Contents

1 ConductPBType Theory

Built: 10 June 2018
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

plCommandPB = securePB | withdrawPB | completePB
        | plIncompletePB

psgCommandPB = actionsInPB | psgIncompletePB

slCommand = PL plCommandPB | PSG psgCommandPB

slOutput = ConductPB | SecurePB | ActionsInPB | WithdrawPB
      | CompletePB | unAuthenticated | unAuthorized

slState = CONDUCT_PB | SECURE_PB | ACTIONS_IN_PB | WITHDRAW_PB
      | COMPLETE_PB

stateRole = PlatoonLeader | PlatoonSergeant

1.2 Theorems

[plCommandPB_distinct_clauses]
 ⊢ securePB $\neq$ withdrawPB $\land$ securePB $\neq$ completePB $\land$
   securePB $\neq$ plIncompletePB $\land$ withdrawPB $\neq$ completePB $\land$
   withdrawPB $\neq$ plIncompletePB $\land$ completePB $\neq$ plIncompletePB

[psgCommandPB_distinct_clauses]
 ⊢ actionsInPB $\neq$ psgIncompletePB

[slCommand_distinct_clauses]
 ⊢ $\forall a'\ a.$ PL $a \neq$ PSG a'

[slCommand_one_one]
 ⊢ $(\forall a\ a'.$ (PL a = PL a') $\iff$ $(a = a'))$ $\land$
   $\forall a\ a'.$ (PSG a = PSG a') $\iff$ $(a = a')$

[slOutput_distinct_clauses]
⊢ ConductPB $\neq$ SecurePB $\land$ ConductPB $\neq$ ActionsInPB $\land$
   ConductPB $\neq$ WithdrawPB $\land$ ConductPB $\neq$ CompletePB $\land$
   ConductPB $\neq$ unAuthenticated $\land$ ConductPB $\neq$ unAuthorized $\land$
   SecurePB $\neq$ ActionsInPB $\land$ SecurePB $\neq$ WithdrawPB $\land$
   SecurePB $\neq$ CompletePB $\land$ SecurePB $\neq$ unAuthenticated $\land$
   SecurePB $\neq$ unAuthorized $\land$ ActionsInPB $\neq$ WithdrawPB $\land$
   ActionsInPB $\neq$ CompletePB $\land$ ActionsInPB $\neq$ unAuthenticated $\land$
   ActionsInPB $\neq$ unAuthorized $\land$ WithdrawPB $\neq$ CompletePB $\land$
   WithdrawPB $\neq$ unAuthenticated $\land$ WithdrawPB $\neq$ unAuthorized $\land$
   CompletePB $\neq$ unAuthenticated $\land$ CompletePB $\neq$ unAuthorized $\land$
   unAuthenticated $\neq$ unAuthorized

[slRole_distinct_clauses]

⊢ PlatoonLeader ≠ PlatoonSergeant

[slState_distinct_clauses]

⊢ CONDUCT_PB ≠ SECURE_PB ∧ CONDUCT_PB ≠ ACTIONS_IN_PB ∧
  CONDUCT_PB ≠ WITHDRAW_PB ∧ CONDUCT_PB ≠ COMPLETE_PB ∧
  SECURE_PB ≠ ACTIONS_IN_PB ∧ SECURE_PB ≠ WITHDRAW_PB ∧
  SECURE_PB ≠ COMPLETE_PB ∧ ACTIONS_IN_PB ≠ WITHDRAW_PB ∧
  ACTIONS_IN_PB ≠ COMPLETE_PB ∧ WITHDRAW_PB ≠ COMPLETE_PB

2 ssmConductPB Theory

Built: 10 June 2018

Parent Theories: ConductPBType, ssm11, OMNIType

2.1 Definitions

[secContextConductPB_def]

⊢ ∀ *plcmd psgcmd incomplete* .
   secContextConductPB *plcmd psgcmd incomplete* =
   [Name PlatoonLeader controls prop (SOME (SLc (PL *plcmd*)));
    Name PlatoonSergeant controls
    prop (SOME (SLc (PSG *psgcmd*)));
    Name PlatoonLeader says
    prop (SOME (SLc (PSG *psgcmd*))) impf prop NONE;
    Name PlatoonSergeant says
    prop (SOME (SLc (PL *plcmd*))) impf prop NONE]

[ssmConductPBStateInterp_def]

⊢ ∀ *slState* . ssmConductPBStateInterp *slState* = TT

2.2 Theorems

[authTestConductPB_cmd_reject_lemma]

⊢ ∀ *cmd*. ¬authTestConductPB (prop (SOME *cmd*))

