Contents

1 projectTypes Theory **3**
1.1 Datatypes 3
1.2 Theorems 3

2 projectUtilities Theory **4**
2.1 Theorems 4

3 projectSM Theory **12**
3.1 Theorems 12

4 projectSecurity Theory **13**
4.1 Definitions 13
4.2 Theorems 14

5 projectAssuranceExec Theory **16**
5.1 Theorems 16

1 projectTypes Theory

Built: 27 December 2018
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

commands = PlatoonLeaderCOM platoonLeaderCom | OmniCOM omniCom

omniCom = none | omniNA

output = Move_to_ORP | Form | Move | Secure_halt | NoActionTaken
| UnAuthenticated | UnAuthorized

platoonLeaderCom = form | move | secureHalt

principal = PlatoonLeader | Omni

state = MOVE_TO_PB | FORM | MOVE | SECURE_HALT

1.2 Theorems

[commands_distinct_clauses]

$\vdash \forall a'\ a.$ PlatoonLeaderCOM $a \neq$ OmniCOM a'

[commands_one_one]

$\vdash (\forall a\ a'.$
(PlatoonLeaderCOM a = PlatoonLeaderCOM a') $\iff$ $(a = a')) \wedge$
$\forall a\ a'.$ (OmniCOM a = OmniCOM a') $\iff$ $(a = a')$

[omniCom_distinct_clauses]

$\vdash$ none $\neq$ omniNA

[output_distinct_clauses]

$\vdash$ Move_to_ORP $\neq$ Form $\wedge$ Move_to_ORP $\neq$ Move $\wedge$
Move_to_ORP $\neq$ Secure_halt $\wedge$ Move_to_ORP $\neq$ NoActionTaken $\wedge$
Move_to_ORP $\neq$ UnAuthenticated $\wedge$ Move_to_ORP $\neq$ UnAuthorized $\wedge$
Form $\neq$ Move $\wedge$ Form $\neq$ Secure_halt $\wedge$ Form $\neq$ NoActionTaken $\wedge$
Form $\neq$ UnAuthenticated $\wedge$ Form $\neq$ UnAuthorized $\wedge$
Move $\neq$ Secure_halt $\wedge$ Move $\neq$ NoActionTaken $\wedge$
Move $\neq$ UnAuthenticated $\wedge$ Move $\neq$ UnAuthorized $\wedge$
Secure_halt $\neq$ NoActionTaken $\wedge$ Secure_halt $\neq$ UnAuthenticated $\wedge$
Secure_halt $\neq$ UnAuthorized $\wedge$
NoActionTaken $\neq$ UnAuthenticated $\wedge$
NoActionTaken $\neq$ UnAuthorized $\wedge$ UnAuthenticated $\neq$ UnAuthorized

[platoonLeaderCom_distinct_clauses]

$\vdash$ form $\neq$ move $\wedge$ form $\neq$ secureHalt $\wedge$ move $\neq$ secureHalt

[principal_distinct_clauses]

⊢ PlatoonLeader ≠ Omni

[state_distinct_clauses]

⊢ MOVE_TO_PB ≠ FORM ∧ MOVE_TO_PB ≠ MOVE ∧
MOVE_TO_PB ≠ SECURE_HALT ∧ FORM ≠ MOVE ∧ FORM ≠ SECURE_HALT ∧
MOVE ≠ SECURE_HALT

2 projectUtilities Theory

Built: 27 December 2018
Parent Theories: projectTypes, satList

2.1 Theorems

[getOmniCOM_def]

⊢ (getOmniCOM [] = NONE) ∧
(∀ *xs* *cmd*.
getOmniCOM (SOME (OmniCOM *cmd*)::*xs*) =
SOME (OmniCOM *cmd*)) ∧
(∀ *xs*. getOmniCOM (NONE::*xs*) = getOmniCOM *xs*) ∧
∀ *xs* v_4.
getOmniCOM (SOME (PlatoonLeaderCOM v_4)::*xs*) = getOmniCOM *xs*

[getOmniCOM_ind]

⊢ ∀ *P*.
P [] ∧ (∀ *cmd* *xs*. *P* (SOME (OmniCOM *cmd*)::*xs*)) ∧
(∀ *xs*. *P* *xs* ⇒ *P* (NONE::*xs*)) ∧
(∀ v_4 *xs*. *P* *xs* ⇒ *P* (SOME (PlatoonLeaderCOM v_4)::*xs*)) ⇒
∀ *v*. *P* *v*

[getOmniCOMx_def]

