Traffic Congestion Management System

Kotikalapudi Raghavendra Bruce M. McMillin
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO 65409-0350

Abstract

The project aims to provide a security model for a road traffic congestion management system. Its key purpose is to organize the flow of traffic so as to minimize congestion. This is done by balancing traffic load effectively (reactive approach) and by providing prior information to commuters such as the shortest time path from source to destination (preventive approach).

As congestion is also due to incidents, this system can work in tandem with incident detection and disaster recovery systems. The cost/benefit ratio of installing the sensor nodes in every road vehicle can be justified by using them for many other purposes such as collision avoidance, or for use in other cyber physical systems.

1. Cyber Physical Aspects

The CPS consists of a distributed network encompassing all the road vehicles by means of wireless sensor nodes. These nodes form a distributed network and transmit the relevant data to traffic control department (server). The physical aspect of the system entails the vehicle position, traffic scenario, weather conditions and any other relevant data that could be used by the CPS. Applications running on the server can analyze these data in real-time and provide interesting services to the users. One such service can be the fastest path calculation from source to destination, taking into account various factors such as traffic, distance, climatic conditions etc.

This data can also be used to regulate traffic signals so as to minimize congestion. A sudden spike in traffic can be regulated by distributing the traffic load among other free paths. Services running on the server can analyze this spike and compute a series of traffic signal changes. This data can be transmitted to the control circuitry (via an ad hoc network) for executing new series of actions.

The above scenario can essentially be modeled as a routing problem in a graph G(V, E), where V represents the destinations and E represents the possible paths amongst the vertices. With traffic represented as cost C for the edge E connecting vertices Vi and Vj, the regulation service aims distribute C among other edges such that the function representing average estimated time of all the vehicles to reach to their destinations is minimized. The minima of the function can be determined using gradient descent algorithms such as back propagation or other evolutionary algorithms.

This spike can also be attributed to the occurrence of an incident such as accidents or any other unexpected situations. These incidents can be detected and relayed to appropriate authorities for recovery.

2. Envisioned Security Concerns

This CPS faces the same basic fallacies as any other ad hoc wireless system.

a) **Lack of availability** is an obvious issue as the network is ad hoc. This can be a concern in an isolated area. This issue can be mitigated to some extent by using heuristic techniques to analyze the last known traffic data and by using GPS navigation system.

b) **Integrity** is certainly very important as manipulation of traffic data (man in middle) can have potentially dangerous implications. For instance, a terrorist group can use this information to target large number of people.

c) **Confidentiality** is important as users whereabouts shouldn't be leaked to third party and should be strictly used for analysis purposes only. Perhaps, the best way to achieve this is by avoiding the use of database. A neural network controller can be trained in real time to estimate future traffic conditions. As there is no persistent data in the system,

accessing and interpreting the data in real time remains as the only prime concern.

3. Survey of Models

Because security policy models have already been developed for computer security, we will review some of the most relevant modeling techniques. In the next section, we will apply these techniques to Traffic congestion management system.

3.1 HRU Model

The Harrison-Ruzzo-Ullman [3] access control matrix model describes the rights of subjects over objects, where each subject is also considered to be an object, by using a matrix. The rights that a particular subject has becomes a row vector in which the entry in a particular column reflects the subset of rights that the subject corresponding to the row has over the object corresponding to the column.

In the HRU model, the access control matrix describes the protection state and can change in discrete transitions which correspond to commands. A protection state is simply an instance of ACM representing the current state of the system. Primitive commands are create-subject, createobject, destroy-subject, destroy-object, and the operations on entries of the matrix corresponding to entering a right into or removing a right from the current subset occupying a position in the matrix.

The primary result regarding the HRU model is that the question of whether a particular state of a given system is safe for a given generic right is undecidable. However, it is known that for a given system is decidable if all the commands for transitions in the system are mono-operational.

3.2 Take Grant Protection Model

The take-grant model [3] is a special kind of HRU model which has only certain transitions enabled. It uses the two special rights of take and grant to indicate when a subject may obtain a right which another subject has over a particular object. It then becomes interesting to ask whether a subject can obtain a right without the cooperation of other subjects. This leads us to mention the concepts of canshare, can-steal, and conspiracy graphs.

Graphs are mentioned because the take-grant system is usually represented visually as a labeled directed graph with edges indicating when an object (the initial end of the edge) has a right (the label) over another object (the terminal end of the edge). A subject has the take right over an object, it can take (copy) without permission any right (labeled edge) which the object has. On the other hand, a subject may (voluntarily) choose to grant a right (which it has over an object) to any object over which it has a grant right.

