Abbreviations $\sim X_1 = (derive_encryption_key(Curve25519(Curve25519(gen, encryption_key)))$ derive_prikey_from_sn(serial_number)),pri_c_1), rand_3),Curve25519(gen,pri_c_1)) \sim M_3 = AES_enc(smartthings,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),bleAuthentication), ~X_2 = AES_enc(serial_number,derive_key(derive_encryption_key(Curve25519(\sim M_1,derive_prikey_from_sn(serial_number)), \sim M), \sim M_2), \sim M_2) = AES enc(serial number, derive key(derive_encryption_key(Curve25519(Curve25519(gen, pri_c_1),derive_prikey_from_sn(serial_number)), rand_3),n1_2),n1_2) ~M_4 = AES_enc(privacy_iv_3,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),n1_2),n1_2) ~M_5 = AES_enc(privacy_seed_3,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),n1_2),n1_2) ~M_6 = AES_enc(poolsize,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),n1_2),n1_2) \sim M_7 = AES_enc(time_sync_3,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),n1_2),n1_2) ~X_3 = AES_enc(smartthings,derive_key(derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number)), $\sim \overline{M}$), ble Authentication), $\sim M$ 9) = AES enc(smartthings, derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)), rand_3),bleAuthentication),n2_2) \sim M_10 = AES_enc(smartthings,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),bleAuthentication),a) ~M 11 = AES enc(serial_number,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_ $\overline{3}$),n2_2),n $\overline{2}$ _2) $\sim X_4 = (AES_enc(AES_dec(\sim M_4, derive_key(derive_encryption_key($ Curve $25\overline{5}19$ ($\sim M_1$, derive prikey from sn(serial number)), \sim M), \sim M_2), \sim M_2),derive_key(derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number)), \sim M), \sim M_9), \sim M_9),AES_enc(AES_dec(\sim M_5,derive_key() derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number), $\sim M$, $\sim M_2$, $\sim M_2$, derive_key(derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number)), ~M),~M_9),~M_9),AES_enc(poolsize,derive_key(derive_encryption_key($\overline{\text{Curve25519}}$ (~M 1, derive prikey from sn(serial number)), \sim M), \sim M_9), \sim M_9),AES_enc(a_1,derive_key(derive_encryption_key(Curve25519(\sim M_1,derive_prikey_from_sn(serial_number)), \sim M), \sim M_9), \sim M_9)) = (AES_enc(privacy_iv_3,derive_key(derive_encryption_key(Curve25519(Curve25519(gen, pri_c_1),derive_prikey_from_sn(serial_number)), rand_3),n2_2),n2_2),AES_enc(privacy_seed_3,derive_key(derive_encryption_key(Curve25519(Curve25519(gen, pri_c_1),derive_prikey_from_sn(serial_number)), rand_3),n2_2),n2_2),AES_enc(poolsize,derive_key(derive_encryption_key(Curve25519(Curve25519(gen, pri_c_1),derive_prikey_from_sn(serial_number)), rand_3),n2_2),n2_2),AES_enc(a_1,derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),n2_2),n2_2)) ~X_5 = get_id(derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3, privacy_seed_3,poolsize) $\sim X_6 = \text{get_id(derive_key(derive_encryption_key(Curve25519()))}$ Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3, privacy_seed_3,poolsize) \sim M_12 = tmp_2 ~M_13 = AES_enc(s,get_id(derive_key(derive_encryption_key(

A trace has been found.

Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3,

privacy_seed_3,poolsize),tmp_2) **Honest Process** Attacker Beginning of process step0s Beginning of process step1s Beginning of process step1p Beginning of process step2p Beginning of process step0p Beginning of process step1c Beginning of process step2c Beginning of process user (serial_number,derive_hashed_sn(serial_number), Curve25519(gen,derive_prikey_from_sn(serial_number))) {8}insert p0p(serial_number,derive_hashed_sn(serial_number), derive_prikey_from_sn(serial_number),Curve25519(gen,derive_prikey_from_sn(serial_number))) {2}insert p0s(serial_number,derive_hashed_sn(serial_number), Curve25519(gen,derive_prikey_from_sn(serial_number))) {27} get p0s(serial_number,derive_hashed_sn(serial_number), Curve25519(gen,derive_prikey_from_sn(serial_number))) derive_hashed_sn(serial_number) {29} new rand_3 (derive_hashed_sn(serial_number),rand_3) {11}new pri_c_1 ~X_1 $(\sim M, \sim M_1) = (rand_3, Curve25519(gen, pri_c_1))$ {34} new n1_2 $\sim M_2 = n1_2$ $\sim M_2 = n1_2$ \sim M₃ \sim M_3 $\sim X_2$ (serial_number,derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)), pri_c_1),rand_3)) {20} new privacy_iv_3 {21} new privacy_seed_3 {23} new time_sync_3 (privacy_iv_3,privacy_seed_3,poolsize,time_sync_3) $(\sim M_4, \sim M_5, \sim M_6, \sim M_7)$ {54}insert p1c(addrB,get_id(derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1),rand_3),privacy),privacy_iv_3,privacy_seed_3,poolsize)) [78] get p0p(serial_number,derive_hashed_sn(serial_number), derive_prikey_from_sn(serial_number),Curve25519(gen,derive_prikey_from_sn(serial_number))) \sim M_8 = derive_hashed_sn(serial_number) $(\sim M, \sim M_1) = (rand_3, Curve25519(gen, pri_c_1))$ {60} new n2_2 $\sim M_9 = n2_2$ ~X 3 ~M 10 ~M 11 ~X 4 {77}insert p1p(addrA,get_id(derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3,privacy_seed_3,poolsize)) {84}get p1c(addrB,get_id(derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3,privacy_seed_3,poolsize)) {89}get p1p(addrA,get_id(derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),privacy),privacy_iv_3,privacy_seed_3,poolsize)) $\sim X_6$ yes confirm {81} new tmp_2 $(\sim M_12, \sim M_13)$