Equivalence of Group-Centric Collaboration with Expedient Insiders (GEI) and LBAC with Collaborative Compartments (LCC)

Tahmina Ahmed[1], Ravi Sandhu[1], Khalid Bijon[1] and Ram Krishnan[2]
[1]Institute for Cyber Security & Department of Computer Science
[2]Institute for Cyber Security & Department of Electrical and Computer Engineering
University of Texas at San Antonio

Abstract

Equivalence of access control models can be proved by comparing their expressive power. Tripunitara and Li [3] have given a generalized theoretical formulation for comparing expressive power of access control models via simulations that preserve security properties which are called state matching reductions. This report gives a formal proof of a state matching reduction from Group-Centric Collaboration with Expedient Insiders (GEI) [1] to LBAC with Collaborative Compartments (LCC), a model defined in this report, and vice versa. So GEI and LCC are equivalent in their expressive power as per [3].

I. Introduction

Tripunitara and Li [3] define access control models as a set of access control schemes where each scheme consists of a set of states and state transition rules. Formally a scheme is represented as a 4-tuple $\langle \Gamma, Q, \vdash, \Psi \rangle$ as follows.

- Γ is a set of states where each state contains the necessary information to decide access control on that particular state.
- Q is a set of queries.
- $\vdash$: $\Gamma \times Q \rightarrow \{true, false\}$ is the entailment relation that verifies whether a query $q \in Q$ holds in a particular state $\gamma \in \Gamma$. If q is valid in state γ it is written as ($\gamma \vdash q$) or ($\gamma \not\vdash q$) otherwise.
- Ψ is a set of state transition rules where each $\psi \in \Psi$ determines how the state changes for that choice of ψ.

Given two access control schemes A=$\langle \Gamma^A, Q^A, \vdash^A, \Psi^A \rangle$ and B=$\langle \Gamma^B, Q^B, \vdash^B, \Psi^B \rangle$ a mapping from A to B is defined as a function σ that maps each pair $\langle \gamma^A, \psi^A \rangle$ to a pair $\langle \gamma^B, \psi^B \rangle$, and each query q^A to q^B. Formally a mapping is represented as $\sigma : (\Gamma^A \times \Psi^A) \cup Q^A \rightarrow (\Gamma^B \times \Psi^B) \cup Q^B$. States γ^A and γ^B are said to be equivalent under the mapping σ when for every $q^A \in Q^A$, $\gamma^A \vdash^A q^A$ if and only if $\gamma^B \vdash^B \sigma(q^A)$. This leads up to the definition for a state matching reduction as follows.

Definition 1. *(State Matching Reduction) Given two schemes A and B, a mapping σ from A to B is a state matching reduction if for every $\gamma^A \in \Gamma^A$ and every $\psi^A \in \Psi^A$, we have the following two properties where $\langle \gamma^B, \psi^B \rangle = \sigma(\langle \gamma^A, \psi^A \rangle)$.*

1) *For every state γ_1^A in scheme A such that $\gamma^A \xrightarrow{*}_{\psi^A} \gamma_1^A$, there exists γ_1^B in scheme B such that $\gamma^B \xrightarrow{*}_{\psi^B} \gamma_1^B$ and γ_1^A and γ_1^B are equivalent.*
2) *For every state γ_1^B in scheme B such that $\gamma^B \xrightarrow{*}_{\psi^B} \gamma_1^B$ there exists γ_1^A in scheme A such that $\gamma^A \xrightarrow{*}_{\psi^A} \gamma_1^A$, and γ_1^A and γ_1^B are equivalent.*

The significance of state-matching reductions is expressed in Theorem 1 of [3] which asserts that: Given two schemes A and B, a mapping σ from A to B is strongly security-preserving (in a precise formal sense) if and only if σ is a state-matching reduction. Two schemes are said to be equivalent if there is a state-matching reduction from one to the other, and vice versa.

The goal of this technical report is to formally prove the equivalence of two specific schemes using the above framework. One scheme is called Group-Centric Collaboration with Expedient Insiders (GEI). It was introduced and motivated in [1]. The other scheme is newly defined in this report. It is called LBAC with Collaborative Compartments (LCC), where LBAC is Lattice-Based Access Control [2]. Motivation and explanation of LCC will be provided in a paper currently under preparation. This report only provides a formal definition of LCC as a scheme.

The rest of the report is organized as follows. Section II gives a definition of GEI as a scheme. Section III does the same for LCC. Section IV defines a mapping from LCC to GEI. Section VI proves that this mapping is a state-matching reduction. Section VII conversely defines a mapping from GEI to LCC, which is formally proved to be state-matching in section VIII.

II. GEI Scheme

The GEI scheme is defined in three tables. The elements of each state $\gamma \in \Gamma$ are defined in Table I. Each $\psi \in \Psi$ is a state transition rule and is defined in the Column 1 of Tables II and III. Each $q \in Q$ is defined in Column 2 of Table II and III.

The GEI Scheme that is defined in this report is slightly different from the one defined in [1] in respect of lattice structure and object version. Here the lattice is more structured with specified categories, security compartments and levels whereas in [1] the lattice is more generic with unspecified structure. In [1] version space is an infinite universal set but in this report version is a finite non-empty set.

III. LCC Scheme

The elements of each state $\gamma \in \Gamma$ are defined in Table IV. Each $\psi \in \Psi$ is state transition rule and is defined in the Column 1 of Table V and VI. Each $q \in Q$ is defined in Column 2 of Table V and VI .

TABLE I
GEI STATE

Element#	**Global Sets and Symbols:**
1.	$CG_\gamma \subset \mathcal{CG}$, is the finite and strict subset of countably infinite set $\mathcal{CG}$.
2.	C_γ = C, is finite set of existing unordered categories
3.	L_γ = L, is finite set of existing hierarchical ordered security levels
4.	SL_γ = SL, is finite lattice of security compartments where SL = L $\times$ 2^C
5.	$\succeq_\gamma$ = $\succeq$, is finite dominance relation $\subseteq$ L $\times$ L where $\forall$l1,l2 $\in$ L and $\forall$c1,c2 $\in$ C. $\succeq$ = $\{$((l1,c1), (l2,c2)) $\mid$ l1$\succeq$ l2 $\wedge$ c1 $\supseteq$ c2$\}$
6.	$\oplus_\gamma$ = $\oplus$, is join operator where (l1,c1) $\oplus$ (l2,c2) = (max(l1,l2),c1$\cup$c2)
7.	$U_\gamma \subset \mathcal{U}$, is finite and strict subset of countably infinite set $\mathcal{U}$.
8.	$O_\gamma \subset \mathcal{O}$, is finite and strict subset of countably infinite set $\mathcal{O}$.
9.	$S_\gamma \subset \mathcal{S}$, is finite and strict subset of countably infinite set $\mathcal{S}$.
10.	$UTYPE_\gamma$ = UTYPE = $\{$insider, expedient_insider, outsider$\}$ is the finite set of user's type
11.	$STYPE_\gamma$ = STYPE = $\{$RO, RW$\}$ is the finite set of subject's type.
12.	Org, is the entity Organization, a Constant.
	User Related State Elements:
13.	hierclearanceOfUser: $U_\gamma \rightarrow L_\gamma$, this function maps each user to a security level.
14.	compcategoryOfUser: $U_\gamma \rightarrow 2^{C_\gamma}$, this function maps each user to compartments.
15.	uCG: $U_\gamma \rightarrow 2^{CG_\gamma}$, this function maps each user to zero or more groups.
16.	orgAdmin: $U_\gamma \rightarrow$ $\{$true,false$\}$, this function maps each user to true if she is an admin of Org
17.	cgAdmin: $U_\gamma \rightarrow 2^{CG_\gamma}$, this function maps each user to zero or more groups if he is an administrative user of the group.
18.	uType: $U_\gamma \rightarrow UTYPE_\gamma$, this function maps each user to a user type.
	Objects Related State Elements:
19.	hierclassificationOfObject: $O_\gamma \rightarrow L_\gamma$, this function maps each object to a security levels.
20.	compcategoryOfObject: $O_\gamma \rightarrow C_\gamma$, this function maps each object to compartment.
21.	origin: $O_\gamma \rightarrow CG_\gamma \cup \{Org\}$, this function maps each object to the entity (group or Org) where it was created.
22.	versions: $O_\gamma \rightarrow 2^{Univ\_V}_{finite}$ - ϕ, this function maps each object to all its existing versions where $\mathcal{UNIV\_V}$ is countably infinite set of all possible versions /* $2^{Univ\_V}_{finite}$ is finite set of existing versions that is a subset of $\mathcal{UNIV\_V}$.*/
	Subject Related State Elements:
23.	hierclearanceOfSubject: $S_\gamma \rightarrow L_\gamma$, this function maps each subject to a security levels.
24.	compcategoryOfSubject: $S_\gamma \rightarrow C_\gamma$, this function maps each subject to compartment.
25.	owner: $S_\gamma \rightarrow U_\gamma$, this function maps each subject to the user who created this.
26.	belongsTo: $S_\gamma \hookrightarrow CG_\gamma$, this function maps each RW subject (not RO subject) to the group where it was created.
27.	type: $S_\gamma \rightarrow STYPE_\gamma$, this function maps each subject to a subject type.
	Object Version Related State Elements:
28.	For each o $\in O_\gamma$, vMember$_o$: versions(o)$\rightarrow$ $2^{CG_\gamma \cup \{Org\}}$- ϕ, this functions maps each version of every object to one or more entity (group or Org) where this version is available to access.
29.	For each o $\in O_\gamma$, hierclassificationOfVersion$_o$: versions(o)$\rightarrow$ L_γ, this function maps each version to a security levels.
30.	For each o $\in O_\gamma$, compcategoryOfVersion$_o$: versions(o) $\rightarrow 2^{C_\gamma}$ this function maps each subject to compartment.

