$\sim$ M\_7 = uidx\_1  $\sim$ M\_8 = pk(skSoC\_2)  $\sim$ M\_9 = pk(skMan\_2)  $\sim$ M = pk(skAlice)  $\sim$ M\_1 = pk(skMan\_2)  $\sim$ M\_10 = pk(skMan\_2)  $\sim$ M 11 = salt 1  $\sim$  M\_12 = H(id\_to\_bitstring(salt\_1))  $\sim$  M\_13 = sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring( A trace has been found. salt\_1))),skAlice)  $\sim X_1 = (\sim M_1, \sim M_1, H(id_to_bitstring(\sim M_1)), \sim M_1)$ (pk(skMan\_2),salt\_1,H(id\_to\_bitstring(salt\_1)),
sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring( salt\_1))),skAlice))  $\sim$ M\_16 = pk(skSoC\_2) ~M\_17 = sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1))),skAlice),skMan\_2)  $\sim$ M 18 = pk(skAlice)  $\sim$  M\_19 = H(id\_to\_bitstring(salt\_1))  $\sim X_2 = (\sim M_8, \sim M_15, \sim M, H(id_{to_bitstring}(\sim M_11)))$ = (pk( skSoC\_2),sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1))),skAlice),skMan\_2), pk(skAlice),H(id\_to\_bitstring(salt\_1))) **Honest Process** Attacker  $\{1\}$  let pkAlice: pkey = pk(skAlice)  $\sim$ M = pk(skAlice) {3}new skMan\_2  $\{4\}$  let pkMan: pkey = pk(skMan\_2)  $\sim$ M\_1 = pk(skMan\_2) Beginning of process processMan Beginning of process processMan {8}new uidx\_2 {8}new uidx\_1 {9}new skSoC\_3 {9}new skSoC\_2 Beginning of process processAlice Beginning of process processSoC(pk(skMan\_2)) Beginning of process processSoC(pk(skMan\_2)) {11}let BuidN: bitstring = id\_to\_bitstring(uidN) {11} let BuidN: bitstring = id\_to\_bitstring(uidN) {12}let proofUidx: bitstring = H(id\_to\_bitstring(uidN)) {12}let proofUidx: bitstring = H(id\_to\_bitstring(uidN))  $\{10\}$  let pkSoc: pkey = pk(skSoC\_3)  $\{10\}$  let pkSoc: pkey = pk(skSoC\_2) (skSoC\_3,pk(skSoC\_3),uidx\_2,H(id\_to\_bitstring(uidN))) {13}let Buidx: bitstring = id\_to\_bitstring(uidx\_2)  $\{60\}$  if  $(pk(skSoC_3) = pk(skSoC_3))$ {14}let batchRoot: bitstring = ccatH(H(id\_to\_bitstring(uidx\_2)),H(id\_to\_bitstring(uidN))) {61}let signedUidx: bitstring = sign(id\_to\_bitstring(uidx\_2),skSoC\_3)  $\{7\}$  let skMan\_1: skey = skMan\_2  $\sim$  M\_2 = sign(id\_to\_bitstring(uidx\_2),skSoC\_3) (skSoC\_2,pk(skSoC\_2),uidx\_1,H(id\_to\_bitstring(uidN))) {13}let Buidx: bitstring = id\_to\_bitstring(uidx\_1)  $\{60\}$  if  $(pk(skSoC_2) = pk(skSoC_2))$ {14}let batchRoot: bitstring = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))) {61}let signedUidx: bitstring = sign(id\_to\_bitstring(uidx\_1),skSoC\_2)  $\{7\}$ let skMan\_1: skey = skMan\_2 (pk(skAlice),sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(uidN))),skMan\_2))  $\sim$ M\_3 = sign(id\_to\_bitstring(uidx\_1),skSoC\_2)  $\{31\}$  if (pk(skAlice) = pk(skAlice)) $(\sim M_4, \sim M_5) \sim M_6, \sim M_7, \sim M_8, \sim M_9)$  $(\sim M, \sim M_5, \sim M_6, \sim M_7, \sim M_8, \sim M_1)$  ${33}$  if (pk(skAlice) = pk(skAlice)){34} let Broot: bitstring = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))) {35}if (checksign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))),skMan\_2),pk(skMan\_2)) = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN)))) {36} if (sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(uidN))), skMan\_2) = sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(uidN))), skMan\_2))  $\sim$ M\_3 = sign(id\_to\_bitstring(uidx\_1),skSqC\_2) {38} if (checksign(sign(id\_to\_bitstring(uidx\_1), skSoC\_2),pk(skSoC\_2)) = id\_to\_bitstring(uidx\_1)) {39}if (ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))) = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))  $\{40\}$  new salt\_1  $\{41\}$  let Hsalt: bitstring = H(id\_to\_bitstring(salt\_1)) {42}let newBRoot: bitstring = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1)))  $\{29\}$  let skAlice\_1: skey = skAlice {43}let newBatchRootProof: bitstring = sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),
