Abbreviations $\sim X_1 = (\sim M_1, 1, a, enc(a_1, \sim M), hmac(enc(a_1, \sim M), 1, a, \sim M))$

(sci_A_3,1,a,enc(a_1,kdf(szk_3,1)),hmac(enc(

a_1,kdf(szk_3,1)),1,a,kdf(szk_3,1)))

Honest Process Attacker $\{1\}$ new szk_3 {2}new sci_A_3 ${3}$ new msg_2 {4}insert freshness(sci_A_3,0,0) {5}insert key_counter(sci_A_3,0,szk_3) Beginning of process Alice Beginning of process Alice $\{10\}$ let kc: nat = 1 $\{10\}$ let kc: nat = 1 Beginning of process Bob Beginning of process Bob $\{9\}$ let szk_1: key = szk_3 $\{9\}$ let szk_1: key = szk_3 $\{11\}$ let sak_alice: key = kdf(szk_3,1) $\{11\}$ let sak_alice: key = kdf(szk_3,1) \sim M = kdf(szk 3,1) {13}new randomness_2 {14} let icv: bitstring = hmac(randomness_2,1,1, kdf(szk_3,1)) $\{8\}$ let sci_A_1: bitstring = sci_A_3 $(\sim M_1, 1, 1, \sim M_2, \sim M_3) = (sci_A_3, 1, 1, enc(randomness_2, 1, 1, enc(randomness_2, 1, 1, enc(randomness_2, 1, 1, enc(randomness_3, 1, 1, 1, enc(rando$ kdf(szk_3,1)),hmac(randomness_2,1,1,kdf(szk_3, $\{16\}$ let packet_number: nat = 2 $\{7\}$ let m: bitstring = msg_2 {19}event message_send_Alice(msg_2,2): blocks \sim M_4 = kdf(szk_3,1) {13}new randomness_3 {14} let icv: bitstring = hmac(randomness_3,1,1, kdf(szk_3,1)) {8}let sci_A_1: bitstring = sci_A_3 $(\sim M_5, 1, 1, \sim M_6, \sim M_7) = (sci_A_3, 1, 1, enc(randomness_3, 1, 1, 1, enc(rando$ kdf(szk_3,1)),hmac(randomness_3,1,1,kdf(szk_3, 1))) $\{16\}$ let packet_number: nat = 2 $\{7\}$ let m: bitstring = msg_2 {19}event message_send_Alice(msg_2,2): blocks $(\sim M_1, 1, 1, a_2, hmac(a_2, 1, 1, \sim M)) = (sci_A_3, 1, 1, \dots)$ a_2,hmac(a_2,1,1,kdf(szk_3,1))) {22} let sci_A_2: bitstring = sci_A_3 {34} get key_counter(=sci_A_3,=1,ska_A: key): else branch taken $\{23\}$ let szk_2: key = szk_3 $\{30\}$ let n_key: key = kdf(szk_3,1) ${31}$ if $((sci_A_3 = sci_A_3) && ((1 = 1) && (hmac(a_2,1,1,kdf(szk_3,1)) = hmac($ a $2,1,1,kdf(szk_3,1))))$ {32} insert freshness(sci_A_3,1,1) {33}insert key_counter(sci_A_3,1,kdf(szk_3,1)) ~X 1 $\{22\}$ let sci A 2: bitstring = sci A 3 {34} get key_counter(sci_A_3,1,kdf(szk_3,1)) $\{29\}$ get freshness(=sci_A_3,=1,fv: nat) suchthat (fv \geq a): else branch taken {25} if ((sci_A_3 = sci_A_3) && (hmac(enc(a_1,kdf(szk_3,1)),1,a,kdf(szk_3,1)) = hmac(enc(a_1,kdf(szk_3,1)),1,a,kdf(szk_3,1))) {27} insert freshness(sci_A_3,1,a) {26} let received_msg: bitstring = dec(enc(a_1, kdf(szk_3,1)),kdf(szk_3,1)) {28} event message_received_Bob(dec(enc(a_1,kdf(szk_3,1)),kdf(szk_3,1)),a)