Supplementary Material

The HLPSL file for the asymmetric-key option EDHOC protocol

```
role initiator(
A,B: agent,
PK_I,PK_R: public_key,
G: text,
H: hash_func,
SND, RCV: channel(dy)
)played_by A def=

local
State: nat,
X,Ci,Cr,CRED_R,CRED_I: text,
Gx,Gy,Gxy : message

init
State:=0

transition
1.              State= 0    /\ RCV(start) =|>
 State':= 2  /\ Ci':= new()
                             /\ X':= new()
                             /\ Gx':= exp(G,X')
                             /\ SND(Gx'.Ci')
2.              State= 2
 /\ RCV(Ci.Cr'.Gy'.{{CRED_R'.H(Gx.Ci.Ci.Cr'.Gy')}_inv(PK_R)}_H(H(exp(Gy',X)).H(Gx.Ci.Ci.Cr'.Gy'))) =|>
 State':= 4  /\ Gxy':= exp(Gy',X)
              /\ CRED_I':= new()

 /\ SND(Cr'.{{CRED_I'.H(H(H(Gx.Ci.Ci.Cr'.Gy').{{CRED_R'.H(Gx.Ci.Ci.Cr'.Gy')}_inv(PK_R)}_H(H(Gxy').H(
Gx.Ci.Ci.Cr'.Gy'))).Cr')}_inv(PK_I)}_H(H(Gxy').H(H(H(Gx.Ci.Ci.Cr'.Gy').{{CRED_R'.H(Gx.Ci.Ci.Cr'.Gy')}_inv
(PK_R)}_H(H(Gxy').H(Gx.Ci.Ci.Cr'.Gy'))).Cr')))

 /\ witness(A,B,k3,H(H(Gxy').H(H(H(Gx.Ci.Ci.Cr'.Gy').{{CRED_R'.H(Gx.Ci.Ci.Cr'.Gy')}_inv(PK_R)}_H(H(Gx
y').H(Gx.Ci.Ci.Cr'.Gy'))).Cr')))
                             /\ request(A,B,k2,H(H(exp(Gy',X)).H(Gx.Ci.Ci.Cr'.Gy')))
end role

role responder(
A,B: agent,
PK_I,PK_R: public_key,
G: text,
H: hash_func,
SND, RCV: channel(dy)
)played_by B def=

local
State: nat,
Y,Ci,Cr,CRED_I,CRED_R: text,
Gx,Gy,Gxy: message

init
State:=1

transition
1.              State= 1    /\ RCV(Gx'.Ci') =|>
 State':= 3  /\ Cr':= new()
                             /\ Y':= new()
                             /\ Gy':= exp(G,Y')
```

```
                    /\ Gxy':= exp(Gx',Y')
                    /\ CRED_R':= new()

 /\ SND(Ci'.Cr'.Gy'.{{CRED_R'.H(Gx'.Ci'.Ci'.Cr'.Gy')}_inv(PK_R)}_H(H(Gxy').H(Gx'.Ci'.Ci'.Cr'.Gy')))
                    /\ witness(B,A,k2,H(H(Gxy').H(Gx'.Ci'.Ci'.Cr'.Gy')))

2.          State= 3
 /\ RCV(Cr.{{CRED_I'.H(H(H(Gx.Ci.Ci.Cr.Gy).{{CRED_R.H(Gx.Ci.Ci.Cr.Gy)}_inv(PK_R)}_H(H(Gxy).H(Gx.
Ci.Ci.Cr.Gy))).Cr)}_inv(PK_I)}_H(H(Gxy).H(H(H(Gx.Ci.Ci.Cr.Gy).{{CRED_R.H(Gx.Ci.Ci.Cr.Gy)}_inv(PK_R)
}_H(H(Gxy).H(Gx.Ci.Ci.Cr.Gy))).Cr))) =|>
 State':= 5
 /\ request(B,A,k3,H(H(Gxy).H(H(H(Gx.Ci.Ci.Cr.Gy).{{CRED_R.H(Gx.Ci.Ci.Cr.Gy)}_inv(PK_R)}_H(H(Gxy).
