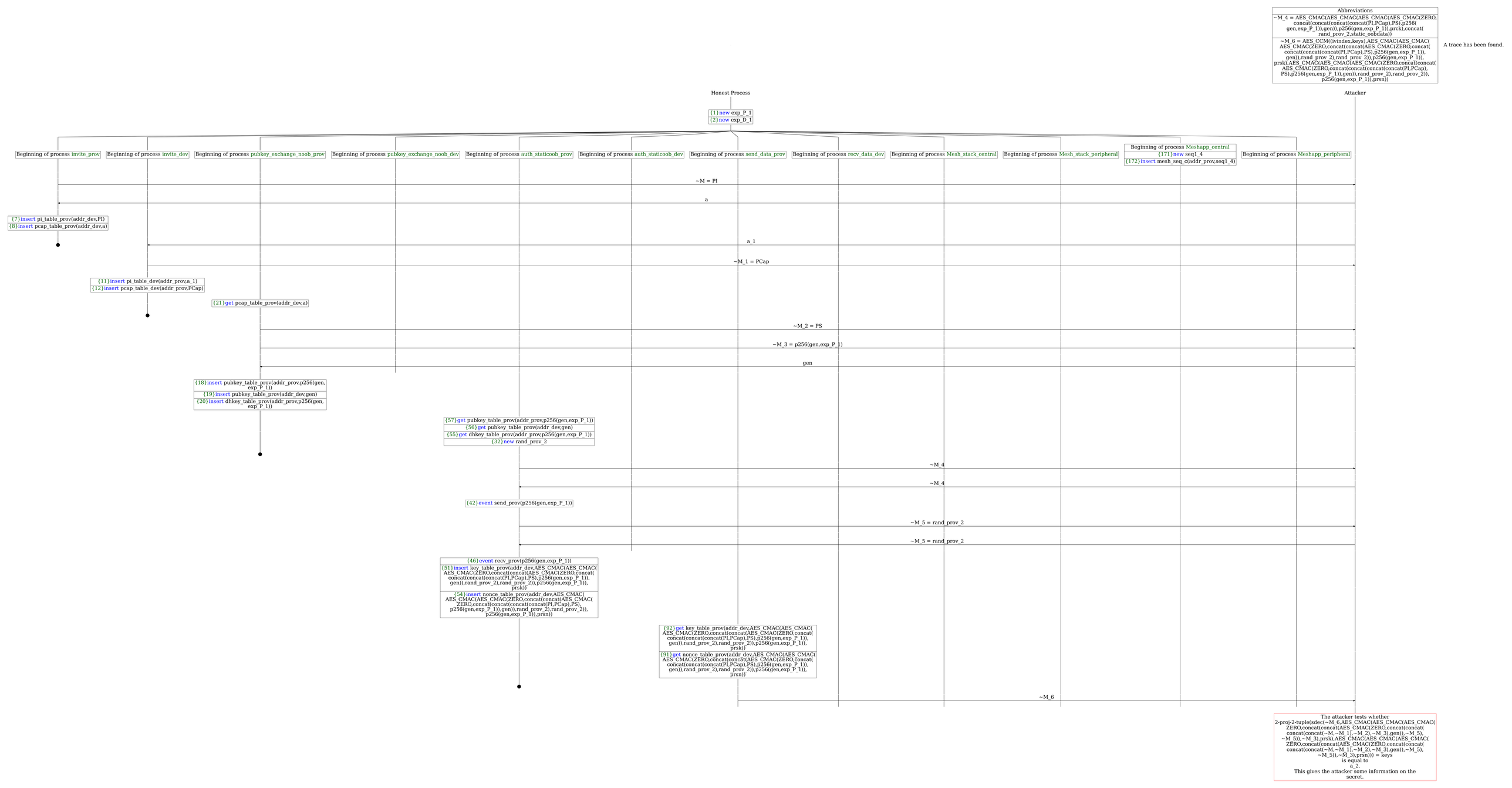
Abbreviations
~M_4 = AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),p256(gen,exp_P_1)),prck),concat(rand_prov_2,static_oobdata))
~M_6 = AES_CCM((ivindex,keys),AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsk),AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsn))
A trace has been found.
Honest Process
Attacker
{1}new exp_P_1
{2}new exp_D_1
Beginning of process invite_prov
Beginning of process invite_dev
Beginning of process pubkey_exchange_noob_prov
Beginning of process pubkey_exchange_noob_dev
Beginning of process auth_staticoob_prov
Beginning of process auth_staticoob_dev
Beginning of process send_data_prov
Beginning of process recv_data_dev
Beginning of process Mesh_stack_central
Beginning of process Mesh_stack_peripheral
Beginning of process Meshapp_central
{171}new seq1_4
{172}insert mesh_seq_c(addr_prov,seq1_4)
Beginning of process Meshapp_peripheral
~M = PI
a
{7}insert pi_table_prov(addr_dev,PI)
{8}insert pcap_table_prov(addr_dev,a)
a_1
~M_1 = PCap
{11}insert pi_table_dev(addr_prov,a_1)
{12}insert pcap_table_dev(addr_prov,PCap)
{21}get pcap_table_prov(addr_dev,a)
~M_2 = PS
~M_3 = p256(gen,exp_P_1)
gen
{18}insert pubkey_table_prov(addr_prov,p256(gen,exp_P_1))
{19}insert pubkey_table_prov(addr_dev,gen)
{20}insert dhkey_table_prov(addr_prov,p256(gen,exp_P_1))
{57}get pubkey_table_prov(addr_prov,p256(gen,exp_P_1))
{56}get pubkey_table_prov(addr_dev,gen)
{55}get dhkey_table_prov(addr_prov,p256(gen,exp_P_1))
{32}new rand_prov_2
~M_4
~M_4
{42}event send_prov(p256(gen,exp_P_1))
~M_5 = rand_prov_2
~M_5 = rand_prov_2
{46}event recv_prov(p256(gen,exp_P_1))
{51}insert key_table_prov(addr_dev,AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsk))
{54}insert nonce_table_prov(addr_dev,AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsn))
{92}get key_table_prov(addr_dev,AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsk))
{91}get nonce_table_prov(addr_dev,AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(PI,PCap),PS),p256(gen,exp_P_1)),gen)),rand_prov_2),rand_prov_2)),p256(gen,exp_P_1)),prsn))
~M_6
The attacker tests whether
2-proj-2-tuple(sdec(~M_6,AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(~M,~M_1),~M_2),~M_3),gen)),~M_5),~M_5)),~M_3),prsk),AES_CMAC(AES_CMAC(AES_CMAC(ZERO,concat(concat(AES_CMAC(ZERO,concat(concat(concat(concat(~M,~M_1),~M_2),~M_3),gen)),~M_5),~M_5)),~M_3),prsn))) = keys
is equal to
a_2.
This gives the attacker some information on the secret.