⊢ (getOmniCOMx [] = NONE) ∧
(∀ *xs* *cmd*.
getOmniCOMx
(Name Omni says prop (SOME (OmniCOM *cmd*))::*xs*) =
SOME (OmniCOM *cmd*)) ∧
(∀ *xs*. getOmniCOMx (TT::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs*. getOmniCOMx (FF::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_2. getOmniCOMx (prop v_2::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_3. getOmniCOMx (notf v_3::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_5 v_4. getOmniCOMx (v_4 andf v_5::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_7 v_6. getOmniCOMx (v_6 orf v_7::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_9 v_8. getOmniCOMx (v_8 impf v_9::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_{11} v_{10}.
getOmniCOMx (v_{10} eqf v_{11}::*xs*) = getOmniCOMx *xs*) ∧

```
(∀ xs v12. getOmniCOMx (v12 says TT::xs) = getOmniCOMx xs) ∧
(∀ xs v12. getOmniCOMx (v12 says FF::xs) = getOmniCOMx xs) ∧
(∀ xs v134.
   getOmniCOMx (Name v134 says prop NONE::xs) =
   getOmniCOMx xs) ∧
(∀ xs v144.
   getOmniCOMx
     (Name PlatoonLeader says prop (SOME v144)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v146.
   getOmniCOMx
     (Name Omni says prop (SOME (PlatoonLeaderCOM v146))::
          xs) =
   getOmniCOMx xs) ∧
(∀ xs v68 v136 v135.
   getOmniCOMx (v135 meet v136 says prop v68::xs) =
   getOmniCOMx xs) ∧
(∀ xs v68 v138 v137.
   getOmniCOMx (v137 quoting v138 says prop v68::xs) =
   getOmniCOMx xs) ∧
(∀ xs v69 v12.
   getOmniCOMx (v12 says notf v69::xs) = getOmniCOMx xs) ∧
(∀ xs v71 v70 v12.
   getOmniCOMx (v12 says (v70 andf v71)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v73 v72 v12.
   getOmniCOMx (v12 says (v72 orf v73)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v75 v74 v12.
   getOmniCOMx (v12 says (v74 impf v75)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v77 v76 v12.
   getOmniCOMx (v12 says (v76 eqf v77)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v79 v78 v12.
   getOmniCOMx (v12 says v78 says v79::xs) =
   getOmniCOMx xs) ∧
(∀ xs v81 v80 v12.
   getOmniCOMx (v12 says v80 speaks_for v81::xs) =
   getOmniCOMx xs) ∧
(∀ xs v83 v82 v12.
   getOmniCOMx (v12 says v82 controls v83::xs) =
   getOmniCOMx xs) ∧
(∀ xs v86 v85 v84 v12.
   getOmniCOMx (v12 says reps v84 v85 v86::xs) =
   getOmniCOMx xs) ∧
(∀ xs v88 v87 v12.
   getOmniCOMx (v12 says v87 domi v88::xs) =
   getOmniCOMx xs) ∧
```

```
(∀ xs v90 v89 v12.
   getOmniCOMx (v12 says v89 eqi v90::xs) = getOmniCOMx xs) ∧
(∀ xs v92 v91 v12.
   getOmniCOMx (v12 says v91 doms v92::xs) =
   getOmniCOMx xs) ∧
(∀ xs v94 v93 v12.
   getOmniCOMx (v12 says v93 eqs v94::xs) = getOmniCOMx xs) ∧
(∀ xs v96 v95 v12.
   getOmniCOMx (v12 says v95 eqn v96::xs) = getOmniCOMx xs) ∧
(∀ xs v98 v97 v12.
   getOmniCOMx (v12 says v97 lte v98::xs) = getOmniCOMx xs) ∧
(∀ xs v99 v12 v100.
   getOmniCOMx (v12 says v99 lt v100::xs) = getOmniCOMx xs) ∧
(∀ xs v15 v14.
   getOmniCOMx (v14 speaks_for v15::xs) = getOmniCOMx xs) ∧
(∀ xs v17 v16.
   getOmniCOMx (v16 controls v17::xs) = getOmniCOMx xs) ∧
(∀ xs v20 v19 v18.
   getOmniCOMx (reps v18 v19 v20::xs) = getOmniCOMx xs) ∧
(∀ xs v22 v21.
   getOmniCOMx (v21 domi v22::xs) = getOmniCOMx xs) ∧
(∀ xs v24 v23.
   getOmniCOMx (v23 eqi v24::xs) = getOmniCOMx xs) ∧
(∀ xs v26 v25.
   getOmniCOMx (v25 doms v26::xs) = getOmniCOMx xs) ∧
(∀ xs v28 v27.
   getOmniCOMx (v27 eqs v28::xs) = getOmniCOMx xs) ∧
(∀ xs v30 v29.
   getOmniCOMx (v29 eqn v30::xs) = getOmniCOMx xs) ∧
(∀ xs v32 v31.
   getOmniCOMx (v31 lte v32::xs) = getOmniCOMx xs) ∧
∀ xs v34 v33. getOmniCOMx (v33 lt v34::xs) = getOmniCOMx xs
```