H(Gx.Ci.Ci.Cr.Gy))).Cr)))

end role

role session(
A,B: agent,
PK_I,PK_R: public_key,
G: text,
H: hash_func
)def=

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

composition
 initiator(A,B,PK_I,PK_R,G,H,SA,RA) /\
 responder(A,B,PK_I,PK_R,G,H,SB,RB)

end role

role environment()
def=

const
 a,b,i: agent,
 pk_a,pk_b,pk_i: public_key,
 g: text,
 h: hash_func,
 k2,k3: protocol_id

intruder_knowledge = {a,b,i,pk_a,pk_b,pk_i,g,h}

composition
 session(a,b,pk_a,pk_b,g,h) /\
 session(i,b,pk_i,pk_b,g,h) /\
 session(a,i,pk_a,pk_b,g,h)

end role

goal

authentication_on k2
authentication_on k3

end goal

environment()
```

The HLPSL file for the symmetric-key option EDHOC protocol

```
role initiator(
A,B: agent,
PSK: symmetric_key,
G: text,
H: hash_func,
SND, RCV: channel(dy)
)played_by A def=

local
State: nat,
X,Ci,Cr: text,
Gx,Gy,Gxy : message

init
State:=0

transition
1.              State= 0    /\ RCV(start) =|>
 State':= 2  /\ Ci':= new()
                            /\ X':= new()
                            /\ Gx':= exp(G,X')
                            /\ SND(Gx'.Ci')

2.              State= 2    /\ RCV(Ci.Cr'.Gy'.{H(Gx.Ci.Ci.Cr'.Gy')}_H(H(PSK.exp(Gy',X)).H(Gx.Ci.Ci.Cr'.Gy'))) =|>
 State':= 4  /\ Gxy':= exp(Gy',X)

 /\ SND(Cr'.{H(H(H(Gx.Ci.Ci.Cr'.Gy').{H(Gx.Ci.Ci.Cr'.Gy')}_H(H(PSK.Gxy').H(Gx.Ci.Ci.Cr'.Gy'))).Cr')}_H(H(
PSK.Gxy').H(H(H(Gx.Ci.Ci.Cr'.Gy').{H(Gx.Ci.Ci.Cr'.Gy')}_H(H(PSK.Gxy').H(Gx.Ci.Ci.Cr'.Gy'))).Cr')))

 /\ witness(A,B,k3,H(H(PSK.Gxy').H(H(H(Gx.Ci.Ci.Cr'.Gy').{H(Gx.Ci.Ci.Cr'.Gy')}_H(H(PSK.Gxy').H(Gx.Ci.Ci.
Cr'.Gy'))).Cr')))
                            /\ request(A,B,k2,H(H(PSK.exp(Gy',X)).H(Gx.Ci.Ci.Cr'.Gy')))
end role

role responder(
A,B: agent,
PSK: symmetric_key,
G: text,
H: hash_func,
SND, RCV: channel(dy)
)played_by B def=

local
State: nat,
Y,Ci,Cr: text,
Gx,Gy,Gxy: message

init
State:=1

transition
1.              State= 1    /\ RCV(Gx'.Ci') =|>
 State':= 3  /\ Cr':= new()
                            /\ Y':= new()
                            /\ Gy':= exp(G,Y')
                            /\ Gxy':= exp(Gx',Y')
                            /\ SND(Ci'.Cr'.Gy'.{H(Gx'.Ci'.Ci'.Cr'.Gy')}_H(H(PSK.Gxy').H(Gx'.Ci'.Ci'.Cr'.Gy')))
                            /\ witness(B,A,k2,H(H(PSK.Gxy').H(Gx'.Ci'.Ci'.Cr'.Gy')))

2.              State= 3
 /\ RCV(Cr.{H(H(H(Gx.Ci.Ci.Cr.Gy).{H(Gx.Ci.Ci.Cr.Gy)}_H(H(PSK.Gxy).H(Gx.Ci.Ci.Cr.Gy))).Cr)}_H(H(PSK
.Gxy).H(H(H(Gx.Ci.Ci.Cr.Gy).{H(Gx.Ci.Ci.Cr.Gy)}_H(H(PSK.Gxy).H(Gx.Ci.Ci.Cr.Gy))).Cr))) =|>
```

```
  State':= 5
  /\ request(B,A,k3,H(H(PSK.Gxy).H(H(H(Gx.Ci.Ci.Cr.Gy).{H(Gx.Ci.Ci.Cr.Gy)}_H(H(PSK.Gxy).H(Gx.Ci.Ci.Cr.
Gy))).Cr)))

end role

role session(
A,B: agent,
PSK: symmetric_key,
G: text,
H: hash_func
)def=

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

composition
  initiator(A,B,PSK,G,H,SA,RA) /\
  responder(A,B,PSK,G,H,SB,RB)

end role

role environment()
def=

const
  a,b,i: agent,
  psk: symmetric_key,
  g: text,
  h: hash_func,
  k2,k3: protocol_id

intruder_knowledge = {a,b,i,g,h}

composition
  session(a,b,psk,g,h) /\
  session(a,i,psk,g,h) /\
  session(i,b,psk,g,h)


end role

goal

authentication_on k2
authentication_on k3

end goal

environment()
```