It is considered less expressive as many extensions [5] to Take Grant model have been proposed. By restricting the types of transitions, take-grant model offers definitive answers to questions such as whether a right can be leaked or not. It also provides better visualization of protection states for a particular system.

3.3 Bell La-Padula Model

The Bell-LaPadula (BLP) is a state machine model used for enforcing access control in government and military applications [6]. It uses clearance levels to maintain confidentiality and access to classified information. Basically, it can be summarized by the phrase no read-up, no write-down, which means that subjects should not be allowed to read anything which is classified as more secret than their own classification, and they should also not be allowed to write to any object which is at a less secret classification level.

A lattice of subsets of categories may supplement the BLP model [3], and provide more expressiveness. When this feature is added, one uses the concept of domination of an objects classification by that of a subject to determine allowable read and write accesses. A classification is a pair consisting of the clearance level and the subset of categories to which it pertains to. One classification dominates another if the clearance level of the first is at least that of the second and if the second component (the subset of categories) of the first contains the second component of the second.

3.4 BIBA Model

The dual of the construction of the Bell-LaPadula is the basis for the Biba Information Integrity model [2]. Intuitively, information integrity is preserved when writing down from a higher level to a lower level of integrity or when reading from a higher level to a lower level. Writing to a higher level is not permitted, therefore, so a higher level subject must lower its integrity level to the low-water mark of the integrity levels of the sources from which it reads.

3.5 Chinese Wall Model

In business, a Chinese wall [4] or firewall is an information barrier implemented within a firm to separate and isolate persons who make investment decisions from persons who are privy to undisclosed material information which may influence those decisions. This is a way of avoiding conflict of interest problems. To achieve this, a policy should be enforced in such a way that SSC and * property is maintained.

SSC property says that if a user (subject) gains access to an object in a particular company dataset, then the user can only access objects within the same CD set. Also, the user cannot access objects within CD's that come under same COI classes. This is to say that when S first accesses O, a *chinese wall* is created between the CD containing O and the other CD's sharing the same COI class.* property prevents the indirect violation of this principle by saying that CD(O') = CD(O) for all unsanitized objects.

3.6 Non Interference, Non Inference and Non Deducibility

These are information flow models that describe the security of the system by analyzing the relation between high level and low level entities/events/action. All three models say that the system in insecure if low level events or actions are influenced by high level actions, but with varying degree of granularity.

A system is non interference [7] secure if the outputs (projection) of the system when a low level command is executed is the same as the outputs produced when low level commands are run after purging the high level commands from the command sequence. This is basically to say that that security is preserved whenever high level users are prevented from influencing the behavior of low level users.

A system is non inference secure if all the traces of the system are valid (consistent with the structure of the system) when high level events are removed the trace set. If this is not the case, then low level events can infer the occurrence of high level events, thereby compromising the security. A system is non deducible [7] if low level events cannot *uniquely* deduce the occurrence of high level events.

4 Specific Models and Rights Leakage

In this section, we will try to show the rights leakage for our infrastructure with respect to a specific model.

4.1 HRU Model

The access control matrix for the system under observation can be constructed as follows:

	AT	D	CS	TDS	TL	V	DU	TF
AT	O		A		R		R	R, W
D		O	A		R	O,C	R	R, W
CS			O	R,W		R	W	R
TDS			R,W	O	W			
TL					O			W

Table 1. ACM

Entities used in the ACM are as described below:

AT - Attacker.
D - Driver controlling the vehicle.
CS - The wireless sensor installed on the car. It is used to send/receive relevant data from the nearest traffic department server.
TDS - Represents the local traffic department's server. It provides services by analyzing data from car sensors.
V - The road vehicle.
DU - The display unit installed within the car. It can be used by the driver to find the shortest time path from source to destination.
TL - Represents the local traffic signals at intersections. It also acts as a communication device by receiving control strategy from traffic department server
TF - Represents the traffic flow or the traffic density for the region under consideration.

Rights are defined as follows:

O = Own
C = Control (Issue control signals)
R = Read
W = Write
A = Accessible (Within physical reach)

4.1.1 Rights Leakage

Consider the following command set of commands

Command C1
If *A* in A[AT, CS]
    delete all rights in A[CS, TDS];

This means that if the attacker has access to the car sensor of the driver, he/she could dispose it (thus, creating an effect of deletion of all rights). This is an attack on availability. Here, only the deletion between TDS and CS are shown as CS is rendered useless without *R, W* access to TDS. Rights deletion to remaining entities are excluded for the sake of simplicity.