[getOmniCOMx_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ cmd xs.
       P (Name Omni says prop (SOME (OmniCOM cmd))::xs)) ∧
    (∀ xs. P xs ⇒ P (TT::xs)) ∧ (∀ xs. P xs ⇒ P (FF::xs)) ∧
    (∀ v2 xs. P xs ⇒ P (prop v2::xs)) ∧
    (∀ v3 xs. P xs ⇒ P (notf v3::xs)) ∧
    (∀ v4 v5 xs. P xs ⇒ P (v4 andf v5::xs)) ∧
    (∀ v6 v7 xs. P xs ⇒ P (v6 orf v7::xs)) ∧
    (∀ v8 v9 xs. P xs ⇒ P (v8 impf v9::xs)) ∧
    (∀ v10 v11 xs. P xs ⇒ P (v10 eqf v11::xs)) ∧
    (∀ v12 xs. P xs ⇒ P (v12 says TT::xs)) ∧
    (∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
    (∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
    (∀ v144 xs.
```

$P\ xs \Rightarrow$
P (Name PlatoonLeader says prop (SOME $v144$)::xs)) $\wedge$
$(\forall v146\ xs.$
$P\ xs \Rightarrow$
P
(Name Omni says prop (SOME (PlatoonLeaderCOM $v146$))::
xs)) $\wedge$
$(\forall v135\ v136\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v135$ meet $v136$ says prop v_{68}::xs)) $\wedge$
$(\forall v137\ v138\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v137$ quoting $v138$ says prop v_{68}::xs)) $\wedge$
$(\forall v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says notf v_{69}::xs)) $\wedge$
$(\forall v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{70} andf v_{71})::xs)) $\wedge$
$(\forall v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{72} orf v_{73})::xs)) $\wedge$
$(\forall v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{74} impf v_{75})::xs)) $\wedge$
$(\forall v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{76} eqf v_{77})::xs)) $\wedge$
$(\forall v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{78} says v_{79}::xs)) $\wedge$
$(\forall v_{12}\ v_{80}\ v_{81}\ xs.$
$P\ xs \Rightarrow P$ (v_{12} says v_{80} speaks_for v_{81}::xs)) $\wedge$
$(\forall v_{12}\ v_{82}\ v_{83}\ xs.$
$P\ xs \Rightarrow P$ (v_{12} says v_{82} controls v_{83}::xs)) $\wedge$
$(\forall v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$P\ xs \Rightarrow P$ (v_{12} says reps v_{84} v_{85} v_{86}::xs)) $\wedge$
$(\forall v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{87} domi v_{88}::xs)) $\wedge$
$(\forall v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{89} eqi v_{90}::xs)) $\wedge$
$(\forall v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{91} doms v_{92}::xs)) $\wedge$
$(\forall v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{93} eqs v_{94}::xs)) $\wedge$
$(\forall v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{95} eqn v_{96}::xs)) $\wedge$
$(\forall v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{97} lte v_{98}::xs)) $\wedge$
$(\forall v_{12}\ v_{99}\ v100\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{99} lt $v100$::xs)) $\wedge$
$(\forall v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P$ (v_{14} speaks_for v_{15}::xs)) $\wedge$
$(\forall v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P$ (v_{16} controls v_{17}::xs)) $\wedge$
$(\forall v_{18}\ v_{19}\ v_{20}\ xs.\ P\ xs \Rightarrow P$ (reps v_{18} v_{19} v_{20}::xs)) $\wedge$
$(\forall v_{21}\ v_{22}\ xs.\ P\ xs \Rightarrow P$ (v_{21} domi v_{22}::xs)) $\wedge$
$(\forall v_{23}\ v_{24}\ xs.\ P\ xs \Rightarrow P$ (v_{23} eqi v_{24}::xs)) $\wedge$
$(\forall v_{25}\ v_{26}\ xs.\ P\ xs \Rightarrow P$ (v_{25} doms v_{26}::xs)) $\wedge$
$(\forall v_{27}\ v_{28}\ xs.\ P\ xs \Rightarrow P$ (v_{27} eqs v_{28}::xs)) $\wedge$
$(\forall v_{29}\ v_{30}\ xs.\ P\ xs \Rightarrow P$ (v_{29} eqn v_{30}::xs)) $\wedge$
$(\forall v_{31}\ v_{32}\ xs.\ P\ xs \Rightarrow P$ (v_{31} lte v_{32}::xs)) $\wedge$
$(\forall v_{33}\ v_{34}\ xs.\ P\ xs \Rightarrow P$ (v_{33} lt v_{34}::xs)) $\Rightarrow$
$\forall v.\ P\ v$

[getPlatoonLeaderCOM_def]

$\vdash$ (getPlatoonLeaderCOM [] = NONE) $\wedge$
$(\forall xs\ cmd.$
getPlatoonLeaderCOM (SOME (PlatoonLeaderCOM cmd)::xs) =
SOME (PlatoonLeaderCOM cmd)) $\wedge$
$(\forall xs.$
getPlatoonLeaderCOM (NONE::xs) = getPlatoonLeaderCOM xs) $\wedge$

$\forall\, xs\ v_5\,.$
  getPlatoonLeaderCOM (SOME (OmniCOM v_5)::xs) =
  getPlatoonLeaderCOM xs

[getPlatoonLeaderCOM_ind]

$\vdash \forall P\,.$
  P [] $\wedge$ ($\forall\, cmd\ xs\,.\ P$ (SOME (PlatoonLeaderCOM cmd)::xs)) $\wedge$
  ($\forall\, xs\,.\ P\ xs \Rightarrow P$ (NONE::xs)) $\wedge$
  ($\forall\, v_5\ xs\,.\ P\ xs \Rightarrow P$ (SOME (OmniCOM v_5)::xs)) $\Rightarrow$
  $\forall\, v\,.\ P\ v$

[getPlatoonLeaderCOMx_def]

$\vdash$ (getPlatoonLeaderCOMx [] = NONE) $\wedge$
($\forall\, xs\ cmd\,.$
  getPlatoonLeaderCOMx
    (Name PlatoonLeader says
     prop (SOME (PlatoonLeaderCOM cmd))::xs) =
  SOME (PlatoonLeaderCOM cmd)) $\wedge$
($\forall\, xs\,.$
  getPlatoonLeaderCOMx (TT::xs) = getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\,.$
  getPlatoonLeaderCOMx (FF::xs) = getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_2\,.$
  getPlatoonLeaderCOMx (prop v_2::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_3\,.$
  getPlatoonLeaderCOMx (notf v_3::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_5\ v_4\,.$
  getPlatoonLeaderCOMx (v_4 andf v_5::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_7\ v_6\,.$
  getPlatoonLeaderCOMx (v_6 orf v_7::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_9\ v_8\,.$
  getPlatoonLeaderCOMx (v_8 impf v_9::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_{11}\ v_{10}\,.$
  getPlatoonLeaderCOMx (v_{10} eqf v_{11}::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_{12}\,.$
  getPlatoonLeaderCOMx (v_{12} says TT::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v_{12}\,.$
  getPlatoonLeaderCOMx (v_{12} says FF::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$
($\forall\, xs\ v134\,.$
  getPlatoonLeaderCOMx (Name $v134$ says prop NONE::xs) =
  getPlatoonLeaderCOMx xs) $\wedge$

