EKE: Encrypted Key Exchange

basic

Protocol Purpose

Encrypted key exchange

Definition Reference

http://citeseer.ist.psu.edu/bellovin92encrypted.html

Model Authors

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

Alice&Bob style

```
A -> B : {Ea}_Kab          |    Key exchange part
B -> A : {{K}_Ea}_Kab      |
A -> B : {Ca}_K            |
B -> A : {Ca,Cb}_K         |    Challenge/Response
A -> B : {Cb}_K            |    Authentication part
```

Model Limitations

None

Problems considered: 3

Attacks Found

```
i -> (a,3): start
(a,3) -> i: {Ea(1)}_kab
```

```
i -> (a,6): {Ea(1)}_kab
(a,6) -> i: {{K(2)}_Ea(1)}_kab
i -> (a,3): {{K(2)}_Ea(1)}_kab
(a,3) -> i: {Na(3)}_K(2) witness(a,b,na,Na(3))
i -> (a,6): {Na(3)}_K(2)
(a,6) -> i: {Na(3),Nb(4)}_K(2) witness(a,b,nb,Nb(4))
i -> (a,3): {Na(3),Nb(4)}_K(2)
(a,3) -> i: {Nb(4)}_K(2)  request(a,b,nb,Nb(4))
```

Parallel session attack, man-in-the-middle between A as initiator and A as responder, attacker masquerades as B, but no secret nonces are exposed.

HLPSL Specification

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

  local State   : nat,
        Ea      : public_key,
        Na,Nb,K : text

  const sec_k1 : protocol_id

  init  State := 0

  transition

   1. State = 0
      /\ Rcv(start)
      =|>
      State' := 1
```

```
      /\ Ea' := new()
      /\ Snd({Ea'}_Kab)

   2. State = 1
      /\ Rcv({{K'}_Ea}_Kab)
      =|>
      State' := 2
      /\ Na' := new()
      /\ Snd({Na'}_K')
      /\ secret(K',sec_k1,{A,B})
      /\ witness(A,B,na,Na')

   3. State = 2
      /\ Rcv({Na.Nb'}_K)
      =|>
      State' := 3
      /\ Snd({Nb'}_K)
      /\ request(A,B,nb,Nb')

end role
```

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

  local State   : nat,
        Na,Nb,K : text,
        Ea      : public_key

  const sec_k2 : protocol_id

  init  State := 0

  transition

   1. State = 0 /\ Rcv({Ea'}_Kab)
      =|>
```

```
      State' := 1
      /\ K' := new()
      /\ Snd({{K'}_Ea'}_Kab)
      /\ secret(K',sec_k2,{A,B})

   2. State = 1 /\ Rcv({Na'}_K)
      =|>
      State' := 2
      /\ Nb' := new()
      /\ Snd({Na'.Nb'}_K)
      /\ witness(B,A,nb,Nb')

   3. State = 2
      /\ Rcv({Nb}_K)
      =|>
      State' := 3
      /\ request(B,A,na,Na)

end role
```

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

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

  composition
     eke_Init(A,B,Kab,SA,RA)
  /\ eke_Resp(B,A,Kab,SB,RB)

end role
```

```
role environment()
def=

  const a, b   : agent,
        kab    : symmetric_key,
```

```
        na, nb : protocol_id

  intruder_knowledge={a,b}

  composition
      session(a,b,kab)
   /\ session(b,a,kab)

end role
```

```
goal

 secrecy_of sec_k1, sec_k2
 %EKE_Init authenticates EKE_Resp on nb
 authentication_on nb
 %EKE_Resp authenticates EKE_Init on na
 authentication_on na

end goal
```

```
environment()
```

References