TP Security Models

# Exercice 1

## Question 1

$ avispa NSPK.hlpsl --output=. --ofmc

% OFMC

% Version of 2006/02/13

SUMMARY

UNSAFE

DETAILS

ATTACK\_FOUND

PROTOCOL

./NSPK.if

GOAL

secrecy\_of\_nb

BACKEND

OFMC

COMMENTS

STATISTICS

parseTime: 0.00s

searchTime: 0.01s

visitedNodes: 10 nodes

depth: 2 plies

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)

% Reached State:

%

% secret(Nb(2),nb,set\_70)

% witness(b,a,alice\_bob\_nb,Nb(2))

% contains(a,set\_70)

% contains(b,set\_70)

% secret(Na(1),na,set\_74)

% witness(a,i,bob\_alice\_na,Na(1))

% contains(a,set\_74)

% contains(i,set\_74)

% state\_bob(b,i,ki,kb,1,dummy\_nonce,dummy\_nonce,set\_78,10)

% state\_alice(a,i,ka,ki,4,Na(1),Nb(2),set\_74,6)

% state\_bob(b,a,ka,kb,3,Na(1),Nb(2),set\_70,3)

% state\_alice(a,b,ka,kb,0,dummy\_nonce,dummy\_nonce,set\_62,3)

% request(a,i,alice\_bob\_nb,Nb(2),6)

$ avispa NSPK.hlpsl --output=. --cl-atse

SUMMARY

UNSAFE

DETAILS

ATTACK\_FOUND

TYPED\_MODEL

PROTOCOL

./NSPK.if

GOAL

Secrecy attack on (n5(Nb))

BACKEND

CL-AtSe

STATISTICS

Analysed : 9 states

Reachable : 8 states

Translation: 0.00 seconds

Computation: 0.00 seconds

ATTACK TRACE

i -> (a,6): start

(a,6) -> i: {n9(Na).a}\_ki

& Secret(n9(Na),set\_74); Add a to set\_74; Add i to set\_74;

i -> (a,3): start

(a,3) -> i: {n1(Na).a}\_kb

& Secret(n1(Na),set\_62); Witness(a,b,bob\_alice\_na,n1(Na));

& Add a to set\_62; Add b to set\_62;

i -> (b,4): {n9(Na).a}\_kb

(b,4) -> i: {n9(Na).n5(Nb)}\_ka

& Secret(n5(Nb),set\_70); Witness(b,a,alice\_bob\_nb,n5(Nb));

& Add a to set\_70; Add b to set\_70;

i -> (a,6): {n9(Na).n5(Nb)}\_ka

(a,6) -> i: {n5(Nb)}\_ki

$ avispa NSPK.hlpsl --output=. --satmc

SUMMARY

UNSAFE

DETAILS

ATTACK\_FOUND

BOUNDED\_NUMBER\_OF\_SESSIONS

BOUNDED\_SEARCH\_DEPTH

BOUNDED\_MESSAGE\_DEPTH

PROTOCOL

NSPK.if

GOAL

secrecy\_of\_nb(nb0(b,4),set\_70)

BACKEND

SATMC

COMMENTS

STATISTICS

attackFound true boolean

upperBoundReached false boolean

graphLeveledOff no boolean

satSolver sim solver

maxStepsNumber 30 steps

stepsNumber 5 steps

atomsNumber 379 atoms

clausesNumber 993 clauses

encodingTime 0.04 seconds

solvingTime 0.0 seconds

if2sateCompilationTime 0.01 seconds

ATTACK TRACE

i -> (a,6) : start

(a,6) -> i : {na0(a,6).a}\_ki

i -> (b,4) : {na0(a,6).a}\_kb

(b,4) -> i : {na0(a,6).nb0(b,4)}\_ka

i -> (a,6) : {na0(a,6).nb0(b,4)}\_ka

(a,6) -> i : {nb0(b,4)}\_ki

$ avispa NSPK.hlpsl --output=. --ta4sp

SUMMARY

INCONCLUSIVE

DETAILS

OVER\_APPROXIMATION

UNBOUNDED\_NUMBER\_OF\_SESSIONS

TYPED\_MODEL

PROTOCOL

./NSPK.if.ta4sp

GOAL

SECRECY - Property with identitier: nb

BACKEND

TA4SP

COMMENTS

Use an under-approximation in order to show a potential attack

The intruder might know some critical information

STATISTICS

Translation: 0.00 seconds

Computation 0.40 seconds

ATTACK TRACE

No Trace can be provided with the current version.