```
(∀ xs v147.
   getPlatoonLeaderCOMx
     (Name PlatoonLeader says prop (SOME (OmniCOM v147))::
          xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v144.
   getPlatoonLeaderCOMx
     (Name Omni says prop (SOME v144)::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v68 v136 v135.
   getPlatoonLeaderCOMx (v135 meet v136 says prop v68::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v68 v138 v137.
   getPlatoonLeaderCOMx
     (v137 quoting v138 says prop v68::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v69 v12.
   getPlatoonLeaderCOMx (v12 says notf v69::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v71 v70 v12.
   getPlatoonLeaderCOMx (v12 says (v70 andf v71)::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v73 v72 v12.
   getPlatoonLeaderCOMx (v12 says (v72 orf v73)::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v75 v74 v12.
   getPlatoonLeaderCOMx (v12 says (v74 impf v75)::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v77 v76 v12.
   getPlatoonLeaderCOMx (v12 says (v76 eqf v77)::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v79 v78 v12.
   getPlatoonLeaderCOMx (v12 says v78 says v79::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v81 v80 v12.
   getPlatoonLeaderCOMx (v12 says v80 speaks_for v81::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v83 v82 v12.
   getPlatoonLeaderCOMx (v12 says v82 controls v83::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v86 v85 v84 v12.
   getPlatoonLeaderCOMx (v12 says reps v84 v85 v86::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v88 v87 v12.
   getPlatoonLeaderCOMx (v12 says v87 domi v88::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v90 v89 v12.
   getPlatoonLeaderCOMx (v12 says v89 eqi v90::xs) =
   getPlatoonLeaderCOMx xs) ∧
```

```
(∀ xs v92 v91 v12.
   getPlatoonLeaderCOMx (v12 says v91 doms v92::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v94 v93 v12.
   getPlatoonLeaderCOMx (v12 says v93 eqs v94::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v96 v95 v12.
   getPlatoonLeaderCOMx (v12 says v95 eqn v96::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v98 v97 v12.
   getPlatoonLeaderCOMx (v12 says v97 lte v98::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v99 v12 v100.
   getPlatoonLeaderCOMx (v12 says v99 lt v100::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v15 v14.
   getPlatoonLeaderCOMx (v14 speaks_for v15::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v17 v16.
   getPlatoonLeaderCOMx (v16 controls v17::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v20 v19 v18.
   getPlatoonLeaderCOMx (reps v18 v19 v20::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v22 v21.
   getPlatoonLeaderCOMx (v21 domi v22::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v24 v23.
   getPlatoonLeaderCOMx (v23 eqi v24::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v26 v25.
   getPlatoonLeaderCOMx (v25 doms v26::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v28 v27.
   getPlatoonLeaderCOMx (v27 eqs v28::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v30 v29.
   getPlatoonLeaderCOMx (v29 eqn v30::xs) =
   getPlatoonLeaderCOMx xs) ∧
(∀ xs v32 v31.
   getPlatoonLeaderCOMx (v31 lte v32::xs) =
   getPlatoonLeaderCOMx xs) ∧
∀ xs v34 v33.
  getPlatoonLeaderCOMx (v33 lt v34::xs) =
  getPlatoonLeaderCOMx xs
```

[getPlatoonLeaderCOMx_ind]