[authTestConductPB_def]

⊢ (authTestConductPB (Name PlatoonLeader says prop *cmd*) ⟺ T) ∧
  (authTestConductPB (Name PlatoonSergeant says prop *cmd*) ⟺
   T) ∧ (authTestConductPB TT ⟺ F) ∧
  (authTestConductPB FF ⟺ F) ∧
  (authTestConductPB (prop *v*) ⟺ F) ∧
  (authTestConductPB (notf v_1) ⟺ F) ∧
  (authTestConductPB (v_2 andf v_3) ⟺ F) ∧
  (authTestConductPB (v_4 orf v_5) ⟺ F) ∧
  (authTestConductPB (v_6 impf v_7) ⟺ F) ∧

(authTestConductPB (v_8 eqf v_9) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says TT) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says FF) $\iff$ F) $\wedge$
(authTestConductPB ($v133$ meet $v134$ says prop v_{66}) $\iff$ F) $\wedge$
(authTestConductPB ($v135$ quoting $v136$ says prop v_{66}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says notf v_{67}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says (v_{68} andf v_{69})) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says (v_{70} orf v_{71})) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says (v_{72} impf v_{73})) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says (v_{74} eqf v_{75})) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{76} says v_{77}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{78} speaks_for v_{79}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{80} controls v_{81}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says reps v_{82} v_{83} v_{84}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{85} domi v_{86}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{87} eqi v_{88}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{89} doms v_{90}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{91} eqs v_{92}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{93} eqn v_{94}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{95} lte v_{96}) $\iff$ F) $\wedge$
(authTestConductPB (v_{10} says v_{97} lt v_{98}) $\iff$ F) $\wedge$
(authTestConductPB (v_{12} speaks_for v_{13}) $\iff$ F) $\wedge$
(authTestConductPB (v_{14} controls v_{15}) $\iff$ F) $\wedge$
(authTestConductPB (reps v_{16} v_{17} v_{18}) $\iff$ F) $\wedge$
(authTestConductPB (v_{19} domi v_{20}) $\iff$ F) $\wedge$
(authTestConductPB (v_{21} eqi v_{22}) $\iff$ F) $\wedge$
(authTestConductPB (v_{23} doms v_{24}) $\iff$ F) $\wedge$
(authTestConductPB (v_{25} eqs v_{26}) $\iff$ F) $\wedge$
(authTestConductPB (v_{27} eqn v_{28}) $\iff$ F) $\wedge$
(authTestConductPB (v_{29} lte v_{30}) $\iff$ F) $\wedge$
(authTestConductPB (v_{31} lt v_{32}) $\iff$ F)

[authTestConductPB_ind]

$\vdash \forall P.$
$(\forall cmd.\ P$ (Name PlatoonLeader says prop cmd)) $\wedge$
$(\forall cmd.\ P$ (Name PlatoonSergeant says prop cmd)) $\wedge$ P TT $\wedge$
P FF $\wedge$ $(\forall v.\ P$ (prop v)) $\wedge$ $(\forall v_1.\ P$ (notf v_1)) $\wedge$
$(\forall v_2\ v_3.\ P\ (v_2$ andf v_3)) $\wedge$ $(\forall v_4\ v_5.\ P\ (v_4$ orf v_5)) $\wedge$
$(\forall v_6\ v_7.\ P\ (v_6$ impf v_7)) $\wedge$ $(\forall v_8\ v_9.\ P\ (v_8$ eqf v_9)) $\wedge$
$(\forall v_{10}.\ P\ (v_{10}$ says TT)) $\wedge$ $(\forall v_{10}.\ P\ (v_{10}$ says FF)) $\wedge$
$(\forall v133\ v134\ v_{66}.\ P\ (v133$ meet $v134$ says prop v_{66})) $\wedge$
$(\forall v135\ v136\ v_{66}.\ P\ (v135$ quoting $v136$ says prop v_{66})) $\wedge$
$(\forall v_{10}\ v_{67}.\ P\ (v_{10}$ says notf v_{67})) $\wedge$
$(\forall v_{10}\ v_{68}\ v_{69}.\ P\ (v_{10}$ says (v_{68} andf v_{69}))) $\wedge$
$(\forall v_{10}\ v_{70}\ v_{71}.\ P\ (v_{10}$ says (v_{70} orf v_{71}))) $\wedge$
$(\forall v_{10}\ v_{72}\ v_{73}.\ P\ (v_{10}$ says (v_{72} impf v_{73}))) $\wedge$
$(\forall v_{10}\ v_{74}\ v_{75}.\ P\ (v_{10}$ says (v_{74} eqf v_{75}))) $\wedge$
$(\forall v_{10}\ v_{76}\ v_{77}.\ P\ (v_{10}$ says v_{76} says v_{77})) $\wedge$
$(\forall v_{10}\ v_{78}\ v_{79}.\ P\ (v_{10}$ says v_{78} speaks_for v_{79})) $\wedge$