Command C2
If *R* in A[AT, DU] and *R* in A[AT, TL] and *W* in A[AT, TF]
    Insert *R* in A[AT, V];

This command indicates that if the attacker can read the display unit for using the the shortest path service and observe the change in the traffic lights at various intersections, then the attacker can predict the target vehicle location with a certain probability. This probability can further be in-

creased by manipulating the traffic control elements such as the traffic flow. This is a threat to confidentiality.

Moreover, by manipulating the traffic flow, the attacker is influencing the behavior of TDS, indirectly affecting the shortest path service and traffic light control modules. Hence, this is also an attack on data integrity. It can also be viewed as a from of usurpation.

4.2 Take Grant Model

This section tries to express the right leakages as a sequence of graph transitions. As the basic TG model is insufficient to express the rights leakage for the infrastructure under consideration, we propose some extensions to the basic TG model.

4.2.1 TG Extensions

We define a defacto delete rule as follows:

Definition 1. *Let A be a subject and B, C be subjects or objects. If A owns B, then A can order B to relinquish its rights over other entities provided B is obedient towards A.*

Ex - Let A be an owner, B be a dog and C be the dog house. A *owns* B and B *sleeps* in C. As an owner, A can command the dog B to relinquish its *sleep* right over C. We however assume the dog to be faithful to its owner.

Theorem 4.1. *Let G represent a TG graph. Let G' $\subseteq$ G consist of subjects or objects X, Z and Y_i $\forall 1 \leq i \leq n$. X has p_i rights to Y_i and Y_i has q_i rights to Z as shown in figure 1*

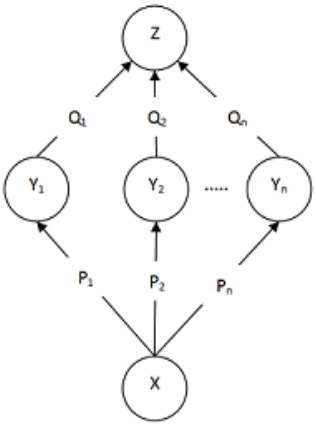

Figure 1. Graph G'

If $\exists$ *command C of form:*
If($\sum S_i$ *in A[X, Y_i])*
do some activity;

then, the transition G $\rightarrow$ *G" caused by command C, can be expressed in TG model as shown in figure 2 with the right propagating across the meta state* ($\sum Y_i$)*, where, P =* $P_1 \cap P_2 \cap ... P_n$*, Q =* $Q_1 \cap Q_2 \cap ... Q_n$*,* $S_i \subseteq P \; \forall \; 1 \leq i \leq n$ *and S =* $S_1 \cup S_2 \cup ... S_n$

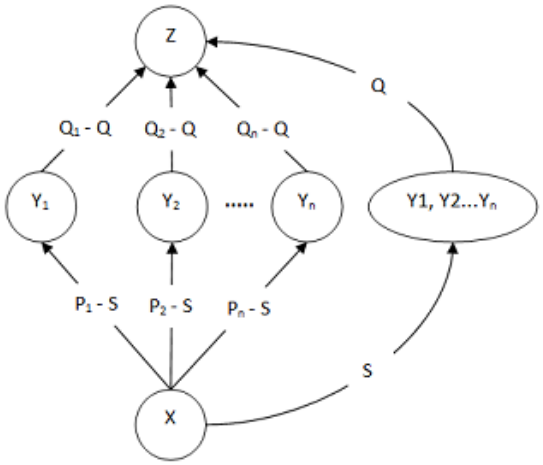

Figure 2. Graph G'

4.2.2 Leakage sequences

The take grant model for the system under consideration can be shown as follows:

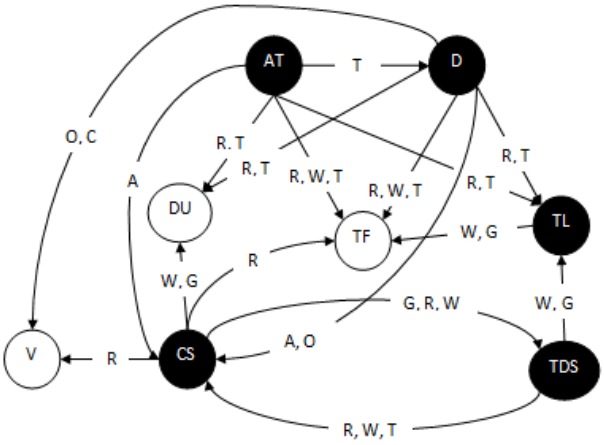

Figure 3. Take Grant Model

Command C1

The threat to availability by this command can be represented as follows:

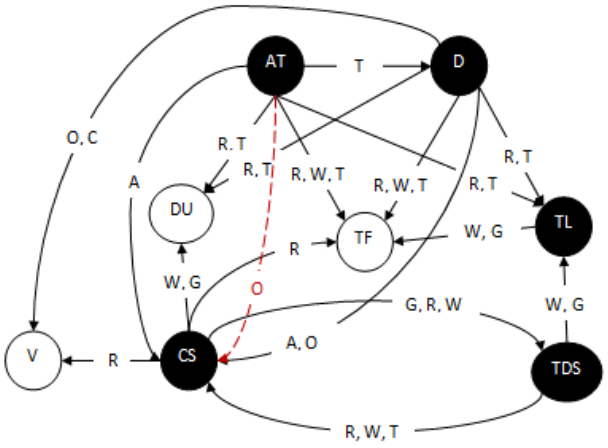

Figure 4. Command C1, Sequence 1

AT takes *own* from D to CS.

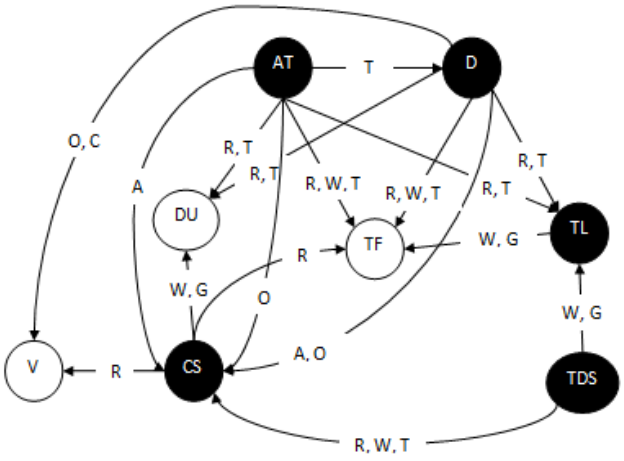

Figure 5. Command C1, Sequence 2

Since, AT now *owns* CS, using defacto rule (1), AT *deletes* all the rights that CS has over TDS.

Command C2

The rights leakage expressed by this command can be represented as follows:

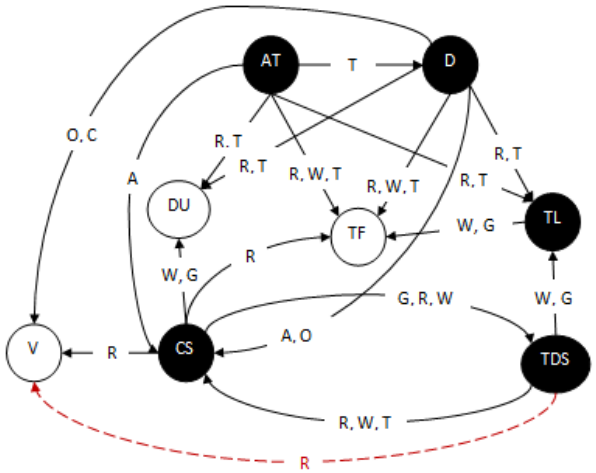

Figure 6. Command C2, Sequence 1

TDS takes (R to V) from CS.

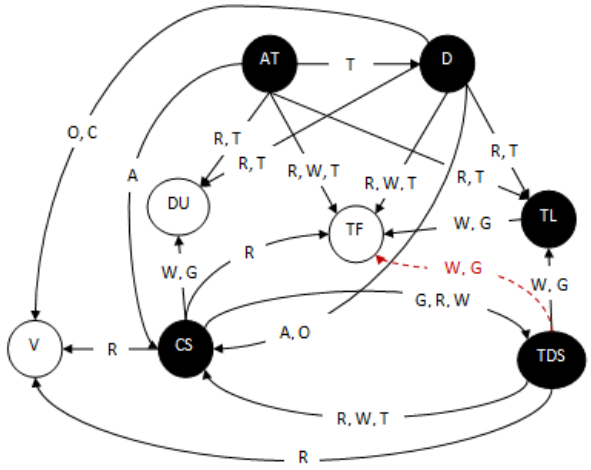

Figure 7. Command C2, Sequence 2

TG gets *W, G* to TF using defacto rules.