TABLE II
STATE TRANSITION AND QUERY OF GSIS-EXPEDIENT-INSIDER(PART 1: ADMIN MODEL)

Op.#	Operation	Authorization Query	State Element Update on State Transition
1.	**Create_Insider**(u1,u2,uType,sl,cp) /*Admin u1 creates user u2 as insider*/	u1 $\in$ U $\wedge$ u2 $\notin$ U $\wedge$ orgAdmin(u1)=True$\wedge$ sl $\in$ L $\wedge$cp $\subseteq$ C$\wedge$ uType=Insider	***if*** uType=Insider ***then*** hierclearanceOfUser$'$(u2)=sl compcategoryOfUser$'$(u2)=cp uType(u2)$'$=Insider U$'$ = U $\cup$ {u2}
2.	**Create_OutSider**(u1,u2,uType,sl,cp) /*Admin u1 creates user u2 as outsider*/	u1 $\in$ U $\wedge$ u2 $\notin$ U $\wedge$ orgAdmin(u1)=True$\wedge$ sl $\in$ L $\wedge$ cp $\subseteq$ C$\wedge$ uType=Outsider	uType(u2)$'$=Outsider U$'$ = U $\cup$ {u2}

3.	**Delete_User**(u1,u2)
	/*Admin u1 creates user
	u2 as outsider*/
		u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$
		orgAdmin(u1)=True$\wedge$ sl $\in$ L
		$\wedge$ cp $\subseteq$ C
			if(utype(u2)=Insider) then
			forall s $\in$ S
			if(owner(s)=u2)
			owner$'$ = owner-{s$\rightarrow$owner(s)}
			S$'$ = S-{s}
			uType$'$ = uType - { u2 $\rightarrow$ uType(u2)}
			U$'$ = U - {u2}

4.	**Establish**(u, cg) /*Admin user u establishes new collaboration group cg*/	u $\in$ U $\wedge$ cg $\notin$ CG $\wedge$ orgAdmin(u)=True	cgAdmin$'$(u) = cgAdmin(u) $\cup$ {cg} CG$'$ = CG $\cup$ {cg}
5.	**Join_Insider**(u1,u2,cg) /*Admin u1 grants cg membership to a true insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cg $\in$ CG $\wedge$ cg $\in$ cgAdmin(u1) $\wedge$ uType(u2) = Insider $\wedge$ cg $\notin$ uCG(u2)	uCG$'$(u2) = uCG(u2) $\cup$ {cg}
6.	**Leave_Insider**(u1,u2,cg) /*Admin u1 revokes cg membership from a true insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cg $\in$ CG $\wedge$ cg $\in$ cgAdmin(u1) $\wedge$ cg $\in$ uCG(u2) $\wedge$ uType(u2) = Insider	uCG$'$(u2) = uCG(u2) - {cg} ***forall*** s $\in$ S ***if*** owner(s) = u2 $\wedge$ belongsTo(s) = cg ***then*** S$'$ = S - {s}
7.	**Join_Outsider**(u1,u2,cg,sl,cp) /*Admin u1 grants cg membership to an expedient insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cg $\in$ CG $\wedge$ cg $\in$ cgAdmin(u1) $\wedge$ cg $\notin$ uCG(u2) $\wedge$ uType(u2)= Outsider $\wedge$ sl $\in$ L $\wedge$ cp $\subseteq$ C	uType$'$(u2) = Expedient_Insider ***if*** uCG(u2) = $\emptyset$ ***then*** hierclearanceOfUser$'$(u2) = sl compcategoryOfUser$'$(u2) = cp uCG$'$(u2) = uCG(u2) $\cup$ {cg}
8.	**Leave_Expedient_Insider**(u1,u2,cg) /*Admin u1 revokes cg membership from an expedient insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cg $\in$ CG $\wedge$ cg $\in$ cgAdmin(u1) $\wedge$ cg $\in$ uCG(u2) $\wedge$ uType(u2) = Expedient_Insider	uCG$'$(u2) = uCG(u2) - {cg} ***forall*** s $\in$ S ***if*** owner(s) = u2 $\wedge$ belongsTo(s) = cg ***then*** S$'$ = S - {s} /*Kill subjects belongsTo the respective insider*/ ***if*** uCG(u2) = $\emptyset$ ***then*** hierclearanceOfUser$'$ = hierclearanceOfUser-{u2$\rightarrow$hierclearanceOfUser(u2)} compcategoryOfUser$'$ = compcategoryOfUser-{u2$\rightarrow$compcategoryOfUser(u2)} uType(u2) = Outsider
9.	**Add**(u,o,v,cg) /*Admin u adds version v of object o from Org to cg*/	u $\in$ U $\wedge$ cg $\in$ CG $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cg $\in$ cgAdmin(u) $\wedge$ cg $\notin$ vMember$_o$(v)	vMember$'_o$(v) = vMember$_o$(v) $\cup$ {cg}
10.	**Remove**(u,o,v,cg) /*Admin u removes version v of object o from cg*/	u $\in$ U $\wedge$ cg $\in$ CG $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cg $\in$ cgAdmin(u) $\wedge$ cg $\in$ vMember$_o$(v)	vMember$'_o$(v) = vMember$_o$(v) - {cg}
11.	**Import**(u,o1,v1,o2,cg) /*Admin u imports version v1 of object o1 to new version v2 of object o2 in Org*/	u $\in$ U $\wedge$ cg $\in$ CG $\wedge$ v1 $\in$ versions(o) $\wedge$ o1,o2 $\in$ O $\wedge$ origin(o2) = Org $\wedge$ cg $\in$ cgAdmin(u) $\wedge$ origin(o1) = cg $\wedge$hierclassificationOfObject(o1) = hierclassificationOfObject(o2) $\wedge$ compcategoryOfObject(o2) $\supseteq$ compcategoryOfObject(o1)	versions$'$(o2) = versions(o2) $\cup$ {v2} vMember$'$(o2,v2) = {Org} hierclassificationOfVersion$_{o2}$(v2) = hierclassificationOfObject(o2) compcategoryOfVersion$_{o2}$(v2)=compcategoryOfObject(o2)
12.	**Merge**(u,o,v,cg) /*Admin u merges version v of object o from cg to Org*/	u $\in$ U $\wedge$ cg $\in$ CG $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cg $\in$ cgAdmin(u) $\wedge$cg $\in$ vMember$_o$(v) $\wedge$ origin(o) = Org $\wedge$ v $\in$ versions(o)	vMember$'_o$(v) = vMember$_o$(v) $\cup$ {Org}
13.	**Disband**(u, cg) /*Admin u disbands a collaboration group cg*/	u $\in$ U $\wedge$ cg $\in$ CG $\wedge$ cg $\in$ cgAdmin(u)	***forall*** u1 $\in$ U ***if*** cg $\in$ uCG(u1) ***then*** uCG$'$(u1) = uCG(u1) - {cg} ***if*** cg $\in$ cgAdmin(u1) ***then*** cgAdmin$'$(u1) = cgAdmin(u1) - {cg} ***forall*** o $\in$ O ***if*** origin(o) = cg ***then*** O$'$ = O - {o} ***forall*** o $\in$ O ***and forall*** v $\in$ versions(o). ***if*** cg $\in$ vMember$_o$(v) ***then*** vMember$'_o$(v) = vMember$_o$(v) - {cg} CG$'$ = CG - {cg} S$'$ = S - $\bigcup_{\forall s \in S.belongsTo(s)=cg}$S