skAlice)  $(\sim M_10, \sim M_11, \sim M_12, \sim M_13)$  $\sim X$  $\{19\}$  if  $(pk(skMan_2) = pk(skMan_2))$ {20}let Bsalt: bitstring = id\_to\_bitstring(salt\_1) {21} if (H(id\_to\_bitstring(salt\_1)) = H(id\_to\_bitstring(salt\_1))) {22} if (checksign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice), pk(skAlice)) = ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1)))) {23}let newOwnershipProof: bitstring = sign(sign(catH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice),skMan\_2) {24} event ManSendCertif(pk(skMan\_2),sign(sign(catH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice),skMan\_2))  $(\sim M_14, \sim M_15) = |(pk(skAlice), sign(sign(ccatH(H($ id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))), skAlice),skMan\_2))  $(\sim M, \sim M_15) = (pk(skAlice), sign(sign(ccatH(H(id to_bitstring($ {26} event ManSendBCCertif(pk(skMan\_2),pk(skAlice)) uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice), skMan\_2))  $\{46\}$  if (pk(skAlice) = pk(skAlice))(pk(skAlice),sign(sign(ccatH(H(id\_to\_bitstring( uidx\_1)),H(id\_to\_bitstring(salt\_1)),skAlice), skMan 2))  ${48}$  if (pk(skAlice) = pk(skAlice)){49} if (sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1))), skAlice), skMan\_2) = sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1))), skAlice), skMan\_2)) {50} if (checksign(sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice), skMan\_2),pk(skMan\_2)) = sign(ccatH(H(id\_to\_bitstring(  $\overline{u}idx_1)$ , H(id\_ $\overline{t}o_bitstring(salt_1))$ , skAlice)) {51} event AliceRecBRProof(pk(skMan\_2),pk(skAlice))  $(\sim M_16, \sim M_17, \sim M_18, \sim M_19)$ {53}new configScript\_1 {54} let config: bitstring = ccat(id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1)) {55}let signedConfig: bitstring = sign(ccat(id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1)),skAlice) {56} event AliceSendModif(pk(skSoC\_2), sign(ccat( id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1)),
skAlice))  $(\sim M 20, \sim M 21) = (pk(skSoC 2), sign(ccat(id to bitstring))$ uidx 1),cfg to bitstring(configScript 1)),skAlice))  $\sim X_2$  $\{64\}$  if  $(pk(skSoC_2) = pk(skSoC_2))$ {65}let certifPartB: bitstring = sign(ccatH(H(id\_to\_bitstring(salt\_1))), H(id\_to\_bitstring(salt\_1))), skAlice) {66} if (checksign(sign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice), skMan\_2),pk(skMan\_2)) = sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice)) {67} let newBatchRootSoC: bitstring = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))) {68} if (checksign(sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice), pk(skAlice)) = ccatH(H(id\_to\_bitstring(uidx\_1)), H(id\_to\_bitstring(salt\_1)))) {69}if (ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))) = ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1)))) {70} event socAcceptCertif(pk(skAlice),sign(sign(catH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(salt\_1))),skAlice),skMan\_2))  $|(\sim M_8, \sim M_21)| = (pk(skSoC_2), sign(ccat(id_to_bitstring))|$ uidx\_1),cfg\_to\_bitstring(configScript\_1)),skAlice))  $\{72\}$  if  $(pk(skSoC_2) = pk(skSoC_2))$ {73}let dataConf: bitstring = ccat(id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1)) {74}if (checksign(sign(ccat(id\_to\_bitstring(uidx\_1), cfg\_to\_bitstring(configScript\_1)),skAlice),pk(skAlice)) = ccat(id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1))) {75}let confId: bitstring = id\_to\_bitstring(uidx\_1)  $\{76\}$  if  $(id_{to}bitstring(uidx_1) = id_{to}bitstring(uidx_1)$ 

{77} event socAcceptModif(pk(skAlice),sign(ccat(id\_to\_bitstring(uidx\_1),cfg\_to\_bitstring(configScript\_1)), skAlice))

Abbreviations

 $\sim$ M\_4 = pk(skAlice)

~M\_5 = sign(ccatH(H(id\_to\_bitstring(uidx\_1)),H(id\_to\_bitstring(uidN))),skMan\_2)

 $\sim$  M\_6 = H(id\_to\_bitstring(uidN))