Appendix A

HLPSL AUTHENTICATION CODE

```
role master (M, S           : agent,
             H, KeyGen      : hash_func,
             PRF            : hash_func,
             Kp, Kca        : public_key,
             %k1, k2, k3, m1, m2, m3: text,  %symmetric_key,
             SND_S, RCV_S   : channel (dy))
played_by M def=
  local  Np, Csus, PMS                  : text,
         SeID                           : text,
         Ns, TNo, Csu, Sh, Rcert        : text,
         Sc, Ske, Cke, Cv, Shd, Ccs     : text,
         State,Y1                       : nat,
          %K2, K3, M1, M2, M3:nat,
         Finished : hash(hash(text.text.text).agent.agent.text.text.text),
         ClientK, ServerK : hash(agent.text.text.hash(text.text.text)),
         Ks                             : public_key,
         Nps                            : text.text,
         X1:agent.text.text.agent
   const sec_clientK,
         sec_serverK,
         nps1, nps2 : protocol_id,
         sid0       : text,  % session id = 0
         request_id : text,
         respond_id : text,
```

```
      %k1,m1:text,
      start_tls  : text
   init State := 0   /\ Y1 := 123585
transition
0. State  = 0 /\ RCV_S(request_id) =|>
   State':= 2 /\ SND_S(respond_id.M)
    2. State  = 2 /\ RCV_S(start_tls) =|>
   State':= 4 /\ Np'   := new()
             /\ Csus' := new()
             /\ TNo'  := new()
             /\ SND_S( TNo'.sid0.Np'.Csus' )    % client hello (SeID=0)
             /\ SND_S( Y1')
% with client authentication
41. State = 4 /\ RCV_S(X1'.
             TNo.SeID'.Ns'.Csu'.         % server hello
             {S.Ks'}_inv(Kca).           % server certificate
             Ske'.                      % server key exchange
             Rcert'.                     % server certificate request
             Shd')                       % server hello done
   =|>
    State':= 6
     /\ PMS'      := new()
     /\ Ccs'      := new()
     /\ Y1'   := new()
    /\ Finished' := H(PRF(PMS'.Np.Ns').M.S.Np.Csu'.SeID')
     /\ ClientK'  := KeyGen(M.Np.Ns'.PRF(PMS'.Np.Ns'))
     /\ ServerK'  := KeyGen(S.Np.Ns'.PRF(PMS'.Np.Ns'))
```

```
      /\ SND_S({M.Kp}_inv(Kca).            % client certificate
            {PMS'}_Ks'.                 % client key exchange
            {H(Np.Ns'.S.PMS')}_inv(Kp). % client certificate verify
            Ccs'.X1'{Y1'}_ks'.                  % change cipher spec
            {Finished'}_ClientK')       % finished
      /\ witness(M,S,nps2,Np.Ns')
  6.state =6 /\ RCV_S(Y1) =|>
           State':= 8
          /\ SND_S(Y1)
       6. State  = 6 /\ RCV_S(Ccs.{Finished}_ServerK) =|>
    State':= 8 /\ secret(ClientK,sec_clientK,{M,S})
            /\ secret(ServerK,sec_serverK,{M,S})
            /\ request(M,S,nps1,Np.Ns)
end role
role server (M, S          : agent,
           H, KeyGen      : hash_func,
           PRF            : hash_func,
           Ks, Kca        : public_key,
          %k1,k2,k3,m1,m2,m3: text, %symmetric_key,
           SND_P, RCV_P   : channel (dy))
played_by S def=
  local  Ns, SeID                    : text,
         PMS                         : text,
         Np, Csus, TNo, Csu, Sh, Sc, Ske : text,
         Cke, Cv, Ccs, Rcert,Shd         : text,
         State ,Y1                   : nat,
          K1, K2, K3, M1, M2, M3: text,
```

```
      Finished : hash(hash(text.text.text).agent.agent.text.text.text),
      ClientK, ServerK : hash(agent.text.text.hash(text.text.text)),
      Kp                          : public_key

const nps1, nps2 : protocol_id,