$(\forall\, v_{10}\ v_{80}\ v_{81}.\ P\ (v_{10}\ \text{says}\ v_{80}\ \text{controls}\ v_{81}))\ \wedge$
$(\forall\, v_{10}\ v_{82}\ v_{83}\ v_{84}.\ P\ (v_{10}\ \text{says}\ \text{reps}\ v_{82}\ v_{83}\ v_{84}))\ \wedge$
$(\forall\, v_{10}\ v_{85}\ v_{86}.\ P\ (v_{10}\ \text{says}\ v_{85}\ \text{domi}\ v_{86}))\ \wedge$
$(\forall\, v_{10}\ v_{87}\ v_{88}.\ P\ (v_{10}\ \text{says}\ v_{87}\ \text{eqi}\ v_{88}))\ \wedge$
$(\forall\, v_{10}\ v_{89}\ v_{90}.\ P\ (v_{10}\ \text{says}\ v_{89}\ \text{doms}\ v_{90}))\ \wedge$
$(\forall\, v_{10}\ v_{91}\ v_{92}.\ P\ (v_{10}\ \text{says}\ v_{91}\ \text{eqs}\ v_{92}))\ \wedge$
$(\forall\, v_{10}\ v_{93}\ v_{94}.\ P\ (v_{10}\ \text{says}\ v_{93}\ \text{eqn}\ v_{94}))\ \wedge$
$(\forall\, v_{10}\ v_{95}\ v_{96}.\ P\ (v_{10}\ \text{says}\ v_{95}\ \text{lte}\ v_{96}))\ \wedge$
$(\forall\, v_{10}\ v_{97}\ v_{98}.\ P\ (v_{10}\ \text{says}\ v_{97}\ \text{lt}\ v_{98}))\ \wedge$
$(\forall\, v_{12}\ v_{13}.\ P\ (v_{12}\ \text{speaks\_for}\ v_{13}))\ \wedge$
$(\forall\, v_{14}\ v_{15}.\ P\ (v_{14}\ \text{controls}\ v_{15}))\ \wedge$
$(\forall\, v_{16}\ v_{17}\ v_{18}.\ P\ (\text{reps}\ v_{16}\ v_{17}\ v_{18}))\ \wedge$
$(\forall\, v_{19}\ v_{20}.\ P\ (v_{19}\ \text{domi}\ v_{20}))\ \wedge$
$(\forall\, v_{21}\ v_{22}.\ P\ (v_{21}\ \text{eqi}\ v_{22}))\ \wedge$
$(\forall\, v_{23}\ v_{24}.\ P\ (v_{23}\ \text{doms}\ v_{24}))\ \wedge$
$(\forall\, v_{25}\ v_{26}.\ P\ (v_{25}\ \text{eqs}\ v_{26}))\ \wedge\ (\forall\, v_{27}\ v_{28}.\ P\ (v_{27}\ \text{eqn}\ v_{28}))\ \wedge$
$(\forall\, v_{29}\ v_{30}.\ P\ (v_{29}\ \text{lte}\ v_{30}))\ \wedge\ (\forall\, v_{31}\ v_{32}.\ P\ (v_{31}\ \text{lt}\ v_{32}))\ \Rightarrow$
$\forall\, v.\ P\ v$

[conductPBNS_def]

$\vdash$ (conductPBNS CONDUCT_PB (exec (PL securePB)) = SECURE_PB) $\wedge$
(conductPBNS CONDUCT_PB (exec (PL plIncompletePB)) =
 CONDUCT_PB) $\wedge$
(conductPBNS SECURE_PB (exec (PSG actionsInPB)) =
 ACTIONS_IN_PB) $\wedge$
(conductPBNS SECURE_PB (exec (PSG psgIncompletePB)) =
 SECURE_PB) $\wedge$
(conductPBNS ACTIONS_IN_PB (exec (PL withdrawPB)) =
 WITHDRAW_PB) $\wedge$
(conductPBNS ACTIONS_IN_PB (exec (PL plIncompletePB)) =
 ACTIONS_IN_PB) $\wedge$
(conductPBNS WITHDRAW_PB (exec (PL completePB)) =
 COMPLETE_PB) $\wedge$
(conductPBNS WITHDRAW_PB (exec (PL plIncompletePB)) =
 WITHDRAW_PB) $\wedge$ (conductPBNS s (trap (PL cmd')) = s) $\wedge$
(conductPBNS s (trap (PSG cmd)) = s) $\wedge$
(conductPBNS s (discard (PL cmd')) = s) $\wedge$
(conductPBNS s (discard (PSG cmd)) = s)