TABLE III
GSIS-EXPEDIENT-INSIDER STATE TRANSITIONS AND QUERIES(PART 2: OPERATIONAL MODEL)

Op.#	Operation	Authorization Query	State Elements Update in State Transition
14.	**CreateRWInCG**(u,s,cg,sl,cp) /*User u creates read-write subject s in a group cg*/	$u \in U \wedge s \notin S \wedge cg \in \text{uCG}(u) \wedge$ $sl \preceq \text{hierclearanceOfUser}(u) \wedge$ $cp \subseteq \text{compcategoryOfUser}(u)$	$\text{owner}'(s) = u$ $\text{hierclearanceOfSubject}'(s) = sl$ $\text{belongsTo}'(s) = cg$ $\text{compcategoryOfSubject}(u)' = cp$ $\text{type}'\ (s) = RW$ $S' = S \cup \{s\}$
15.	**CreateRWInOrg**(u,s,sl,cp) /*Only true insider creates read-write subject in Org*/	$u \in U \wedge s \notin S \wedge \text{utype}(u) = \text{Insider}$ $\wedge\ sl \preceq \text{hierclearanceOfUser}(u) \wedge$ $cp \subseteq \text{compcategoryOfUser}(u)$	$\text{owner}'(s) = u$ $\text{hierclearanceOfSubject}'(s) = sl$ $\text{belongsTo}'(s) = cg$ $\text{compcategoryOfSubject}(u)' = cp$ $\text{type}'\ (s) = RW$ $S' = S \cup \{s\}$
16.	**CreateRO**(u,s,sl,cp) /*User u creates read-only subject s*/	$u \in U \wedge s \notin S \wedge$ $sl \preceq \text{hierclearanceOfUser}(u) \wedge$ $cp \subseteq \text{compcategoryOfUser}(u)$	$\text{owner}'(s) = u$ $\text{hierclearanceOfSubject}'(s) = sl$ $\text{type}'(s) = RO$ $\text{compcategoryOfSubject}(u)' = cp$ $S' = S \cup \{s\}$
17.	**Read**(s,o,v) /*Subject s reads the version v of object o*/	$s \in S \wedge o \in O \wedge v \in \text{versions}(o) \wedge$ $\text{hierclearanceOfSubject}(s) \succeq$ $\text{hierclassificationOfVersion}_o(v) \wedge$ $\text{compcategoryOfSubject}(s) \supseteq$ $\text{compcategoryOfVersion}_o(v) \wedge$ $(\text{type}(s) = RO \wedge$ $((\text{uCG}(\text{owner}(s)) \cap \text{vMember}_o(v) \neq \phi)$ $\vee\ (\text{utype}(\text{owner}(s)) = \text{Insider} \wedge$ $\{\text{Org}\} \in \text{vMember}_o(v)))) \vee$ $(\text{type}(s) = RW \wedge$ $(\text{belongsTo}(s) \in \text{vMember}_o(v))))$	None
18.	**Update**(s,o,v) /*Subject s updates the version v of object o. This function returns updated version v1*/	$s \in S \wedge o \in O \wedge v \in \text{versions}(o) \wedge$ $\text{hierclearanceOfSubject}(s) =$ $\text{hierclassificationOfVersion}_o(v) \wedge$ $\text{compcategoryOfSubject}(s) =$ $\text{compcategoryOfVersion}_o(v) \wedge$ $(\text{type}(s) = RW \wedge$ $\text{belongsTo}(s) \in \text{vMember}_o(v))$	$\text{versions}'(o) = \text{versions}(o) \cup \{v1\}$ $\text{vMember}'_o(v1) = \text{vMember}_o(v1) \cup \{cg\}$ $\text{hierclassificationOfVersion}'_o(v1) = \text{hierclassificationOfVersion}_o(v)$ $\text{compcategoryOfVersion}'_o(v1) = \text{compcategoryOfVersion}_o(v)$
19.	**Create**(s,o) /*Subject s creates version v of object o. This function returns newly created version v*/	$s \in S \wedge o \notin O \wedge \text{type}(s)=RW$	$O' = O \cup \{o\}$ $\text{versions}'(o) = \{v\}$ $\text{vMember}'_o(v) = \{\text{belongsTo}(s)\}$ $\text{origin}'(o) = \text{belongsTo}(s)$ $\text{hierclassificatonOfObject}'(o) = \text{hierclearanceOfSubject}(s)$ $\text{hierclassificatonOfVersion}'_o(v) = \text{hierclearanceOfSubject}(s)$ $\text{compcategoryOfVersion}'_o = \text{compcatgoryOfSubject}(s)$
20.	**Kill**(u,s) /*User u kills subject s*/	$u \in U \wedge s \in S \wedge$ $\text{owner}(s) = u \vee$ $\text{belongsTo}(s) \in \text{cgAdmin}(u)$	$\text{owner}' = \text{owner} - \{s \rightarrow \text{owner}(s)\}$ $\text{type}' = \text{type} - \{s \rightarrow \text{type}(s)\}$ $\text{hierclearanceOfSubject}' =$ $\text{hierclearanceOfSubject} - \{s \rightarrow \text{hierclearanceOfSubject}(s)\}$ $\text{compcategoryOfSubject}' = \text{compcategoryOfSubject} -$ $\{s \rightarrow \text{compcategoryOfSubject}(s)\}$ $\text{belongsTo}' = \text{belongsTo} - \{s \rightarrow \text{belongsTo}(s)\}$ $S' = S - \{s\}$

