IKEv2: Internet Key Exchange, version 2

authentication based on digital signatures

Protocol Purpose

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

IKEv2 exists in several variants, the defining difference being the authentication method used.

This variant, which we call IKEv2-DS, uses digital signatures.

Definition Reference

[Kau03]

Model Authors

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

Alice&Bob style

IKEv2-DS proceeds in two so-called exchanges. In the first, called IKE_SA_INIT, the users exchange nonces and perform a Diffie-Hellman exchange, establishing an initial security association called the IKE_SA. The second exchange, IKE_SA_AUTH, then authenticates the previous messages, exchanges the user identities, and establishes the first so-called "child security association" or CHILD_SA which will be used to secure the subsequent IPsec tunnel. A (respectively B) generates a nonce Na and a Diffie-Hellman half key KEa (respectively KEb). In addition, SAa1 contains A's cryptosuite offers and SAb1 B's preference for the establishment of the IKE_SA. Similarly SAa2 and SAb2 for the establishment of the CHILD_SA.

```
IKE_SA_INIT
1. A -> B: SAa1, KEa, Na
2. B -> A: SAb1, KEb, Nb
IKE_SA_AUTH
3. A -> B: {A, AUTHa, SAa2}K
```

```
    where K = H(Na.Nb.SAa1.g^KEa^KEb) and
      AUTHa = {SAa1.g^KEa.Na.Nb}inv(Ka)
 4. B -> A: {B, AUTHb, SAb2}K
    where
      AUTHb = {SAb1.g^KEb.Na.Nb}inv(Kb)
```

Note that because we abstract away from the negotiation of cryptographic algorithms, we have SAa1 = SAb1 and SAa2 = SAb2.

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.
- IKEv2-DS includes provisions for the optional exchange of public-key certificates. This is not included in our model.
- 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

With this variant of IKEv2, we find an attack analogous to the one that Meadows reports on in [Mea99]. In essence, the intruder is able to mount a man-in-the-middle attack between agents *a* and *b*. The trace below illustrates how the intruder convinces *b* that he was talking with *a*, when in fact *a* has not participated in the same session. Rather, the intruder has merely relayed messages from a different session with *a*, a session in which *a* expects to talk to the intruder.

```
i -> (a,6): start
(a,6) -> i: SA1(1),exp(g,DHX(1)),Ni(1)
i -> (b,3): SA1(1),exp(g,DHX(1)),Ni(1)
(b,3) -> i: SA1(1),exp(g,DHY(2)),Nr(2)
i -> (a,6): SA1(1),exp(g,DHY(2)),Nr(2)
(a,6) -> i: {a,{SA1(1),exp(g,DHX(1)),Ni(1),Nr(2)}inv(ka),
                SA2(3)}(f(Ni(1),Nr(2),SA1(1),exp(exp(g,DHY(2)),DHX(1))))
i -> (b,3): {a,{SA1(1),exp(g,DHX(1)),Ni(1),Nr(2)}inv(ka),
                SA2(3)}(f(Ni(1),Nr(2),SA1(1),exp(exp(g,DHX(1)),DHY(2))))
(b,3) -> i: {b,{SA1(1),exp(g,DHY(2)),Nr(2),Ni(1)}inv(kb),
                SA2(3)}(f(Ni(1),Nr(2),SA1(1),exp(exp(g,DHX(1)),DHY(2))))
```

This attack is of questionable validity, as the intruder has not actually learned the key that b believes to have established with a. Thus, the intruder cannot exploit the authentication flaw to further purposes. The attack can be precluded if we add key confirmation to the protocol. That is, if we extend the protocol to include messages in which the exchanged key is actually used, then this attack is no longer possible. In specification IKEv2-DSX we do just this.

HLPSL Specification

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

  local Ni, SA1, SA2, DHX: text,
        Nr: text,
        KEr: message, %% more specific: exp(text,text)
        SK: message,
        State: nat

  const sec_a_SK : protocol_id

  init  State := 0

  transition

  %% The IKE_SA_INIT exchange:
  %% We have abstracted away from the negotiation of cryptographic
  %% parameters.  Alice sends a nonce SAi1, which is meant to
  %% model Alice sending only a single crypto-suite offer.  Bob must
  %% then respond with the same nonce.
  1. State = 0  /\ RCV_B(start) =|>
     State':= 2 /\ SA1' := new()
                /\ DHX' := new()
                /\ Ni' := new()
                /\ SND_B( SA1'.exp(G,DHX').Ni' )

  %% Alice receives message 2 of IKE_SA_INIT, checks that Bob has
  %% indeed sent the same nonce in SAr1, and then sends the first
  %% message of IKE_AUTH.
  %% As authentication Data, she signs her first message and Bob's nonce.
  2. State = 2  /\ RCV_B(SA1.KEr'.Nr') =|>
     State':= 4 /\ SA2' := new()
                /\ SK' := F(Ni.Nr'.SA1.exp(KEr',DHX))
                /\ SND_B( {A.{SA1.exp(G,DHX).Ni.Nr'}_(inv(Ka)).SA2'}_SK' )
                /\ witness(A,B,sk2,F(Ni.Nr'.SA1.exp(KEr',DHX)))

  3. State = 4  /\ RCV_B({B.{SA1.KEr.Nr.Ni}_(inv(Kb)).SA2}_SK) =|>
     State':= 9 /\ secret(SK,sec_a_SK,{A,B})
                /\ request(A,B,sk1,SK)

end role
```

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

  local Ni, SA1, SA2: text,
        Nr, DHY: text,
        SK, KEi: message,
        State: nat

  const sec_b_SK : protocol_id
  init  State := 1

  transition

  1. State = 1  /\ RCV_A( SA1'.KEi'.Ni' ) =|>
     State':= 3 /\ DHY' := new()
                /\ Nr' := new()
                /\ SND_A(SA1'.exp(G,DHY').Nr')
                /\ SK' := F(Ni'.Nr'.SA1'.exp(KEi',DHY'))
                /\ witness(B,A,sk1,F(Ni'.Nr'.SA1'.exp(KEi',DHY')))

  2. State = 3  /\ RCV_A( {A.{SA1.KEi.Ni.Nr}_(inv(Ka)).SA2'}_SK ) =|>
     State':= 9 /\ SND_A( {B.{SA1.exp(G,DHY).Nr.Ni}_(inv(Kb)).SA2'}_SK )
                /\ secret(SK,sec_b_SK,{A,B})
                /\ request(B,A,sk2,SK)

end role
```

```
role session(A, B: agent,
             Ka, Kb: public_key,
             G: text, F: function)
def=

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

  composition
           alice(A,B,G,F,Ka,Kb,SA,RA)
        /\ bob(B,A,G,F,Kb,Ka,SB,RB)

end role
```

```
role environment()
def=

  const sk1,sk2    : protocol_id,
        a, b       : agent,
        ka, kb, ki : public_key,
        g:text, f       : function

  intruder_knowledge = {g,f,a,b,ka,kb,i,ki,inv(ki)
                       }

  composition

        session(a,b,ka,kb,g,f)
     /\ session(a,i,ka,ki,g,f)
     /\ session(i,b,ki,kb,g,f)

end role
```

```
goal

  %secrecy_of SK
  secrecy_of sec_a_SK, sec_b_SK

  %Alice authenticates Bob on sk1
  authentication_on sk1
  %Bob authenticates Alice on sk2
  authentication_on sk2

end goal
```

```
environment()
```

References

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

[Mea99] Catherine Meadows. Analysis of the Internet Key Exchange Protocol Using the NRL

Protocol Analyzer. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1999.