Abbreviations ~X_1 = (derive_encryption_key(Curve25519(Curve25519(gen, derive_prikey_from_sn(serial_number)),pri_c_1), rand_3),Curve25519(gen,pri_c_1)) ~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), \sim 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) ~X_3 = (AES_dec(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),derive_key(derive_encryption_key(Curve25519(Curve25519(gen,derive_prikey_from_sn(serial_number)),pri_c_1), rand_3),n1_2),n1_2),derive_encryption_key(Curve25519(Curve25519(Gen,derive_prikey_from_sn(serial_number)), pri_c_1),rand_3)) ~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) ~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) derive_key(derive_encryption_key(Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),
rand_3),bleAuthentication),n2_2) ~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) \sim M_11 = AES_enc(serial_number,derive_key(derive_encryption_key(A trace has been found. Curve25519(Curve25519(gen,pri_c_1),derive_prikey_from_sn(serial_number)),rand_3),n2_2),n2_2) derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number)),~M),~M_2),~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(Curve25519(~M_1,derive_prikey_from_sn(serial_number)),~M_9),~M_9),~M_9),a_1) ~X_6 = 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) ~X_7 = 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) ~M_12 = SHA256(concat(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),a_2)) ~X_8 = SHA256(concat(get_id(derive_key(derive_encryption_key(Curve25519(~M_1,derive_prikey_from_sn(serial_number)), 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),salt2_2))

Honest Process Attacker Beginning of process user Beginning of process step3c Beginning of process step0p Beginning of process step1c Beginning of process step2c Beginning of process step2p Beginning of process step3p Beginning of process step1s Beginning of process step1p Beginning of process step0s (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))) derive_hashed_sn(serial_number) {29}new rand_3 (derive_hashed_sn(serial_number),rand_3) $(\sim M, \sim M_1) = (rand_3, Curve25519(gen, pri_c_1))$ $\sim M_2 = n1_2$ $\sim M_2 = n1_2$ ~M_3 ~M_3 ~X_2 {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) $(\sim M, \sim M_1) = (rand_3, Curve25519(gen, pri_c_1))$ $\sim M_9 = n2_2$