TABLE IV
ATTRIBUTE SPECIFICATION OF LBAC WITH COLLABORATIVE COMPARTMENTS

Element#	**Global Sets and Symbols:**
1.	$CC_\gamma \subset \mathcal{CC}$, is finite and strict subset of countably infinite set of unordered collaborative categories $\mathcal{CC}$
2.	C_γ = C, is finite set of existing unordered categories
3.	L_γ = L, is finite set of existing hierarchical ordered security levels
4.	SysHigh, SysHigh, the system High (constant label) that dominates every security labels $\in$ SL
5.	SysLow, the system Low (constant label) that is dominated by every security labels $\in$ SL
6.	$SL_\gamma \subset \mathcal{SL}$, is finite and strict subset of countably infinite security labels $\mathcal{SL}$ where $\mathcal{SL} = \{(L \times 2^{\mathcal{C}}) \times (\mathcal{CC} \cup \{Org\})\} \cup \{SysHigh, SysLow\}$
7.	$\succeq_\gamma \subset \succeq$, is finite and strict subset of countably infinite dominance relation $\succeq \subseteq \mathcal{SL} \times \mathcal{SL}$ where
	$\forall$l1,l2 $\in$ L and $\forall$c1,c2 $\in$ C and $\forall$cc1,cc2 $\in$ CC. $\succeq = \{((l1,c1,cc1), (l2,c2,cc2)) \mid cc1=cc2 \wedge l1 \succeq l2 \wedge c1 \supseteq c2\}$
	$\forall$l $\in$ L and $\forall$c $\in$ C and $\forall$cc $\in$ CC. $\succeq = \{(SysHigh, (l,c,cc)\}$
	$\forall$l $\in$ L and $\forall$c $\in$ C and $\forall$cc $\in$ CC. $\succeq = \{((l,c,cc),SysLow\}$
8.	$\oplus_\gamma = \oplus$, is join operator where
	$\forall$l1,l2 $\in$ L and $\forall$c1,c2 $\in$ C and $\forall$cc1,cc2 $\in$ CC.
	(l1,c1,cc1) $\oplus$ (l2,c2,cc2) = (max(l1,l2),c1$\cup$c2,cc1), if cc1=cc2
	(l1,c1,cc1) $\oplus$ (l2,c2,cc2)=SysHigh, if cc1$\neq$cc2
	$\forall$l $\in$ L and $\forall$c $\in$ C and $\forall$cc $\in$ CC.
	(l,c,cc) $\oplus$ SysHigh = SysHigh, Syshigh $\oplus$ (l,c,cc) = SysHigh
	(l,c,cc) $\oplus$ SysLow = (l,c,cc), SysLow $\oplus$ (l,c,cc) = (l,c,cc)
9.	$U_\gamma \subset \mathcal{U}$, is finite and strict subset of countably infinite set $\mathcal{U}$.
10.	$O_\gamma \subset \mathcal{O}$, is finite and strict subset of countably infinite set $\mathcal{O}$.
11.	$S_\gamma \subset \mathcal{S}$, is finite and strict subset of countably infinite set $\mathcal{S}$.
12.	$UTYPE_\gamma$ = UTYPE = {insider, expedient_insider, outsider} is the finite set of user's type
13.	$STYPE_\gamma$ = STYPE = {RO, RW} is the finite set of subject's type.
14.	Org, is the entity Organization, a Constant.
	User Related State Elements:
15.	hierclearanceOfUser: $U_\gamma \rightarrow L_\gamma$, this function maps each user to a security level.
16.	compcategoryOfUser: $U_\gamma \rightarrow 2^{C_\gamma}$, this function maps each user to compartments.
17.	uCC: $U_\gamma \rightarrow 2^{CC_\gamma}$, this function maps each user to zero or more collaborative compartments.
18.	orgAdmin: $U_\gamma \rightarrow$ {true,false}, this function maps each user to true if she is an admin of Org
19.	ccAdmin: $U_\gamma \rightarrow 2^{CC_\gamma}$, this function maps each user to zero or more groups if he is an administrative user of a collaboration group.
20.	uType: $U_\gamma \rightarrow UTYPE_\gamma$, this function maps each user to a user type.
	Objects Related State Elements:
21.	hierclassificationOfObject: $O_\gamma \rightarrow L_\gamma$, this function maps each object to a security levels.
22.	compcategoryOfObject: $O_\gamma \rightarrow C_\gamma$, this function maps each object to compartment.
23.	origin: $O_\gamma \rightarrow CC_\gamma \cup \{Org\}$, this function maps each object to the entity (collaboration category or Org) where it was created.
24.	versions: $O_\gamma \rightarrow 2^{Univ\_V}_{finite} - \phi$, this function maps each object to all its existing versions
	where $\mathcal{UNIV\_V}$ is countably infinite set of all possible versions
	/* $2^{Univ\_V}_{finite}$ is finite set of existing versions that is a subset of $\mathcal{UNIV\_V}$.*/
	Subject Related State Elements:
25.	hierclearanceOfSubject: $S_\gamma \rightarrow L_\gamma$, this function maps each subject to a security levels.
26.	compcategoryOfSubject: $S_\gamma \rightarrow C_\gamma$, this function maps each subject to compartment.
27.	owner: $S_\gamma \rightarrow U_\gamma$, this function maps each subject to the user who created this.
28.	belongsTo: $S_\gamma \hookrightarrow CC_\gamma$, this function maps each RW subject (not RO subject) to the collaboration category where it was created.
29.	type: $S_\gamma \rightarrow STYPE_\gamma$, this function maps each subject to a subject type.
	Object Version Related State Elements:
30.	For each o $\in O_\gamma$, vMember$_o$: versions(o)$\rightarrow 2^{CC_\gamma \cup \{Org\}} - \phi$, this functions maps each version of every object to one or more entity (collab. category or Org) where this version is available to access.
31.	For each o $\in O_\gamma$, hierclassificationOfVersion$_o$: versions(o)$\rightarrow L_\gamma$, this function maps each version to a security levels.
32.	For each o $\in O_\gamma$, compcategoryOfVersion$_o$: versions(o) $\rightarrow 2^{C_\gamma}$ this function maps each subject to compartment.

TABLE V
STATE TRANSITION AND QUERY OF LBAC WITH COLLABORATIVE COMPARTMENTS(PART 1: ADMIN MODEL)

Op.#	Operation	Authorization Query	State Element Update on State Transition
1.	**Create_Insider**(u1,u2,uType,sl,cp) /*Admin u1 creates user u2 as insider*/	u1 $\in$ U $\wedge$ u2 $\notin$ U $\wedge$ orgAdmin(u1)=True$\wedge$ s1 $\in$ L $\wedge$cp $\subseteq$ C$\wedge$ uType=Insider	***if*** uType=Insider ***then*** hierclearanceOfUser$'$(u2)=sl compcategoryOfUser$'$(u2)=cp uType(u2)$'$=Insider U$'$ = U $\cup$ {u2}
2.	**Create_OutSider**(u1,u2,uType,sl,cp) /*Admin u1 creates user u2 as outsider*/	u1 $\in$ U $\wedge$ u2 $\notin$ U $\wedge$ orgAdmin(u1)=True$\wedge$ S1 $\in$ L $\wedge$ cp $\subseteq$ C$\wedge$ uType=Outsider	uType(u2)$'$=Outsider U$'$ = U $\cup$ {u2}

3.	**Delete_User**(u1,u2)
	/*Admin u1 creates user
	u2 as outsider*/
		u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$
		orgAdmin(u1)=True$\wedge$ S1 $\in$ L
		$\wedge$ cp $\subseteq$ C
			if(utype(u2)=Insider) then
			forall s $\in$ S
			if(owner(s)=u2)
			owner$'$ = owner-{s$\rightarrow$owner(s)}
			S$'$ = S-{s}
			uType$'$ = uType - { u2 $\rightarrow$ uType(u2)}
			U$'$ = U - {u2}

