de session chiffré avec la clé publique de l'attaquant. L'attaquant obtient donc la clef de session.

2. ./avispa Ex2_B.hlpsl --ofmc

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/Documents/avispa-1.1/testsuite/results/Ex2_B.if
GOAL
  as_specified
BACKEND
  OFMC
```

Analyse et interprétation:
L'attaque n'est plus possible car cette fois ci, la clef de session n'est plus transmise en entier, c'est à dire que Na et Nb ne sont plus envoyé ensemble. L'attaquant n'est plus en mesure de récupérer la clef de session.

3. ./avispa Ex2_C.hlpsl --ofmc

```
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/Documents/avispa-1.1//testsuite/results/Ex2_C.if
GOAL
  weak_authentication_on_k2
BACKEND
  OFMC
ATTACK TRACE
i -> (a,6): start
(a,6) -> i: a.{Na(1)}_ki
i -> (b,3): a.{Na(1)}_kb
(b,3) -> i: b.{Nb(2)}_ka
i -> (a,6): i.{Nb(2)}_ka
(a,6) -> i: {zero.Msg(3)}_(Na(1).Nb(2))
i -> (b,3): {zero.Msg(3)}_(Na(1).Nb(2))
(b,3) -> i: {one.Msg(3)}_(Na(1).Nb(2))
```

Analyse et interprétation :
L'attaquant parvient à se faire passer pour Alice auprès de Bob et pour Bob auprès d'Alice. Il s'authentifie comme étant Alice.

4.
La solution ici est de chiffrer aussi l'identité de l'émetteur avec le nonce Na ou Nb.

1. A -> B: {A,Na}_Kb
2. B -> A: {B,Nb}_Ka
3. A -> B: {zero,Msg}_(Na,Nb)
4. B -> A: {one,Msg}_(Na,Nb)

Exercice 3 :

1. ./avispa NSPKxor.hlpsl --ofmc
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS

Analyse et interprétation :
Dans ce cas, Bob ajoute simplement une opération logique (Xor) avec le nonce de Alice et son identité. L'attaquant ne peut trouver d'attaque ici, le goal est sur l'authentification et sur la secrecy de na.

2.
Nous devons ajouter ceci dans environnement () :
na, nb, alice_bob_nb: protocol_id

et dans goal :
secrecy_of nb

3.
Il faut ajouter dans le rôle d'Alice et Bob:
witness(A,B,bob_alice_na,Na')
request (B,A,bob_alice_na,Na)

puis ceci dans environnement () :
na, nb, alice_bob_nb: protocol_id

et dans goal :
secrecy_of nb

4.
Rôle de Bob :
request (B,A,bob_alice_nb,Nb)

dans environnement :
na, nb, alice_bob_nb, bob_alice_na, bob_alice_nb: protocol_id

puis dans goal :
authentication_on bob_alice_nb

5.
Na : text,

Avec l'option "--typed_model=yes" le protocol est safe alors qu'il ne l'est pas avec l'option --typed_model=no.

Exercice 4 :

1.
/avispa AS_RPC_1_W.hlpsl --ofmc
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/m/monnetpa/Documents/avispa-1.1//testsuite/results/AS_RPC_1_W.if
GOAL
as_specified
BACKEND
OFMC

Analyse et interprétation :
Initialement, le protocole est déjà sûr.

2.
Il faut simplement ajouter cette ligne dans goal :
authentication_on alice_bob_kpab

3.

Dans le rôle d’Alice :

```
A --> B : {Nap}_|Kpab|
                    State':= 6 /\ Nap’ := new() /\ Snd( {Nap’}_|Kpab|)
                                  /\  witness(A,B,nap ,Nap’)
                                  /\ request(A,B,alice_bob_kpab,Kpab')
                                  /\ secret(Nap',nap,{A,B})
```

```
B --> A : {Succ(Nap)}_|Kpab|
State = 6 /\ Rcv({Succ(Nap’)}_|Kpab|) =|>
          State’:=7
```

Dans le rôle de Bob :

```
A --> B : {Nap}_|Kpab|
State = 5 /\ Rcv({Nap’}_|Kpab|) =|>

B --> A : {Succ(Nap)}_|Kpab|

State’:=6 /\ Snd({Succ(Nap’)}_|Kpab|)
            /\ request(B, A, nap, Nap’)
```

et finalement :

```
secrecy_of nbp, kpab, nap
authentication_on alice_bob_nap
```

Exercice 5 :

```
%Alice - Bob
%
%   1. A -> B: G^|Na|, {N}_|Kb|
%   2. B -> A: G^|Nb|, {N}_|Ka|
%   3. A -> B: {Secret}_|(G^Na)^Nb|
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%HLPSL:

role alice (A, B: agent,
          Ka, Kb: public_key,
          SND, RCV: channel (dy),
```

```
        G : text)
played_by A def=

  local State : nat,
        N, Na, Nb, Scrt: text


  init State := 0

  transition

        0.  State  = 0 /\ RCV(start) =|>
    State':= 2 /\ Na' := new() /\ N' := new() /\ SND(exp(G,|Na'|).{N'}_|Kb|)


        2.  State  = 2 /\ RCV(exp(G,|Nb'|).{N'}_|Ka|) =|>
    State':= 4 /\ Scrt' = new() /\ SND({Scrt'}_|exp(exp((G,Na)),Nb)|)
                          /\ Secret(Scrt', scr, {A,B})



end role



role bob(A, B: agent,
        Ka, Kb: public_key,
        SND, RCV: channel (dy),
        G : text)
played_by B def=

  local State : nat,
    N, Na, Nb: text

  init State := 1

  transition

        1.  State  = 1 /\ RCV(exp(G,|Na'|).{N'}_|Kb|) =|>
    State':= 3 /\ Nb' := new() /\ N' := new() /\ SND(exp(G,|Nb'|).{N'}_|Ka|)


        3.  State  = 3 /\ RCV({Scrt'}_|exp(exp((G,Na)),Nb)|) =|>
    State':= 5

end role
```

```
role session(A, B: agent, Ka, Kb: public_key, G:text) def=

  local SA, RA, SB, RB: channel (dy)

  composition

    alice(A,B,Ka,Kb,SA,RA,G)
          /\ bob  (A,B,Ka,Kb,SB,RB,G)

end role



role environment() def=

          const a, b        : agent,
      ka, kb, ki   : public_key,
      scr : protocol_id,
      g : text

          intruder_knowledge = {a, b, ka, kb, ki, inv(ki), g}

          composition

    session(a,b,ka,kb,g)
          /\ session(a,i,ka,ki,g)
          /\ session(i,b,ki,kb,g)

end role



goal

  secrecy_of scr


end goal



environment()
```