```
⊢ ∀ P.
    P [] ∧
```

$(\forall\, cmd\ xs.$
$\quad P$
$\quad\quad$ (Name PlatoonLeader says
$\quad\quad$ prop (SOME (PlatoonLeaderCOM cmd))::xs)) $\wedge$
$(\forall\, xs.\ P\ xs \Rightarrow P$ (TT::xs)) $\wedge$ $(\forall\, xs.\ P\ xs \Rightarrow P$ (FF::xs)) $\wedge$
$(\forall\, v_2\ xs.\ P\ xs \Rightarrow P$ (prop v_2::xs)) $\wedge$
$(\forall\, v_3\ xs.\ P\ xs \Rightarrow P$ (notf v_3::xs)) $\wedge$
$(\forall\, v_4\ v_5\ xs.\ P\ xs \Rightarrow P$ (v_4 andf v_5::xs)) $\wedge$
$(\forall\, v_6\ v_7\ xs.\ P\ xs \Rightarrow P$ (v_6 orf v_7::xs)) $\wedge$
$(\forall\, v_8\ v_9\ xs.\ P\ xs \Rightarrow P$ (v_8 impf v_9::xs)) $\wedge$
$(\forall\, v_{10}\ v_{11}\ xs.\ P\ xs \Rightarrow P$ (v_{10} eqf v_{11}::xs)) $\wedge$
$(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says TT::xs)) $\wedge$
$(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says FF::xs)) $\wedge$
$(\forall\, v134\ xs.\ P\ xs \Rightarrow P$ (Name $v134$ says prop NONE::xs)) $\wedge$
$(\forall\, v147\ xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad$ (Name PlatoonLeader says prop (SOME (OmniCOM $v147$))::
$\quad\quad\quad xs$)) $\wedge$
$(\forall\, v144\ xs.$
$\quad P\ xs \Rightarrow P$ (Name Omni says prop (SOME $v144$)::xs)) $\wedge$
$(\forall\, v135\ v136\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P$ ($v135$ meet $v136$ says prop v_{68}::xs)) $\wedge$
$(\forall\, v137\ v138\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P$ ($v137$ quoting $v138$ says prop v_{68}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says notf v_{69}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{70} andf v_{71})::xs)) $\wedge$
$(\forall\, v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{72} orf v_{73})::xs)) $\wedge$
$(\forall\, v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{74} impf v_{75})::xs)) $\wedge$
$(\forall\, v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says (v_{76} eqf v_{77})::xs)) $\wedge$
$(\forall\, v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{78} says v_{79}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{80}\ v_{81}\ xs.$
$\quad P\ xs \Rightarrow P$ (v_{12} says v_{80} speaks_for v_{81}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{82}\ v_{83}\ xs.$
$\quad P\ xs \Rightarrow P$ (v_{12} says v_{82} controls v_{83}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$\quad P\ xs \Rightarrow P$ (v_{12} says reps v_{84} v_{85} v_{86}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{87} domi v_{88}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{89} eqi v_{90}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{91} doms v_{92}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{93} eqs v_{94}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{95} eqn v_{96}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{97} lte v_{98}::xs)) $\wedge$
$(\forall\, v_{12}\ v_{99}\ v100\ xs.\ P\ xs \Rightarrow P$ (v_{12} says v_{99} lt $v100$::xs)) $\wedge$
$(\forall\, v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P$ (v_{14} speaks_for v_{15}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P$ (v_{16} controls v_{17}::xs)) $\wedge$
$(\forall\, v_{18}\ v_{19}\ v_{20}\ xs.\ P\ xs \Rightarrow P$ (reps v_{18} v_{19} v_{20}::xs)) $\wedge$
$(\forall\, v_{21}\ v_{22}\ xs.\ P\ xs \Rightarrow P$ (v_{21} domi v_{22}::xs)) $\wedge$
$(\forall\, v_{23}\ v_{24}\ xs.\ P\ xs \Rightarrow P$ (v_{23} eqi v_{24}::xs)) $\wedge$

```
(∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
(∀ v27 v28 xs. P xs ⇒ P (v27 eqs v28::xs)) ∧
(∀ v29 v30 xs. P xs ⇒ P (v29 eqn v30::xs)) ∧
(∀ v31 v32 xs. P xs ⇒ P (v31 lte v32::xs)) ∧
(∀ v33 v34 xs. P xs ⇒ P (v33 lt v34::xs)) ⇒
∀ v. P v
```

3 projectSM Theory

Built: 27 December 2018
Parent Theories: projectUtilities, ssm

3.1 Theorems

[NOut_def]

```
⊢ (NOut MOVE_TO_PB (exec x) =
   if getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM form) then
     Form
   else NoActionTaken) ∧
  (NOut FORM (exec x) =
   if getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM move) then
     Move
   else NoActionTaken) ∧
  (NOut MOVE (exec x) =
   if
     getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM secureHalt)
   then
     Secure_halt
   else NoActionTaken) ∧ (NOut s (trap v0) = UnAuthorized) ∧
  (NOut s (discard v1) = UnAuthenticated)
```

[NOut_ind]

```
⊢ ∀ P.
    (∀ x. P MOVE_TO_PB (exec x)) ∧ (∀ x. P FORM (exec x)) ∧
    (∀ x. P MOVE (exec x)) ∧ (∀ s v0. P s (trap v0)) ∧
    (∀ s v1. P s (discard v1)) ∧
    (∀ v6. P SECURE_HALT (exec v6)) ⇒
    ∀ v v1. P v v1
```

[NS_def]

```
⊢ (NS MOVE_TO_PB (exec x) =
   if getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM form) then
     FORM
   else MOVE_TO_PB) ∧
  (NS FORM (exec x) =
   if getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM move) then
     MOVE
```