4.	**Establish**(u, cc) /*Admin user u establishes new collab compartment cc*/	u $\in$ U $\wedge$ cc $\notin$ CC $\wedge$ orgAdmin(u)=True	ccAdmin$'$(u) = ccAdmin(u) $\cup$ {cc} CC$'$ = CC $\cup$ {cc} SL' = {(L $\times$ 2^{C}).(CC $\cup$ {Org})} $\cup$ {SysHigh} $\cup$ {SysLow} $\forall$l1,l2 $\in$ L and $\forall$c1,c2 $\in$ C and $\forall$cc1,cc2 $\in$ CC. $\succeq'$ = {((l1,c1,cc1), (l2,c2,cc2)) | cc1=cc2 $\wedge$ l1$\succeq$ l2 $\wedge$ c1 $\supseteq$ c2}
5.	**Add_Clearance**(u1,u2,cc) /*Admin u1 grants cc clearance to a true insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cc $\in$ CC $\wedge$ cc $\in$ ccAdmin(u1) $\wedge$ uType(u2) = Insider $\wedge$ cc $\notin$ uCC(u2)	uCC$'$(u2) = uCC(u2) $\cup$ {cc}
6.	**Remove_Clearance**(u1,u2,cc) /*Admin u1 revokes cc membership from a true insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cc $\in$ CC $\wedge$ cc $\in$ ccAdmin(u1) $\wedge$ cc $\in$ uCC(u2) $\wedge$ uType(u2) = Insider	uCC$'$(u2) = uCC(u2) - {cc} ***forall*** s $\in$ S ***if*** owner(s) = u2 $\wedge$ belongsTo(s) = cc ***then*** S$'$ = S - {s}
7.	**Join_Outsider**(u1,u2,cc,sl,cp) /*Admin u1 grants cc membership to an expedient insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cc $\in$ CC $\wedge$ cc $\in$ ccAdmin(u1) $\wedge$ cc $\notin$ uCC(u2) $\wedge$ uType(u2)= Outsider $\wedge$ s1 $\in$ L $\wedge$ cp $\subseteq$ C	uType$'$(u2) = Expedient_Insider ***if*** uCC(u2) = $\emptyset$ ***then*** hierclearanceOfUser$'$(u2) = sl compcategoryOfUser$'$(u2) = cp uCC$'$(u2) = uCC(u2) $\cup$ {cc}
8.	**Leave_Expedient_Insider**(u1,u2,cc) /*Admin u1 revokes cc membership from an expedient insider u2*/	u1 $\in$ U $\wedge$ u2 $\in$ U $\wedge$ cc $\in$ CC $\wedge$ cc $\in$ ccAdmin(u1) $\wedge$ cc $\in$ uCC(u2) $\wedge$ uType(u2) = Expedient_Insider	uCC$'$(u2) = uCC(u2) - {cc} ***forall*** s $\in$ S ***if*** owner(s) = u2 $\wedge$ belongsTo(s) = cc ***then*** S$'$ = S - {s} /*Kill subjects belongsTo the respective insider*/ ***if*** uCC(u2) = $\emptyset$ ***then*** hierclearanceOfUser$'$ = hierclearanceOfUser-{u2$\rightarrow$hierclearanceOfUser(u2)} compcategoryOfUser$'$ = compcategoryOfUser-{u2$\rightarrow$compcategoryOfUser(u2)} uType(u2) = Outsider
9.	**Add**(u,o,v,cc) /*Admin u adds version v of object o from Org to cc*/	u $\in$ U $\wedge$ cc $\in$ CC $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cc $\in$ ccAdmin(u) $\wedge$ cc $\notin$ vMember$_o$(v)	vMember$'_o$(v) = vMember$_o$(v) $\cup$ {cc}
10.	**Remove**(u,o,v,cc) /*Admin u removes version v of object o from cc*/	u $\in$ U $\wedge$ cc $\in$ CC $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cc $\in$ ccAdmin(u) $\wedge$ cc $\in$ vMember$_o$(v)	vMember$'_o$(v) = vMember$_o$(v) - {cc}
11.	**Import**(u,o1,v1,o2,cc) /*Admin u imports version v1 of object o1 to new version v2 of object o2 in Org*/	u $\in$ U $\wedge$ cc $\in$ CC $\wedge$ v1 $\in$ versions(o) $\wedge$ o1,o2 $\in$ O $\wedge$ origin(o2) = Org $\wedge$ cc $\in$ ccAdmin(u) $\wedge$ origin(o1) = cc $\wedge$hierclassificationOfObject(o1) = hierclassificationOfObject(o2) $\wedge$ compcategoryOfObject(o2) $\supseteq$ compcategoryOfObject(o1)	versions$'$(o2) = versions(o2) $\cup$ {v2} vMember$'$(o2,v2) = {Org} hierclassificationOfVersion$_{o2}$(v2) = hierclassificationOfObject(o2) compcategoryOfVersion$_{o2}$(v2)=compcategoryOfObject(o2)
12.	**Merge**(u,o,v,cc) /*Admin u merges version v of object o from cc to Org*/	u $\in$ U $\wedge$ cc $\in$ CC $\wedge$ o $\in$ O $\wedge$ v $\in$ versions(o) $\wedge$ cc $\in$ ccAdmin(u) $\wedge$cc $\in$ vMember$_o$(v) $\wedge$ origin(o) = Org $\wedge$ v $\in$ versions(o)	vMember$'_o$(v) = vMember$_o$(v) $\cup$ {Org}
13.	**Disband**(u, cc) /*Admin u disbands a collaboration group cc*/	u $\in$ U $\wedge$ cc $\in$ CC $\wedge$ cc $\in$ ccAdmin(u)	***forall*** u1 $\in$ U ***if*** cc $\in$ uCC(u1) ***then*** uCC$'$(u1) = uCC(u1) - {cc} ***if*** cc $\in$ ccAdmin(u1) ***then*** ccAdmin$'$(u1) = ccAdmin(u1) - {cc} ***forall*** o $\in$ O ***if*** origin(o) = cc ***then*** O$'$ = O - {o} ***forall*** o $\in$ O ***and forall*** v $\in$ versions(o). ***if*** cc $\in$ vMember$_o$(v) ***then*** vMember$'_o$(v) = vMember$_o$(v) - {cc} CC$'$ = CC - {cc} S$'$ = S - $\bigcup_{\forall s \in S.belongsTo(s)=cc}$S

TABLE VI
STATE TRANSITION AND QUERIES OF LBAC WITH COLLABORATIVE COMPARTMENTS (PART 2: OPERATIONAL MODEL)

Op.#	Operation	Authorization Query	State Elements Update in State Transition
14.	**CreateRWInCG**(u,s,cc,sl,cp) /*User u creates read-write subject s in a group cc*/	$u \in U \wedge s \notin S \wedge cc \in uCC(u) \wedge$ $sl \preceq$ hierclearanceOfUser(u) $\wedge$ $cp \subseteq$ compcategoryOfUser(u)	owner$'$(s) = u hierclearanceOfSubject$'$(s) = sl belongsTo$'$(s) = cc compcategoryOfSubject(u)$'$ = cp type$'$ (s) = RW $S' = S \cup \{s\}$
15.	**CreateRWInOrg**(u,s,sl,cp) /*Only true insider creates read-write subject in Org*/	$u \in U \wedge s \notin S \wedge$ utype(u) = Insider $\wedge$ $sl \preceq$ hierclearanceOfUser(u) $\wedge$ $cp \subseteq$ compcategoryOfUser(u)	owner$'$(s) = u hierclearanceOfSubject$'$(s) = sl belongsTo$'$(s) = cc compcategoryOfSubject(u)$'$ = cp type$'$ (s) = RW $S' = S \cup \{s\}$
16.	**CreateRO**(u,s,sl,cp) /*User u creates read-only subject s*/	$u \in U \wedge s \notin S \wedge$ $sl \preceq$ hierclearanceOfUser(u) $\wedge$ $cp \subseteq$ compcategoryOfUser(u)	owner$'$(s) = u hierclearanceOfSubject$'$(s) = sl type$'$(s) = RO compcategoryOfSubject(u)$'$ = cp $S' = S \cup \{s\}$
17.	**Read**(s,o,v) /*Subject s reads the version v of object o*/	$s \in S \wedge o \in O \wedge v \in$ versions(o) $\wedge$ hierclearanceOfSubject(s) $\succeq$ hierclassificationOfVersion$_o$(v) $\wedge$ compcategoryOfSubject(s) $\supseteq$ compcategoryOfVersion$_o$(v)$\wedge$ (type(s) = RO $\wedge$ ((uCC(owner(s)) $\cap$ vMember$_o$(v)$\neq \phi$) $\vee$ (utype(owner(s)) = Insider $\wedge$ $\{Org\} \in$ vMember$_o$(v)))) $\vee$ (type(s) = RW $\wedge$ (belongsTo(s) $\in$ vMember$_o$(v))))	None
18.	**Update**(s,o,v) /*Subject s updates the version v of object o. This function returns updated version v1*/	$s \in S \wedge o \in O \wedge v \in$ versions(o) $\wedge$ hierclearanceOfSubject(s) = hierclassificationOfVersion$_o$(v) $\wedge$ compcategoryOfSubject(s) = compcategoryOfVersion$_o$(v)$\wedge$ (type(s) = RW $\wedge$ belongsTo(s) $\in$ vMember$_o$(v))	versions$'$(o) = versions(o) $\cup$ {v1} vMember$'_o$(v1) = vMember$_o$(v1) $\cup$ {cc} hierclassificationOfVersion$'_o$(v1) = hierclassificationOfVersion$_o$(v) compcategoryOfVersion$'_o$(v1) = compcategoryOfVersion$_o$(v)
19.	**Create**(s,o) /*Subject s creates version v of object o. This function returns newly created version v*/	$s \in S \wedge o \notin O \wedge$ type(s)=RW	$O' = O \cup \{o\}$ versions$'$(o) = {v} vMember$'_o$(v) = {belongsTo(s)} origin$'$(o) = belongsTo(s) hierclassificatonOfObject$'$(o) = hierclearanceOfSubject(s) hierclassificatonOfVersion$'_o$(v) = hierclearanceOfSubject(s) compcategoryOfVersion$'_o$ = compcatgoryOfSubject(s)
20.	**Kill**(u,s) /*User u kills subject s*/	$u \in U \wedge s \in S \wedge$ owner(s) = u $\vee$ belongsTo(s) $\in$ ccAdmin(u)	owner$'$ = owner - {s $\rightarrow$ owner(s)} type$'$ = type - {s $\rightarrow$ type(s)} hierclearanceOfSubject$'$ = hierclearanceOfSubject - {s $\rightarrow$ hierclearanceOfSubject(s)} compcategoryOfSubject$'$ = compcategoryOfSubject - {s $\rightarrow$ compcategoryOfSubject(s)} belongsTo$'$ = belongsTo - {s $\rightarrow$ belongsTo(s)} $S' = S - \{s\}$

IV. Mapping from LCC to GEI

Let, γ^{LCC} is the state of LCC scheme where state elements are given in Table IV, ψ^{LCC} are state-change rules that given in column 1 of table V and table VI and Q^{LCC} is the set of authorization queries as mentioned in column 2 of table V and table VI. σ is a mapping that produces output $\langle\gamma^{GEI}, \psi^{GEI}\rangle$ for each input $\langle\gamma^{LCC}, \psi^{LCC}\rangle$ and q^{GEI} for each $q^{LCC} \in Q^{LCC}$. Here, γ^{GEI} is state of GEI scheme given in Table I, ψ^{GEI} is the state-change rule that given in column 1 of table II and Q^{GEI} are queries given in Column 2 of Table II.