## Question 2

L’outils AVISPA analyse le protocole qu’on lui donne. Il est capable ensuite de nous donner différentes informations sur le protocole. Il nous indique si ce dernier est sécurisé ou non et s’il a trouvé une attaque. Il nous donne également le but du protocole ainsi que des statistiques des performance sur l’exécution.

## Question 3

Il suffit que Bob envoie (lors du SND) son Identité à Alice avec Na, Nb et qu'Alice s'attende à recevoir (lors du RCV) l'identité de Bob.

# Exercice 2

## a)

Attaque 1

Le Secret ni est mis en cause, c'est une attaque de type Man In The Middle.

Attaque 2

C'est une attaque de type Man In The Middle, Alice pense recevoir un message de Bob, mais en réalité de Eve, et chiffre donc sa réponse avec la clé publique d'Eve.

C'est le Secret nr qui est en cause.

Attaque 3

Même attaque que précédemment mais ici c'est le Nisynch qui est en cause.

Attaque 4

Même attaque que la première.

## b)

Une image contenant capture d’écran

Description générée automatiquement

# Exercice 3

# Exercice 4

## Question 1

$> proverif -in horn needham.horn  
Initial clauses:  
Clause 14: c:(v\_25,v\_26) -> c:v\_26  
Clause 13: c:(v\_20,v\_21) -> c:v\_20  
Clause 12: c:v\_18 & c:v\_19 -> c:(v\_18,v\_19)  
Clause 11: c:c[]  
Clause 10: c:pk(sA[])  
Clause 9: c:pk(sB[])  
Clause 8: c:x\_17 & c:encrypt(m\_16,pk(x\_17)) -> c:m\_16  
Clause 7: c:x\_15 -> c:pk(x\_15)  
Clause 6: c:x\_14 & c:y\_13 -> c:encrypt(x\_14,y\_13)  
Clause 5: c:pk(x\_12) -> c:encrypt((Na[pk(x\_12)],pk(sA[])),pk(x\_12))  
Clause 4: c:pk(x\_10) & c:encrypt((Na[pk(x\_10)],y\_11),pk(sA[])) -> c:encrypt((y\_11,k[pk(x\_10)]),pk(x\_10))  
Clause 3: c:encrypt((x\_8,y\_9),pk(sB[])) -> c:encrypt((x\_8,Nb[x\_8,y\_9]),y\_9)  
Clause 2: c:encrypt((x\_6,pk(sA[])),pk(sB[])) & c:encrypt((Nb[x\_6,pk(sA[])],z\_7),pk(sB[])) -> c:encrypt(secret[],pk(z\_7))  
Clause 1: c:new-name[!att = v\_4]  
Completing...  
goal reachable: c:secret[]  
Abbreviations:  
Na\_190 = Na[pk(x\_174)]  
Nb\_191 = Nb[Na\_190,pk(sA[])]  
k\_192 = k[pk(x\_174)]  
clause 8 c:secret[]  
    duplicate c:x\_188  
    clause 2 c:encrypt(secret[],pk(x\_188))  
        duplicate c:encrypt((Na\_190,pk(sA[])),pk(sB[]))  
        clause 6 c:encrypt((Nb\_191,x\_188),pk(sB[]))  
            apply 2-tuple c:(Nb\_191,x\_188)  
                apply 1-proj-2-tuple c:Nb\_191  
                    clause 8 c:(Nb\_191,k\_192)  
                        duplicate c:x\_174  
                        clause 4 c:encrypt((Nb\_191,k\_192),pk(x\_174))  
                            duplicate c:pk(x\_174)  
                            clause 3 c:encrypt((Na\_190,Nb\_191),pk(sA[]))  
                                clause 6 c:encrypt((Na\_190,pk(sA[])),pk(sB[]))  
                                    apply 2-tuple c:(Na\_190,pk(sA[]))  
                                        apply 1-proj-2-tuple c:Na\_190  
                                            clause 8 c:(Na\_190,pk(sA[]))  
                                                duplicate c:x\_174  
                                                clause 5 c:encrypt((Na\_190,pk(sA[])),pk(x\_174))  
                                                    clause 7 c:pk(x\_174)  
                                                        any c:x\_174  
                                        clause 10 c:pk(sA[])  
                                    duplicate c:pk(sB[])  
                any c:x\_188  
            clause 9 c:pk(sB[])  
  
RESULT goal reachable: c:secret[]

## Question 2

Proverif à atteint son but de trouver le secret. Pour cela il effectuer une attaque de type man in the middle, et a réussi à récupérer la clef pour déchiffrer le secret.

## Question 3

Pour éviter cette attaque, il faut que a et b signe leur envoies.

# Exercice 5