```
    else FORM) ∧
   (NS MOVE (exec x) =
    if
      getPlatoonLeaderCOM x = SOME (PlatoonLeaderCOM secureHalt)
    then
      SECURE_HALT
    else MOVE) ∧ (NS s (trap v0) = s) ∧ (NS s (discard v1) = s)
```

[NS_ind]

```
⊢ ∀P.
    (∀x. P MOVE_TO_PB (exec x)) ∧ (∀x. P FORM (exec x)) ∧
    (∀x. P MOVE (exec x)) ∧ (∀s v0. P s (trap v0)) ∧
    (∀s v1. P s (discard v1)) ∧
    (∀v6. P SECURE_HALT (exec v6)) ⇒
    ∀v v1. P v v1
```

4 projectSecurity Theory

Built: 27 December 2018
Parent Theories: projectUtilities, ssm

4.1 Definitions

[globalAuth_def]

```
⊢ ∀x. globalAuth x = [TT]
```

[stateAuth_def]

```
⊢ ∀s x.
    stateAuth s x =
    if s = MOVE_TO_PB then
      if
        getPlatoonLeaderCOMx x = SOME (PlatoonLeaderCOM form)
      then
        [Name PlatoonLeader controls
         prop (SOME (PlatoonLeaderCOM form))]
      else [prop NONE]
    else if s = FORM then
      if
        getPlatoonLeaderCOMx x = SOME (PlatoonLeaderCOM move)
      then
        [Name PlatoonLeader controls
         prop (SOME (PlatoonLeaderCOM move))]
      else [prop NONE]
    else if s = MOVE then
      if
        getPlatoonLeaderCOMx x =
        SOME (PlatoonLeaderCOM secureHalt)
```

```
      then
        [Name PlatoonLeader controls
         prop (SOME (PlatoonLeaderCOM secureHalt))]
      else [prop NONE]
    else [prop NONE]
```

4.2 Theorems

[authentication_def]

$$
\begin{aligned}
&\vdash \texttt{(authentication}\\
&\quad \texttt{(Name PlatoonLeader says}\\
&\quad \texttt{prop (SOME (PlatoonLeaderCOM } x'\texttt{)))} \iff \texttt{T)} \wedge\\
&\texttt{(authentication (Name Omni says prop (SOME (OmniCOM } x\texttt{)))} \iff\\
&\texttt{T)} \wedge \texttt{(authentication TT} \iff \texttt{F)} \wedge \texttt{(authentication FF} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (prop } v\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (notf } v_1\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_2\texttt{ andf }v_3\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_4\texttt{ orf }v_5\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_6\texttt{ impf }v_7\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_8\texttt{ eqf }v_9\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says TT)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says FF)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says prop NONE)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication}\\
&\quad \texttt{(Name Omni says prop (SOME (PlatoonLeaderCOM } v144\texttt{)))} \iff\\
&\texttt{F)} \wedge\\
&\texttt{(authentication}\\
&\quad \texttt{(Name PlatoonLeader says prop (SOME (OmniCOM } v145\texttt{)))} \iff\\
&\texttt{F)} \wedge \texttt{(authentication (Name }v_{66}\texttt{ says notf }v_{77}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says (}v_{78}\texttt{ andf }v_{79}\texttt{))} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says (}v_{80}\texttt{ orf }v_{81}\texttt{))} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says (}v_{82}\texttt{ impf }v_{83}\texttt{))} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says (}v_{84}\texttt{ eqf }v_{85}\texttt{))} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{86}\texttt{ says }v_{87}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{88}\texttt{ speaks\_for }v_{89}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{90}\texttt{ controls }v_{91}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says reps }v_{92}\ v_{93}\ v_{94}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{95}\texttt{ domi }v_{96}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{97}\texttt{ eqi }v_{98}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v_{99}\texttt{ doms }v100\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v101\texttt{ eqs }v102\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v103\texttt{ eqn }v104\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v105\texttt{ lte }v106\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (Name }v_{66}\texttt{ says }v107\texttt{ lt }v108\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_{67}\texttt{ meet }v_{68}\texttt{ says }v_{11}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_{69}\texttt{ quoting }v_{70}\texttt{ says }v_{11}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_{12}\texttt{ speaks\_for }v_{13}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (}v_{14}\texttt{ controls }v_{15}\texttt{)} \iff \texttt{F)} \wedge\\
&\texttt{(authentication (reps }v_{16}\ v_{17}\ v_{18}\texttt{)} \iff \texttt{F)} \wedge
\end{aligned}
$$

```
(authentication (v19 domi v20) ⟺ F) ∧
(authentication (v21 eqi v22) ⟺ F) ∧
(authentication (v23 doms v24) ⟺ F) ∧
(authentication (v25 eqs v26) ⟺ F) ∧
(authentication (v27 eqn v28) ⟺ F) ∧
(authentication (v29 lte v30) ⟺ F) ∧
(authentication (v31 lt v32) ⟺ F)
```

[authentication_ind]