[conductPBNS_ind]

$\vdash \forall P.$
    P CONDUCT_PB (exec (PL securePB)) $\wedge$
    P CONDUCT_PB (exec (PL plIncompletePB)) $\wedge$
    P SECURE_PB (exec (PSG actionsInPB)) $\wedge$
    P SECURE_PB (exec (PSG psgIncompletePB)) $\wedge$
    P ACTIONS_IN_PB (exec (PL withdrawPB)) $\wedge$
    P ACTIONS_IN_PB (exec (PL plIncompletePB)) $\wedge$
    P WITHDRAW_PB (exec (PL completePB)) $\wedge$
    P WITHDRAW_PB (exec (PL plIncompletePB)) $\wedge$

$(\forall s\ cmd.\ P\ s$ (trap (PL $cmd))) \wedge$
$(\forall s\ cmd.\ P\ s$ (trap (PSG $cmd))) \wedge$
$(\forall s\ cmd.\ P\ s$ (discard (PL $cmd))) \wedge$
$(\forall s\ cmd.\ P\ s$ (discard (PSG $cmd))) \wedge$
P CONDUCT_PB (exec (PL withdrawPB)) $\wedge$
P CONDUCT_PB (exec (PL completePB)) $\wedge$
$(\forall v_{11}.\ P$ CONDUCT_PB (exec (PSG $v_{11}))) \wedge$
$(\forall v_{13}.\ P$ SECURE_PB (exec (PL $v_{13}))) \wedge$
P ACTIONS_IN_PB (exec (PL securePB)) $\wedge$
P ACTIONS_IN_PB (exec (PL completePB)) $\wedge$
$(\forall v_{17}.\ P$ ACTIONS_IN_PB (exec (PSG $v_{17}))) \wedge$
P WITHDRAW_PB (exec (PL securePB)) $\wedge$
P WITHDRAW_PB (exec (PL withdrawPB)) $\wedge$
$(\forall v_{20}.\ P$ WITHDRAW_PB (exec (PSG $v_{20}))) \wedge$
$(\forall v_{21}.\ P$ COMPLETE_PB (exec $v_{21})) \Rightarrow$
$\forall v\ v_1.\ P\ v\ v_1$

[conductPBOut_def]

$\vdash$ (conductPBOut CONDUCT_PB (exec (PL securePB)) = ConductPB) $\wedge$
  (conductPBOut CONDUCT_PB (exec (PL plIncompletePB)) =
  ConductPB) $\wedge$
  (conductPBOut SECURE_PB (exec (PSG actionsInPB)) =
  SecurePB) $\wedge$
  (conductPBOut SECURE_PB (exec (PSG psgIncompletePB)) =
  SecurePB) $\wedge$
  (conductPBOut ACTIONS_IN_PB (exec (PL withdrawPB)) =
  ActionsInPB) $\wedge$
  (conductPBOut ACTIONS_IN_PB (exec (PL plIncompletePB)) =
  ActionsInPB) $\wedge$
  (conductPBOut WITHDRAW_PB (exec (PL completePB)) =
  WithdrawPB) $\wedge$
  (conductPBOut WITHDRAW_PB (exec (PL plIncompletePB)) =
  WithdrawPB) $\wedge$
  (conductPBOut s (trap (PL cmd')) = unAuthorized) $\wedge$
  (conductPBOut s (trap (PSG cmd)) = unAuthorized) $\wedge$
  (conductPBOut s (discard (PL cmd')) = unAuthenticated) $\wedge$
  (conductPBOut s (discard (PSG cmd)) = unAuthenticated)

[conductPBOut_ind]