1. σ mapping of γ^{LCC} to γ^{GEI}

- σ provides one-to-one mapping from Element# 1,2,3,9,10,11,12,13,14 of Table IV to Element# 1,2,3,7,8,9,10,11,12 of Table I.
- For Element# 4, SL_γ^{GEI}=L_γ^{LCC}∗$2^{C_\gamma^{LCC}}$
- For Element# 5, $\succeq_\gamma^{GEI}$ = $SL_\gamma^{GEI} \times SL_\gamma^{GEI}$ where, $\forall$l1,l2 $\in L_\gamma^{LCC}$ and $\forall$c1,c2 $\in C_\gamma^{LCC}$. $\succeq$ = $\{$((l1,c1), (l2,c2)) $|$ l1$\succeq$ l2 $\wedge$ c1 $\supseteq$ c2$\}$
- For Element# 6, $\oplus_\gamma^{GEI}$ = (l1,c1) $\oplus$ (l2,c2) = (max(l1,l2),c1$\cup$c2) where l1,l2$\in L_\gamma^{LCC}$ and c1,c2 $\in C_\gamma^{LCC}$
- σ provides one-to-one mapping from Element# 13-30 of Table IV to Element# 15-32 of Table I.

2. σ mapping of ψ^{LCC} to ψ^{GEI}

The ψ^{GEI} is the set of operations that is given in Column 1 of Table II and III can be mapped from each corresponding operations given in Column 1 of Table V and VI. For example, Op# 5 Join_Insider of Table II is mapped from Op# 5 Add_Clearance of Table V.

3. σ mapping of Q^{LCC} to Q^{GEI}

Finally, the authorization queries in Q^{LCC} are mapped to the corresponding queries given in Column 2 of Table II and III. Note that, Q^{LCC} is the set of queries given in Column 2 of Table V and VI. For example, if the q^{LCC} is the query given in row 1,column 2 of Table V then it can be mapped with the q^{GEI} which is the query of row 1, column 2 of Table II.

V. Proof of State Matching Reduction from LCC to GEI

Lemma 1. *The mapping from LCC to GEI defined in section IV satisfies property 1 of Definition 1.*

Proof: According to property 1 of definition 7 of definition 1 for every state $\gamma^{LCC} \in \Gamma^{LCC}$ and every $\psi^{LCC} \in \Psi^{LCC}$, $\langle\gamma^{GEI}, \psi^{GEI}\rangle = \sigma(\langle\gamma^{LCC}, \psi^{LCC}\rangle)$ has the following property :
For every state γ_1^{LCC} in scheme LCC such that $\gamma^{LCC} \xrightarrow{*}_\psi \gamma_1^{LCC}$, there exists a state γ_1^{GEI} in scheme GEI such that,

I) $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$
II) for every query $q^{LCC} \in Q^{LCC}$, $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ if and only if $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$.

II can be decomposed into two directions:

II.a) The "if" direction:
$\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC}) => \gamma_1^{LCC} \vdash^{LCC} q^{LCC}$.
II.b) The "only if" direction:
$\gamma_1^{LCC} \vdash^{LCC} q^{LCC} => \gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$.

Proof By Induction: Induction on n steps in $\gamma^{LCC} \xrightarrow{n}_{\psi} \gamma_1^{LCC}$.

Base Case: Let n=0.

I):$\gamma^{LCC} = \gamma_1^{LCC}$ and $\gamma^{GEI} = \gamma_1^{GEI}$
Thus, $\sigma(\gamma_1^{LCC})$ =$\sigma(\gamma^{LCC})$ = γ^{GEI} = γ_1^{GEI}.
So, $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$.
Therefore, we can say that I of assertion 1 holds for basis case.

II.a): If γ_1^{LCC} = γ^{LCC} and $\gamma^{LCC} \vdash^{LCC} q^{LCC}$
then $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$
Again, If $\sigma(\gamma^{LCC}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{LCC}) \mapsto Q^{GEI}$ and $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$
Finally, as γ_1^{LCC} = γ^{LCC} and γ_1^{GEI} = γ^{GEI} we can say, If $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$.

II.b): If $\sigma(\gamma^{LCC}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{LCC}) \mapsto Q^{GEI}$ and $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ then $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$.
Therefore, as γ_1^{LCC} = γ^{LCC} and γ_1^{GEI} = γ^{GEI} we can say, If $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ then $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$.
Thus, II of property 1 holds for base case.

Inductive Hypothesis: Property 1 holds for n = k.

Inductive Steps: Let, n=k+1.

I):$\gamma^{LCC} \xrightarrow{k}_{\psi} \gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$
According to the inductive hypothesis there exists,
$\gamma^{GEI} \xrightarrow{k}_{\psi^{GEI}} \gamma_k^{GEI}$ for $\gamma^{LCC} \xrightarrow{k}_{\psi^{LCC}} \gamma_k^{LCC}$

In order to prove I of property 1 we need to prove that there exists,
$\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$ for $\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$

We have shown in section IV, for every $\psi^{LCC} \in \Psi^{LCC}$ and $\psi^{GEI} \in \Psi^{GEI}$ and $q^{LCC} \in Q^{LCC}$ and $q^{GEI} \in Q^{GEI}$ there exists $\sigma(\psi^{LCC}) \mapsto \psi^{GEI}$ and $\sigma(q^{LCC}) \mapsto q^{GEI}$. So we can say that, for every $\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$ there exists
$\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$
Therefore property I holds.

II.a): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC}) => \gamma_1^{GEI} \vdash^{GEI} q^{LCC}$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{LCC} \in \Psi^{LCC}$ has a corresponding $\sigma(\psi^{LCC}) = \psi^{GEI} \in \Psi^{GEI}$ so we can say that II.a of property 1 holds.

II.b): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} q^{LCC} => \gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{LCC} \in \Psi^{LCC}$ has a corresponding $\sigma(\psi^{LCC}) = \psi^{GEI} \in \Psi^{GEI}$ so we can say that II.b of property 1 holds.

■

Lemma 2. *The mapping from LCC to GEI defined in section IV satisfies property 2 of Definition 1.*

Proof: According to property 2 of definition 7 of definition 1 for every state $\gamma^{LCC} \in \Gamma^{LCC}$ and every $\psi^{LCC} \in \Psi^{LCC}$, $\langle \gamma^{GEI}, \psi^{GEI} \rangle = \sigma(\langle \gamma^{LCC}, \psi^{LCC} \rangle)$ has the following property :
For every state γ_1^{GEI} in scheme GEI such that $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$, there exists a state γ_1^{LCC} in scheme LCC such that,

I) $\gamma^{LCC} \xrightarrow{*}_{\psi} \gamma_1^{LCC}$

II) for every query $q^{LCC} \in Q^{LCC}$, $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ if and only if $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$.

II can be decomposed into two directions:

II.a) The "if" direction:
$\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC}) => \gamma_1^{LCC} \vdash^{LCC} q^{LCC}$.
II.b) The "only if" direction:
$\gamma_1^{LCC} \vdash^{LCC} q^{LCC} => \gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$

Proof By Induction: Induction on n steps in $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$

Base Case: Let n=0.

I): $\gamma^{GEI} = \gamma_1^{GEI}$ and $\gamma^{LCC} = \gamma_1^{LCC}$
Thus, $\gamma_1^{GEI} = \gamma^{GEI} = \sigma(\gamma^{LCC}) = \sigma(\gamma_1^{LCC})$.
So, $\gamma^{LCC} \xrightarrow{*}_{\psi} \gamma_1^{LCC}$.
Therefore, we can say that I of assertion 1 holds for basis case.

II.a): If $\gamma_1^{LCC} = \gamma^{LCC}$ and $\gamma^{LCC} \vdash^{LCC} q^{LCC}$
then $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$

Again, If $\sigma(\gamma^{LCC}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{LCC}) \mapsto Q^{GEI}$ and $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$

Finally, as $\gamma_1^{LCC} = \gamma^{LCC}$ and $\gamma_1^{GEI} = \gamma^{GEI}$ we can say, If $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$.

