subprotocol for the establishment of child SAs

Protocol Purpose

IKE is designed to perform mutual authentication and key exchange prior to setting up an IPsec connection.

This subprotocol of IKE, known as CREATE_CHILD_SA, is used to establish child security associations once an initial SA has been set up using the two initial exchanges of IKEv2.

Definition Reference

[Kau03]

Model Authors

- Sebastian Mödersheim, ETH Zürich, December 2003
- Paul Hankes Drielsma, ETH Zürich, December 2003

Alice&Bob style

IKEv2-CHILD consists of a single exchange called CREATE_CHILD_SA. Given a previously set up security association with key K, the users exchange two messages encrypted with K. These messages exchanges nonces and perform a Diffie-Hellman exchange, establishing a new security association called. A (respectively B) generates a nonce Na and a Diffie-Hellman half key KEa (respectively KEb). In addition, SAa contains A's cryptosuite offers and SAb B's preference for the establishment of the new SA. Authentication is provided based on the use of K, which is assumed to be known only to A and B.

```
CREATE_CHILD_SA
1. A -> B: {SAa, Na, KEa}K
2. B -> A: {SAb, Nr, KEb}K
```

Note that because we abstract away from the negotiation of cryptographic algorithms, we have SAa = SAb.

Model Limitations

Issues abstracted from:

- The parties, Alice and Bob, should negotiate mutually acceptable cryptographic algorithms. This we abstract by modelling that Alice sends only a single offer for a crypto-suite, and Bob must accept this offer.
- There are goals of IKEv2 which we do not yet consider. For instance, identity hiding.
- We do not model the exchange of traffic selectors, which are specific to the IP network model and would be meaningless in our abstract communication model.

Problems considered: 3

Attacks Found

None.

HLPSL Specification

```
role alice(A,B:agent,
           G: text,
           F: function,
           SK: symmetric_key,
           SND_B, RCV_B: channel (dy))
played_by A
def=

  local Ni, SA, DHX: text,
        Nr: text,
        KEr: message, % more specifically: exp(text,text)
        CSK: message, % CHILD_SA to be established.
        State: nat,
        MA,MB: text

  const sec_a_CSK : protocol_id
```

```
  init  State := 0

  transition

  1. State = 0  /\ RCV_B(start) =|>
     State':= 2 /\ SA' := new()
                /\ Ni' := new()
                /\ DHX' := new()
                /\ SND_B( {SA'.Ni'.exp(G,DHX')}_SK )
                /\ witness(A,B,ni,Ni')

  2. State = 2  /\ RCV_B({SA.Nr'.KEr'}_SK) =|>
     State':= 4 /\ MA' := new()
                /\ CSK' := F(Ni.Nr'.SA.exp(KEr',DHX))
                /\ SND_B( {MA'.zero}_CSK' )

  4. State = 4  /\ RCV_B({MB'.one}_CSK) =|>
     State':= 6 /\ request(A,B,nr,Nr)
                /\ secret(CSK,sec_a_CSK,{A,B})

end role
```

```
role bob (B,A:agent,
          G: text,
          F: function,
          SK: symmetric_key,
          SND_A, RCV_A: channel (dy))
played_by B
def=

  local Ni, SA: text,
        Nr, DHY: text,
        KEi, CSK: message,
        State: nat,
        MA,MB: text

  const sec_b_CSK : protocol_id
```

```
  init  State := 1

  transition

  1. State = 1  /\ RCV_A( {SA'.Ni'.KEi'}_SK ) =|>
     State':= 3 /\ Nr' := new()
                /\ DHY' := new()
                /\ CSK' := F(Ni'.Nr'.SA'.exp(KEi',DHY'))
                /\ SND_A( {SA'.Nr'.exp(G,DHY')}_SK )
                /\ witness(B,A,nr,Nr')

  2. State = 3  /\ RCV_A( {MA'.zero}_CSK ) =|>
     State':= 5 /\ MB' := new()
                /\ SND_A( {MB'.one}_CSK )
                /\ request(B,A,ni,Ni)
                /\ secret(CSK,sec_b_CSK,{A,B})

end role
```

```
role session(A, B: agent,
             SK: symmetric_key,
             G: text, F: function)
def=

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

  composition
           alice(A,B,G,F,SK,SAC,RA)
        /\ bob(B,A,G,F,SK,SB,RB)
end role
```

```
role environment()
def=

  const ni,nr         : protocol_id,
        a, b          : agent,
        kab, kai, kbi : symmetric_key,
```

```
        g:text, f          : function,
        zero, one     : text

  intruder_knowledge = {g,f,a,b,i,kai,kbi,zero,one
                        }

  composition

        session(a,b,kab,g,f)
     /\ session(a,i,kai,g,f)
     /\ session(i,b,kbi,g,f)

end role
```

```
goal
        %secrecy_of CSK
        secrecy_of sec_a_CSK,sec_b_CSK

        %Alice authenticates Bob on nr
        authentication_on nr
        %Bob authenticates Alice on ni
        authentication_on ni
end goal
```

```
environment()
```

References

[Kau03] Charlie Kaufman. Internet Key Exchange (IKEv2) Protocol, October 2003. Work in Progress.