Abbreviations
~M_3 = sci_A_3
~M_4 = enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))
~M_5 = hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10))),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))
~M = sci_A_3

A trace has been found.

Honest Process

Attacker

{1}new szk_3

{2}new sci_A_3

{3}new msg_3

{6}insert freshness(sci_A_3,0,0)

{9}new sak_2

{7}let kek: key = kdf(szk_3,10)

{10}let msg_1: bitstring = key_enc(sak_2,kdf(szk_3,10))

{8}let ick: key = kdf(szk_3,20)

{4}let an: nat = 1

{11}let icv: bitstring = hmac(key_enc(sak_2,kdf(szk_3,10)),1,1,kdf(szk_3,20))

{5}let fresh: nat = 1

(~M,1,1,~M_1,~M_2) = (sci_A_3,1,1,key_enc(sak_2,kdf(szk_3,10)),hmac(key_enc(sak_2,kdf(szk_3,10)),1,1,kdf(szk_3,20)))

Phase 1

!

!

Beginning of process Alice

{17}let an_1: nat = 1

{15}let sci_A_1: bitstring = sci_A_3

Beginning of process Bob

Beginning of process Bob

Beginning of process Bob

Beginning of process Bob

Beginning of process Bob

(~M,1,1,~M_1,~M_2) = (sci_A_3,1,1,key_enc(sak_2,kdf(szk_3,10)),hmac(key_enc(sak_2,kdf(szk_3,10)),1,1,kdf(szk_3,20)))

{25}let sci_A_2: bitstring = sci_A_3

{39}get association_keys(=sci_A_3,=1,ska_A: key): else branch taken

{26}let szk_2: key = szk_3

{33}let ick_1: key = kdf(szk_3,20)

{34}if ((sci_A_3 = sci_A_3) && (hmac(key_enc(sak_2,kdf(szk_3,10)),1,1,kdf(szk_3,20)) = hmac(key_enc(sak_2,kdf(szk_3,10)),1,1,kdf(szk_3,20))))

{37}insert freshness(sci_A_3,1,1)

{35}let kek_1: key = kdf(szk_3,10)

{36}let sak_1: key = key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))

{38}insert association_keys(sci_A_3,1,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{23}get association_keys(sci_A_3,1,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{18}let packet_number: nat = 2

{21}event message_send_Alice(key_enc(sak_2,kdf(szk_3,10)),2)

{19}let msg_2: bitstring = enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{20}let icv2: bitstring = hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

(~M_3,1,2,~M_4,~M_5)

(~M,1,2,~M_4,~M_5)

{25}let sci_A_2: bitstring = sci_A_3

{39}get association_keys(sci_A_3,1,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{32}get freshness(=sci_A_3,=1,fv: nat) suchthat (fv ≥ 2): else branch taken

{28}if ((sci_A_3 = sci_A_3) && (hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))) = hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))))

(~M,1,2,~M_4,~M_5)

{25}let sci_A_2: bitstring = sci_A_3

{39}get association_keys(sci_A_3,1,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{32}get freshness(=sci_A_3,=1,fv: nat) suchthat (fv ≥ 2): else branch taken

{28}if ((sci_A_3 = sci_A_3) && (hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))) = hmac(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),1,2,key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))))

{30}insert freshness(sci_A_3,1,2)

{29}let received_msg: bitstring = dec(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10)))

{31}event message_received_Bob(dec(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),2)

{31}event message_received_Bob(dec(enc(key_enc(sak_2,kdf(szk_3,10)),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),key_dec(key_enc(sak_2,kdf(szk_3,10)),kdf(szk_3,10))),2)