     sid0       : text,  % session id = 0
     request_id : text,
     respond_id : text,
     start_tls  : text
init State := 1 /\  Y1 :=12325235
transition
1. State  = 1 /\ RCV_P(start) =|>
   State':= 3 /\ SND_P(request_id)
3. State  = 3 /\ RCV_P(respond_id.M) =|>
   State':= 5 /\ SND_P(start_tls)
% with client authentication
51. State  = 5 /\ RCV_P(TNo'.sid0.Np'.Csus')     % client hello
   =|>
    State':= 7
     /\ Ns'   := new()
     /\ SeID' := new()
     /\ Shd'  := new()
     /\ Rcert'  := new()
     /\ Ske'  := new()
     /\ Csu'  := new()
     /\ SND_P(TNo'.SeID'.Ns'.Csu'.         % server hello
            {S.Ks}_inv(Kca).             % server certificate
```

```
            Ske'.                    % server key exchange
            Rcert'.                  % server certificate request
           Shd')                     % server hello done
   /\ witness(S,M,nps1,Np'.Ns')


 % Server certificate
 7. State  = 7
    /\ RCV_P({M.Kp'}_inv(Kca).          % client certificate
           {PMS'}_Ks.                  % client key exchange
           {H(Np.Ns.S.PMS')}_inv(Kp').  % client certificate verify
           Ccs'.                       % change cipher spec
           {Finished'}_ClientK'        % finished
          )
    /\ Finished' = H(PRF(PMS'.Np.Ns).M.S.Np.Csu.SeID)
    /\ ClientK'  = KeyGen(M.Np.Ns.PRF(PMS'.Np.Ns))
   =|>
   State' := 11
    /\ ServerK'  := KeyGen(S.Np.Ns.PRF(PMS'.Np.Ns))
    %/\ SND_P(Ccs'.{Finished'}_ServerK')
    /\ request(S,M,nps2,Np.Ns)
     /\ SND_P(Y1)
%9.State  = 9 /\ SND_P(K1)
 %=|> State' := 11
 % without client authentication
 %9. State  = 9
    %/\ RCV_P({PMS'}_Ks.                % client key exchange
```

```
            %{H(Ns.S.PMS')}_inv(Kp).     % client certificate verify
          %     Ccs'.                     % change cipher spec
           %   {Finished'}_ClientK'        % finished
           %  )
       %/\ Finished' = H(PRF(PMS'.Np.Ns).M.S.Np.Csu.SeID)
      % /\ ClientK'  = KeyGen(M.Np.Ns.PRF(PMS'.Np.Ns))
    % =|>
     %State' := 11
      % /\ ServerK'  := KeyGen(S.Np.Ns.PRF(PMS'.Np.Ns))
      % /\ SND_P(Ccs'.{Finished'}_ServerK')
      %/\ request(S,M,nps2,Np.Ns)
11.State  = 11 /\ RCV_P(M1)
          =|> State' := 13
              /\ ServerK'  := KeyGen(S.Np.Ns.PRF(PMS'.Np.Ns))
              /\ SND_P(Ccs'.{Finished'}_ServerK')
               /\ request(S,M,nps2,Np.Ns)
end role
role session(M, S          : agent,
             Kp, Ks, Kca    : public_key,
             H, KeyGen      : hash_func,
             PRF            : hash_func)
def=
  local SP, SS, RP, RS : channel (dy),
        K1, K2, K3, M1, M2, M3: text,
      % init K1 :=1111
     NsSet : text set
         init NsSet := {}
```

```
  composition
          master(  M,S,H,KeyGen,PRF,Kp,Kca,SP,RP)
       /\ server(M,S,H,KeyGen,PRF,Ks,Kca,SS,RS)
end role
role environment()
def=
 const p,s             : agent,
        kp, ks, ki, kca : public_key,
        h, keygen       : hash_func,
        %k1,m1 :text,
        prf             : hash_func
  intruder_knowledge = {p,s, h,keygen,prf, kp,ks,kca,ki,inv(ki),
{i.ki}_inv(kca)        }
  composition
       session(p,s,kp,ks,kca,h,keygen,prf,NsSet)
%   /\  session(p,i,kp,ki,kca,h,keygen,prf)
   /\  session(i,s,ki,ks,kca,h,keygen,prf,NsSet)
end role
goal
       %secrecy_of ClientK, ServerK
       secrecy_of sec_clientK, sec_serverK
       %Master authenticates Server on nps1
       authentication_on nps1
       %Server authenticates Master on nps2
       authentication_on nps2
end goal
environment()
```