II.b): If $\sigma(\gamma^{LCC}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{LCC}) \mapsto Q^{GEI}$
and $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ then $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$.

Therefore, as $\gamma_1^{LCC} = \gamma^{LCC}$ and $\gamma_1^{GEI} = \gamma^{GEI}$ we can say, If $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ then $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$.
Thus, II of property 1 holds for base case.

Inductive Hypothesis: Property 1 holds for n = k.

Inductive Steps: Let, n=k+1.

I):$\gamma^{GEI} \xrightarrow{k}_{\psi^{GEI}} \gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$
According to the inductive hypothesis there exists,
$\gamma^{LCC} \xrightarrow{k}_{\psi^{LCC}} \gamma_k^{LCC}$ for $\gamma^{GEI} \xrightarrow{k}_{\psi^{GEI}} \gamma_k^{GEI}$
In order to prove I of property 1 we need to prove that there exists,
$\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$ for $\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$

We have shown in section IV, for every $\psi^{LCC} \in \Psi^{LCC}$ and $\psi^{GEI} \in \Psi^{GEI}$ and $q^{LCC} \in Q^{LCC}$ and $q^{GEI} \in Q^{GEI}$ there exists $\sigma(\psi^{LCC}) \mapsto \psi^{GEI}$ and $\sigma(q^{LCC}) \mapsto q^{GEI}$. So we can say that, for every $\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$ there exists
$\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$
Therefore property I holds.

II.a): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC}) => \gamma_1^{GEI} \vdash^{GEI} q^{LCC}$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{LCC} \in \Psi^{LCC}$ has a corresponding $\sigma(\psi^{LCC}) = \psi^{GEI} \in \Psi^{GEI}$ so we can say that II.a of property 1 holds.

II.b): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} q^{LCC} => \gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{LCC} \in \Psi^{LCC}$ has a corresponding $\sigma(\psi^{LCC}) = \psi^{GEI} \in \Psi^{GEI}$ so we can say that II.b of property 1 holds.

■

VI. Mapping from GEI to LCC

In LCC scheme SysLow and SysHigh are two security labels unpopulated with no objects, subjects, object version or user for the following reasons:

1) Add_Clearance and Join_Outsider operations do not assign any user these two clearances
2) By CreateRO, CreateRWInOrg or CreateRWInCG operations user can only assign any of his clearance to newly create subject those are not SysLow or SysHigh
3) Create operation only create object with subject's security clearance
4) All object versions share same security classification. Merge, Import, Update operation do not change them.

For this reason, during mapping from GEI to LCC we do not consider SysLow or SysHigh. However, they are necessary to create a lattice in LCC .

Again, all object versions share same classification, there is no application of $\oplus_\gamma$ in the system.

Let, γ^{GEI} is the state of GEI scheme where state elements are given in Table I , ψ^{GEI} are state-change rules that given in column 1 of table II and Q^{GEI} is the set of authorization queries as mentioned in column 2 of table II. σ is a mapping that produces output $\langle\gamma^{LCC}, \psi^{LCC}\rangle$ for each input $\langle\gamma^{GEI}, \psi^{GEI}\rangle$ and q^{LCC} for each q$^{GEI} \in$ Q^{GEI}. Here, γ^{LCC} is state of LCC scheme given in Table IV, ψ^{LCC} is the state-change rule that given in column 1 of table V and VI and Q^{LCC} are queries given in Column 2 of Table V and table VI.

1. σ **mapping of** γ^{GEI} to γ^{LCC}

- σ provides one-to-one mapping from Element# 1,2,3,7,8,9,10,11,12 of Table I to Element# 1,2,3,9,10,11,12,13,14 of Table IV.
- For Element# 6, SL_{γ}^{LCC}=(SL_{γ}^{GEI}).$(CG \cup Org)$
- For Element# 7, $\succeq_{\gamma}^{LCC}$ = $SL_{\gamma}^{LCC} \times SL_{\gamma}^{LCC}$ where, ∀l1,l2 ∈ SL$_{\gamma}^{GEI}$ and ∀c1,c2 ∈ C$_{\gamma}^{GEI}$ ∀cg1,cg2 ∈ CG_{γ}^{GEI}. $\succeq$ = {((l1,c1,cg1), (l2,c2,cg2)) | l1$\succeq$ l2 ∧ c1 ⊇ c2 ∧ $cg1 = cg2$}
- For Element# 8, $\oplus_{\gamma}^{LCC}$ = (l1,c1,cg1) ⊕ (l2,c2,cg2) = (max(l1,l2),c1∪c2,$cg1$) if $cg1$ = $cg2$ where l1,l2∈ L$_{\gamma}^{GEI}$ and c1,c2 ∈ C$_{\gamma}^{GEI}$ and cg1, cg2 ∈ CG$_{\gamma}^{GEI}$
- σ provides one-to-one mapping from Element# Element# 15-32 of Table I to 13-30 of Table IV.

2. σ **mapping of** ψ^{GEI} **to** ψ^{LCC}

The ψ^{GEI} is the set of state transition rules that is given in Column 1 of Table V and VI can be mapped from each corresponding operations given in Column of 1 of Table II and III. For example, Op# 5 Add_Clearance of Table V is mapped from Op# 5 Join_Insider of Table II .

3. σ **mapping of** Q^{GEI} **to** Q^{LCC}

Finally, the authorization queries in Q^{GEI} are mapped to the corresponding queries given in Column 2 of Table V and VI. Note that, Q^{GEI} is the set of queries given in Column 2 of Table II and III. For example, if the q^{GEI} is the query given in row 1, column 2 of Table II then it can be mapped with the q^{LCC} which is the query of row 1,column 2 of Table V.

VII. Proof of State Matching Reduction from GEI to LCC

Lemma 3. *The mapping from GEI to LCC defined in section VI satisfies property 1 of Definition 1.*

Proof: According to property 1 of definition 7 of definition 1 for every state $\gamma^{GEI} \in \Gamma^{GEI}$ and every $\psi^{GEI} \in \Psi^{GEI}$, $\langle\gamma^{LCC}, \psi^{LCC}\rangle = \sigma(\langle\gamma^{GEI}, \psi^{GEI}\rangle)$ has the following property :
For every state γ_1^{GEI} in scheme GEI such that $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$, there exists a state γ_1^{LCC} in scheme LCC such that,

I) $\gamma^{LCC} \xrightarrow{*}_{\psi^{LCC}(=\sigma(\psi^{GEI})} \gamma_1^{LCC}$

II) for every query $q^{GEI} \in Q^{GEI}$, $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ if and only if $\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$.

II can be decomposed into two directions:

II.a) The "if" direction:
$\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI}) => \gamma_1^{GEI} \vdash^{GEI} q^{GEI}$
II.b) The "only if" direction:
$\gamma_1^{GEI} \vdash^{GEI} q^{GEI} => \gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$.

Proof By Induction: Induction on n steps in $\gamma^{GEI} \xrightarrow{n}_{\psi} \gamma_1^{GEI}$.

Base Case: Let n=0.

I):$\gamma^{GEI} = \gamma_1^{GEI}$
Thus, $\sigma(\gamma_1^{GEI}) = \sigma(\gamma^{GEI}) = \gamma^{LCC} = \gamma_1^{LCC}$.
So, $\gamma^{LCC} \xrightarrow{*}_{\psi^{LCC}} \gamma_1^{LCC}$.
Therefore, we can say that I of assertion 1 holds for basis case.

II.a): If $\gamma_1^{GEI} = \gamma^{GEI}$ and $\gamma^{GEI} \vdash^{GEI} q^{GEI}$
then $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ for every $q^{GEI} \in Q^{GEI}$

Again, If $\sigma(\gamma^{GEI}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{GEI}) \mapsto Q^{GEI}$ and $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$

Finally, as $\gamma_1^{LCC} = \gamma^{LCC}$ and $\gamma_1^{GEI} = \gamma^{GEI}$ we can say, If $\gamma_1^{LCC} \vdash^{LCC} q^{LCC}$ then $\gamma_1^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ for every $q^{LCC} \in Q^{LCC}$.

II.b): If $\sigma(\gamma^{LCC}) \mapsto \gamma^{GEI}$ and $\sigma(Q^{LCC}) \mapsto Q^{GEI}$
and $\gamma^{GEI} \vdash^{GEI} \sigma(q^{LCC})$ then $\gamma^{LCC} \vdash^{LCC} q^{LCC}$ for every $q^{LCC} \in Q^{LCC}$.