```
⊢ ∀P.
    (∀x.
       P
         (Name PlatoonLeader says
          prop (SOME (PlatoonLeaderCOM x)))) ∧
    (∀x. P (Name Omni says prop (SOME (OmniCOM x)))) ∧ P TT ∧
    P FF ∧ (∀v. P (prop v)) ∧ (∀v1. P (notf v1)) ∧
    (∀v2 v3. P (v2 andf v3)) ∧ (∀v4 v5. P (v4 orf v5)) ∧
    (∀v6 v7. P (v6 impf v7)) ∧ (∀v8 v9. P (v8 eqf v9)) ∧
    (∀v66. P (Name v66 says TT)) ∧
    (∀v66. P (Name v66 says FF)) ∧
    (∀v66. P (Name v66 says prop NONE)) ∧
    (∀v144.
       P
         (Name Omni says
          prop (SOME (PlatoonLeaderCOM v144)))) ∧
    (∀v145.
       P
         (Name PlatoonLeader says
          prop (SOME (OmniCOM v145)))) ∧
    (∀v66 v77. P (Name v66 says notf v77)) ∧
    (∀v66 v78 v79. P (Name v66 says (v78 andf v79))) ∧
    (∀v66 v80 v81. P (Name v66 says (v80 orf v81))) ∧
    (∀v66 v82 v83. P (Name v66 says (v82 impf v83))) ∧
    (∀v66 v84 v85. P (Name v66 says (v84 eqf v85))) ∧
    (∀v66 v86 v87. P (Name v66 says v86 says v87)) ∧
    (∀v66 v88 v89. P (Name v66 says v88 speaks_for v89)) ∧
    (∀v66 v90 v91. P (Name v66 says v90 controls v91)) ∧
    (∀v66 v92 v93 v94. P (Name v66 says reps v92 v93 v94)) ∧
    (∀v66 v95 v96. P (Name v66 says v95 domi v96)) ∧
    (∀v66 v97 v98. P (Name v66 says v97 eqi v98)) ∧
    (∀v66 v99 v100. P (Name v66 says v99 doms v100)) ∧
    (∀v66 v101 v102. P (Name v66 says v101 eqs v102)) ∧
    (∀v66 v103 v104. P (Name v66 says v103 eqn v104)) ∧
    (∀v66 v105 v106. P (Name v66 says v105 lte v106)) ∧
    (∀v66 v107 v108. P (Name v66 says v107 lt v108)) ∧
    (∀v67 v68 v11. P (v67 meet v68 says v11)) ∧
    (∀v69 v70 v11. P (v69 quoting v70 says v11)) ∧
    (∀v12 v13. P (v12 speaks_for v13)) ∧
    (∀v14 v15. P (v14 controls v15)) ∧
```

```
(∀ v16 v17 v18. P (reps v16 v17 v18)) ∧
(∀ v19 v20. P (v19 domi v20)) ∧
(∀ v21 v22. P (v21 eqi v22)) ∧
(∀ v23 v24. P (v23 doms v24)) ∧
(∀ v25 v26. P (v25 eqs v26)) ∧ (∀ v27 v28. P (v27 eqn v28)) ∧
(∀ v29 v30. P (v29 lte v30)) ∧ (∀ v31 v32. P (v31 lt v32)) ⇒
∀ v. P v
```

5 projectAssuranceExec Theory

Built: 27 December 2018
Parent Theories: projectSecurity

5.1 Theorems

[FORM_exec_move_lemma1]

```
⊢ ∀ M Oi Os.
    CFGInterpret (M, Oi, Os)
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM move))]::ins) FORM
         outs) ⇒
    (M, Oi, Os) satList
    propCommandList
      [Name PlatoonLeader says
       prop (SOME (PlatoonLeaderCOM move))]
```

[FORM_exec_move_lemma2]

```
⊢ ∀ NS Out M Oi Os.
    TR (M, Oi, Os)
      (exec
         (inputList
            [Name PlatoonLeader says
             prop (SOME (PlatoonLeaderCOM move))]))
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM move))]::ins) FORM
         outs)
      (CFG authentication stateAuth globalAuth ins
         (NS FORM
            (exec
               (inputList
                  [Name PlatoonLeader says
                   prop (SOME (PlatoonLeaderCOM move))])))
         (Out FORM
            (exec
               (inputList
```

```
                  [Name PlatoonLeader says
                   prop (SOME (PlatoonLeaderCOM move))])::
              outs)) ⟺
     authenticationTest authentication
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM move))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM move))]::ins) FORM
          outs) ∧
     (M,Oi,Os) satList
     propCommandList
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM move))]
```

[FORM_exec_move_thm]

```
 ⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os) (exec [SOME (PlatoonLeaderCOM move)])
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM move))]::ins) FORM
          outs)
       (CFG authentication stateAuth globalAuth ins
          (NS FORM (exec [SOME (PlatoonLeaderCOM move)]))
          (Out FORM (exec [SOME (PlatoonLeaderCOM move)])::
               outs)) ⟺
     authenticationTest authentication
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM move))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM move))]::ins) FORM
          outs) ∧
     (M,Oi,Os) satList [prop (SOME (PlatoonLeaderCOM move))]
```

[MOVE_exec_secureHalt_lemma1]