$\vdash \forall P.$
  P CONDUCT_PB (exec (PL securePB)) $\wedge$
  P CONDUCT_PB (exec (PL plIncompletePB)) $\wedge$
  P SECURE_PB (exec (PSG actionsInPB)) $\wedge$
  P SECURE_PB (exec (PSG psgIncompletePB)) $\wedge$
  P ACTIONS_IN_PB (exec (PL withdrawPB)) $\wedge$
  P ACTIONS_IN_PB (exec (PL plIncompletePB)) $\wedge$
  P WITHDRAW_PB (exec (PL completePB)) $\wedge$
  P WITHDRAW_PB (exec (PL plIncompletePB)) $\wedge$
  $(\forall s\ cmd.\ P\ s$ (trap (PL $cmd))) \wedge$

```
(∀ s cmd. P s (trap (PSG cmd))) ∧
(∀ s cmd. P s (discard (PL cmd))) ∧
(∀ s cmd. P s (discard (PSG cmd))) ∧
P CONDUCT_PB (exec (PL withdrawPB)) ∧
P CONDUCT_PB (exec (PL completePB)) ∧
(∀ v₁₁. P CONDUCT_PB (exec (PSG v₁₁))) ∧
(∀ v₁₃. P SECURE_PB (exec (PL v₁₃))) ∧
P ACTIONS_IN_PB (exec (PL securePB)) ∧
P ACTIONS_IN_PB (exec (PL completePB)) ∧
(∀ v₁₇. P ACTIONS_IN_PB (exec (PSG v₁₇))) ∧
P WITHDRAW_PB (exec (PL securePB)) ∧
P WITHDRAW_PB (exec (PL withdrawPB)) ∧
(∀ v₂₀. P WITHDRAW_PB (exec (PSG v₂₀))) ∧
(∀ v₂₁. P COMPLETE_PB (exec v₂₁)) ⇒
∀ v  v₁. P  v  v₁
```

[PlatoonLeader_exec_plCommandPB_justified_thm]

```
⊢ ∀ NS  Out  M  Oi  Os.
    TR (M, Oi, Os) (exec (SLc (PL plCommand)))
      (CFG authTestConductPB ssmConductPBStateInterp
        (secContextConductPB plCommand psgCommand incomplete)
        (Name PlatoonLeader says
         prop (SOME (SLc (PL plCommand)))::ins) s outs)
      (CFG authTestConductPB ssmConductPBStateInterp
        (secContextConductPB plCommand psgCommand incomplete)
        ins (NS s (exec (SLc (PL plCommand))))
        (Out s (exec (SLc (PL plCommand)))::outs)) ⟺
    authTestConductPB
      (Name PlatoonLeader says
       prop (SOME (SLc (PL plCommand)))) ∧
    CFGInterpret (M, Oi, Os)
      (CFG authTestConductPB ssmConductPBStateInterp
        (secContextConductPB plCommand psgCommand incomplete)
        (Name PlatoonLeader says
         prop (SOME (SLc (PL plCommand)))::ins) s outs) ∧
    (M, Oi, Os) sat prop (SOME (SLc (PL plCommand)))
```

[PlatoonLeader_plCommandPB_lemma]

```
⊢ CFGInterpret (M, Oi, Os)
    (CFG authTestConductPB ssmConductPBStateInterp
      (secContextConductPB plCommand psgCommand incomplete)
      (Name PlatoonLeader says
       prop (SOME (SLc (PL plCommand)))::ins) s outs) ⇒
  (M, Oi, Os) sat prop (SOME (SLc (PL plCommand)))
```

[PlatoonSergeant_exec_psgCommandPB_justified_thm]

```
⊢ ∀ NS  Out  M  Oi  Os.
    TR (M, Oi, Os) (exec (SLc (PSG psgCommand)))
```

```
   (CFG authTestConductPB ssmConductPBStateInterp
      (secContextConductPB plCommand psgCommand incomplete)
      (Name PlatoonSergeant says
       prop (SOME (SLc (PSG psgCommand))))::ins) s outs)
   (CFG authTestConductPB ssmConductPBStateInterp
      (secContextConductPB plCommand psgCommand incomplete)
      ins (NS s (exec (SLc (PSG psgCommand))))
      (Out s (exec (SLc (PSG psgCommand)))::outs)) ⟺
authTestConductPB
   (Name PlatoonSergeant says
    prop (SOME (SLc (PSG psgCommand)))) ∧
CFGInterpret (M,Oi,Os)
   (CFG authTestConductPB ssmConductPBStateInterp
      (secContextConductPB plCommand psgCommand incomplete)
      (Name PlatoonSergeant says
       prop (SOME (SLc (PSG psgCommand))))::ins) s outs) ∧
(M,Oi,Os) sat prop (SOME (SLc (PSG psgCommand)))
```

[PlatoonSergeant_psgCommandPB_lemma]

⊢ CFGInterpret (M,Oi,Os)
```
   (CFG authTestConductPB ssmConductPBStateInterp
      (secContextConductPB plCommand psgCommand incomplete)
      (Name PlatoonSergeant says
       prop (SOME (SLc (PSG psgCommand))))::ins) s outs) ⟹
(M,Oi,Os) sat prop (SOME (SLc (PSG psgCommand)))
```

Index