Therefore, as $\gamma_1^{LCC} = \gamma^{LCC}$ and $\gamma_1^{GEI} = \gamma^{GEI}$ we can say, If $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ then $\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$ for every $q^{GEI} \in Q^{GEI}$.
Thus, II of property 1 holds for base case.

Inductive Hypothesis: Property 1 holds for n = k.

Inductive Steps: Let, n=k+1.

I):$\gamma^{GEI} \xrightarrow{k}_{\psi} \gamma_k^{GEI} \xrightarrow{1}_{\psi} \gamma_1^{GEI}$
According to the inductive hypothesis there exists,
If $\gamma^{GEI} \xrightarrow{k}_{\psi^{GEI}} \gamma_k^{GEI}$ then $\gamma^{LCC} \xrightarrow{k}_{\psi^{LCC}} \gamma_k^{LCC}$
In order to prove I of property 1 we need to prove that there exists,
$\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$ for $\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$

We have shown in section VI, for every $\psi^{LCC} \in \Psi^{LCC}$ and $\psi^{GEI} \in \Psi^{GEI}$ and $q^{LCC} \in Q^{LCC}$ and $q^{GEI} \in Q^{GEI}$ there exists $\sigma(\psi^{GEI}) \mapsto \psi^{LCC}$ and $\sigma(q^{GEI}) \mapsto q^{LCC}$. So we can say that, for every $\gamma_k^{GEI} \xrightarrow{1}_{\psi} \gamma_1^{GEI}$ there exists
$\gamma_k^{LCC} \xrightarrow{1}_{\psi^{LCC}} \gamma_1^{LCC}$
Therefore property I holds.

II.a): We need to prove that,
$\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI}) => \gamma_1^{GEI} \vdash^{GEI} q^{GEI}$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{GEI} \in \Psi^{GEI}$ has a corresponding $\sigma(\psi^{GEI}) = \psi^{LCC} \in \Psi^{LCC}$ so we can say that II.a of property 1 holds.

II.b): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} q^{GEI} => \gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$

As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{GEI} \in \Psi^{GEI}$ has a corresponding $\sigma(\psi^{GEI})$ = $\psi^{LCC} \in \Psi^{LCC}$ so we can say that II.b of property 1 holds.

■

Lemma 4. *The mapping from GEI to LCC defined in section VI satisfies property 2 of Definition 1.*

Proof: According to property 2 of definition 7 of definition 1 for every state $\gamma^{GEI} \in \Gamma^{GEI}$ and every $\psi^{GEI} \in \Psi^{GEI}$, $\langle\gamma^{LCC}, \psi^{LCC}\rangle$ = $\sigma(\langle\gamma^{GEI}, \psi^{GEI}\rangle)$ has the following property :
For every state γ_1^{LCC} in scheme LCC such that $\gamma^{LCC} \xrightarrow{*}{}_{\psi}^{LCC}$ $(= \sigma(\psi^{GEI}))\gamma_1^{LCC}$, there exists a state γ_1^{GEI} in scheme GEI such that,

I) $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$

II) for every query $q^{GEI} \in Q^{GEI}$, $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ if and only if $\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$.

II can be decomposed into two directions:

II.a) The “if” direction:
$\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI}) => \gamma_1^{GEI} \vdash^{GEI} q^{GEI}$
II.b) The “only if” direction:
$\gamma_1^{GEI} \vdash^{GEI} q^{GEI} => \gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$.

Proof By Induction: Induction on n steps in $\gamma^{LCC} \xrightarrow{n}_{\psi^{LCC}} \gamma_1^{LCC}$.

Base Case: Let n=0.

I):$\gamma^{LCC} = \gamma_1^{LCC}$ and $\gamma^{LCC} = \sigma(\gamma^{GEI})$
Thus, $\sigma(\gamma_1^{GEI})$ = $\sigma(\gamma^{GEI})$ = γ^{LCC} = γ_1^{LCC}.
So, $\gamma^{GEI} \xrightarrow{*}_{\psi^{GEI}} \gamma_1^{GEI}$.
Therefore, we can say that I of assertion 1 holds for basis case.

II.a): If γ_1^{GEI} = γ^{GEI} and $\gamma^{GEI} \vdash^{GEI} q^{GEI}$
then $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ for every $q^{GEI} \in Q^{GEI}$

Again, If $\sigma(\gamma^{GEI}) \mapsto \gamma^{LCC}$ and $\sigma(Q^{GEI}) \mapsto Q^{LCC}$ and then $\gamma^{GEI} \vdash^{GEI} q^{GEI}$ $\gamma^{LCC} \vdash^{LCC} \sigma(q^{GEI})$ for every $q^{GEI} \in Q^{GEI}$

Finally, as γ_1^{LCC} = γ^{LCC} and γ_1^{GEI} = γ^{GEI} we can say, If $\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$ then $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ for every $q^{GEI} \in Q^{GEI}$.

II.b): If $\sigma(\gamma^{GEI}) \mapsto \gamma^{LCC}$ and $\sigma(Q^{GEI}) \mapsto Q^{LCC}$
and $\gamma^{GEI} \vdash^{GEI} q^{GEI}$ then $\gamma^{LCC} \vdash^{LCC} \sigma(q^{GEI})$ for every $q^{GEI} \in Q^{GEI}$.

Therefore, as γ_1^{LCC} = γ^{LCC} and γ_1^{GEI} = γ^{GEI} we can say, If $\gamma_1^{GEI} \vdash^{GEI} q^{GEI}$ then $\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$ for every $q^{GEI} \in Q^{GEI}$.
Thus, II.b of property 2 holds for base case.

Inductive Hypothesis: Property 1 holds for n = k.

Inductive Steps: Let, n=k+1.

I):$\gamma^{GEI} \xrightarrow{k}_{\psi} \gamma_k^{GEI} \xrightarrow{1}_{\psi} \gamma_1^{GEI}$
According to the inductive hypothesis there exists,
$\gamma^{LCC} \xrightarrow{k}_{\psi^{LCC}} \gamma_k^{LCC}$ for $\gamma^{GEI} \xrightarrow{k}_{\psi^{GEI}} \gamma_k^{GEI}$
In order to prove I of property 1 we need to prove that there exists,
$\gamma_k^{LCC} \xrightarrow{1}_{\psi} \gamma_1^{LCC}$ for $\gamma_k^{GEI} \xrightarrow{1}_{\psi^{GEI}} \gamma_1^{GEI}$

We have shown in section VI, for every $\psi^{LCC} \in \Psi^{LCC}$ and $\psi^{GEI} \in \Psi^{GEI}$ and $q^{LCC} \in Q^{LCC}$ and $q^{GEI} \in Q^{GEI}$ there exists $\sigma(\psi^{GEI}) \mapsto \psi^{LCC}$ and $\sigma(q^{LCC}) \mapsto q^{GEI}$. So we can say that, for every $\gamma_k^{GEI} \xrightarrow{1}_{\psi} \gamma_1^{GEI}$ there exists
$\gamma_k^{LCC} \xrightarrow{1}_{\psi^{LCC}} \gamma_1^{LCC}$
Therefore property I holds.

II.a): We need to prove that,
$\gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI}) => \gamma_1^{GEI} \vdash^{GEI} q^{GEI}$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\psi^{GEI} \in \Psi^{GEI}$ has a corresponding $\sigma(\psi^{GEI}) = \psi^{LCC} \in \Psi^{LCC}$ so we can say that II.a of property 2 holds.

II.b): We need to prove that,
$\gamma_1^{GEI} \vdash^{GEI} q^{GEI} => \gamma_1^{LCC} \vdash^{LCC} \sigma(q^{GEI})$
As, γ_k^{LCC} and γ_k^{GEI} are equivalent, and every $\sigma(\psi^{GEI}) = \psi^{LCC} \in \Psi^{LCC}$ has a corresponding $\psi^{GEI} \in \Psi^{GEI}$ so we can say that II.b of property 2 holds.

■

REFERENCES

[1] K. Bijon, R. Sandhu, and R. Krishnan. A group-centric model for collaboration with expedient insiders in multilevel systems. In *International Symposium on Security in Collaboration Technologies and Systems*, 2012.

[2] R. Sandhu. Lattice-based access control models. *Computer*, 26(11):9 –19, nov. 1993.

[3] M. V. Tripunitara and N. Li. Comparing the expressive power of access control models. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 62–71, New York, NY, USA, 2004. ACM.