```
 ⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM secureHalt))]::ins)
          MOVE outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM secureHalt))]
```

[MOVE_exec_secureHalt_lemma2]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os)
      (exec
         (inputList
            [Name PlatoonLeader says
             prop (SOME (PlatoonLeaderCOM secureHalt))]))
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM secureHalt))]::ins)
         MOVE outs)
      (CFG authentication stateAuth globalAuth ins
         (NS MOVE
            (exec
               (inputList
                  [Name PlatoonLeader says
                   prop
                     (SOME (PlatoonLeaderCOM secureHalt))]))))
         (Out MOVE
            (exec
               (inputList
                  [Name PlatoonLeader says
                   prop
                     (SOME (PlatoonLeaderCOM secureHalt))]))::
              outs)) ⟺
    authenticationTest authentication
      [Name PlatoonLeader says
       prop (SOME (PlatoonLeaderCOM secureHalt))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM secureHalt))]::ins)
         MOVE outs) ∧
    (M,Oi,Os) satList
    propCommandList
      [Name PlatoonLeader says
       prop (SOME (PlatoonLeaderCOM secureHalt))]
```

[MOVE_exec_secureHalt_thm]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os) (exec [SOME (PlatoonLeaderCOM secureHalt)])
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM secureHalt))]::ins)
         MOVE outs)
      (CFG authentication stateAuth globalAuth ins
         (NS MOVE (exec [SOME (PlatoonLeaderCOM secureHalt)]))
         (Out MOVE
            (exec [SOME (PlatoonLeaderCOM secureHalt)])::
```

```
           outs)) ⟺
     authenticationTest authentication
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM secureHalt))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM secureHalt))]::ins)
          MOVE outs) ∧
     (M,Oi,Os) satList
     [prop (SOME (PlatoonLeaderCOM secureHalt))]
```

[MOVE_TO_PB_exec_form_lemma1]

```
 ⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM form))]::ins)
          MOVE_TO_PB outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name PlatoonLeader says
        prop (SOME (PlatoonLeaderCOM form))]
```

[MOVE_TO_PB_exec_form_lemma2]

```
 ⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os)
       (exec
          (inputList
             [Name PlatoonLeader says
              prop (SOME (PlatoonLeaderCOM form))]))
       (CFG authentication stateAuth globalAuth
          ([Name PlatoonLeader says
            prop (SOME (PlatoonLeaderCOM form))]::ins)
          MOVE_TO_PB outs)
       (CFG authentication stateAuth globalAuth ins
          (NS MOVE_TO_PB
             (exec
                (inputList
                   [Name PlatoonLeader says
                    prop (SOME (PlatoonLeaderCOM form))])))
          (Out MOVE_TO_PB
             (exec
                (inputList
                   [Name PlatoonLeader says
                    prop (SOME (PlatoonLeaderCOM form))]))::
               outs)) ⟺
     authenticationTest authentication
       [Name PlatoonLeader says
```

```
      prop (SOME (PlatoonLeaderCOM form))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM form))]::ins)
         MOVE_TO_PB outs) ∧
    (M,Oi,Os) satList
    propCommandList
      [Name PlatoonLeader says
       prop (SOME (PlatoonLeaderCOM form))]
```

[MOVE_TO_PB_exec_form_thm]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os) (exec [SOME (PlatoonLeaderCOM form)])
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM form))]::ins)
         MOVE_TO_PB outs)
      (CFG authentication stateAuth globalAuth ins
         (NS MOVE_TO_PB (exec [SOME (PlatoonLeaderCOM form)]))
         (Out MOVE_TO_PB
            (exec [SOME (PlatoonLeaderCOM form)])::outs)) ⟺
    authenticationTest authentication
      [Name PlatoonLeader says
       prop (SOME (PlatoonLeaderCOM form))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name PlatoonLeader says
           prop (SOME (PlatoonLeaderCOM form))]::ins)
         MOVE_TO_PB outs) ∧
    (M,Oi,Os) satList [prop (SOME (PlatoonLeaderCOM form))]
```

Index

projectAssuranceExec Theory, 16
Theorems, 16
FORM_exec_move_lemma1, 16
FORM_exec_move_lemma2, 16
FORM_exec_move_thm, 17
MOVE_exec_secureHalt_lemma1, 17
MOVE_exec_secureHalt_lemma2, 18
MOVE_exec_secureHalt_thm, 18
MOVE_TO_PB_exec_form_lemma1, 19
MOVE_TO_PB_exec_form_lemma2, 19
MOVE_TO_PB_exec_form_thm, 20
projectSecurity Theory, 13
Definitions, 13
globalAuth_def, 13
stateAuth_def, 13
Theorems, 14
authentication_def, 14
authentication_ind, 15
projectSM Theory, 12
Theorems, 12
NOut_def, 12
NOut_ind, 12
NS_def, 12
NS_ind, 13
projectTypes Theory, 3
Datatypes, 3
Theorems, 3
commands_distinct_clauses, 3
commands_one_one, 3
omniCom_distinct_clauses, 3
output_distinct_clauses, 3
platoonLeaderCom_distinct_clauses, 3
principal_distinct_clauses, 4
state_distinct_clauses, 4
projectUtilities Theory, 4
Theorems, 4
getOmniCOM_def, 4
getOmniCOM_ind, 4
getOmniCOMx_def, 4
getOmniCOMx_ind, 6
getPlatoonLeaderCOM_def, 7
getPlatoonLeaderCOM_ind, 8
getPlatoonLeaderCOMx_def, 8
getPlatoonLeaderCOMx_ind, 10