SPEKE (with strong password-only authentication)

Protocol Purpose

Strong Password-Only Authenticated Key Exchange

Definition Reference

http://citeseer.ist.psu.edu/jablon96strong.html

Model Authors

- Haykal Tej, Siemens CT IC 3, 2003
- Sebastian Mödersheim, ETH Zürich, December 2003

Alice&Bob style

```
A -> B : exp(S(A,B), Na)      |    key exchange part
B -> A : exp(S(A,B), Nb)      |

              both A and B compute
              K = exp(exp(S(A,B),Na), Nb) = exp(exp(S(A,B),Nb), Na)

A -> B : {Ca}_K               |
B -> A : {Cb,Ca}_K            |    challenge/response
A -> B : {Cb}_K               |    authentication part

S(A,B): password (shared key)
```

Model Limitations

None

Problems considered: 3

Attacks Found

None

Further Notes

None

HLPSL Specification

```
role speke_Init (A,B: agent,
                 Kab: symmetric_key,
                 Snd,Rcv: channel(dy))
played_by A
def=

  local  State: nat,
         Na,Ca: text,
         Cb   : text,
         X,K  : message

  const  sec_i_Ca, sec_i_Cb : protocol_id

  init   State := 0

  transition

   1. State  = 0 /\ Rcv(start) =|>
      State':= 1 /\ Na' := new()
                 /\ Snd(exp(Kab, Na'))

   2. State  = 1 /\ Rcv(X') =|>
      State':= 2 /\ Ca' := new()
                 /\ K' := exp(X',Na)
                 /\ Snd({Ca'}_exp(X',Na))
                 /\ secret(Ca',sec_i_Ca,{A,B})
                 /\ witness(A,B,ca,Ca')

   3. State  = 2 /\ Rcv({Cb'.Ca}_K ) =|>
      State':= 3 /\ Snd({Cb'}_K)
                 /\ secret(Cb',sec_i_Cb,{A,B})
                 /\ request(A,B,cb,Cb')

end role
```

```
role speke_Resp (A,B: agent,
                 Kab: symmetric_key,
                 Snd,Rcv: channel(dy))
played_by B
def=

  local State: nat,
        Nb,Cb: text,
        Ca   : text,
        Y,K  : message

  const sec_r_Ca, sec_r_Cb : protocol_id

  init  State := 0

  transition

   1. State  = 0 /\ Rcv(Y') =|>
      State':= 1 /\ Nb' := new()
                 /\ Snd(exp(Kab, Nb'))
                 /\ K' = exp(Y', Nb')

   2. State  = 1 /\ Rcv({Ca'}_K) =|>
      State':= 2 /\ Cb' := new()
                 /\ Snd({Cb'.Ca'}_K)
                 /\ secret(Ca',sec_r_Ca,{A,B})
                 /\ secret(Cb',sec_r_Cb,{A,B})
                 /\ witness(B,A,cb,Cb')
                 /\ request(B,A,ca,Ca')

   3. State  = 2 /\ Rcv({Cb}_K) =|>
```

```
      State':= 3

end role
```

```
role session (A,B: agent,
              Kab: symmetric_key)
def=

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

   composition

       speke_Init(A,B,Kab,SA,RA)
    /\ speke_Resp(A,B,Kab,SB,RB)

end role
```

```
role environment()
def=

  const a, b          : agent,
        kab, kai, kbi : symmetric_key,
        ca, cb        : protocol_id

  intruder_knowledge = {a, b, kai, kbi}

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

end role
```

```
goal

   %secrecy_of Ca, Cb
   secrecy_of sec_i_Ca,sec_i_Cb,
              sec_r_Ca,sec_r_Cb

   %SPEKE_Init authenticates SPEKE_Resp on cb
   authentication_on cb
   %SPEKE_Resp authenticates SPEKE_Init on ca
   authentication_on ca

end goal
```

```
environment()
```

References