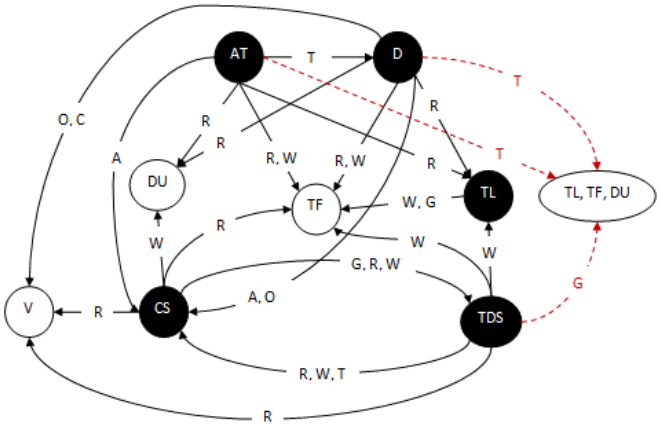

Figure 8. Command C2, Sequence 3

The graph from figure 7 can be represented as shown above by using theorem 4.1.

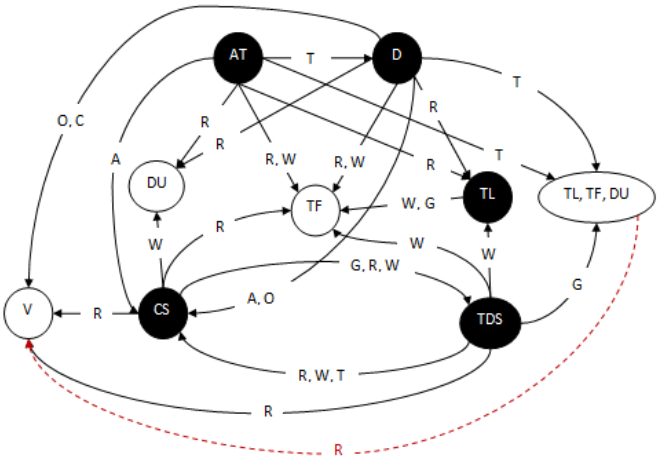

Figure 9. Command C2, Sequence 4

Meta state TL, TF, DU takes *R* from TDS to V.

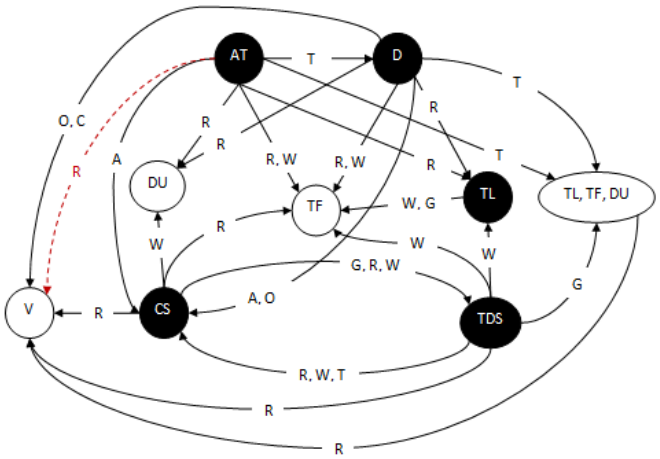

Figure 10. Command C2, Sequence 5

AT takes *R* from (TL, TF, DU) to V.

4.2.3 Conspiracy Graph

The Take grant protection graph for the system is designed as:

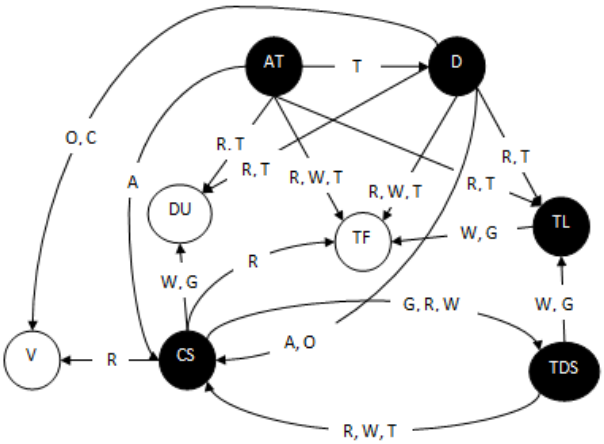

Figure 11. Take Grant Model

Access sets for the subjects are:

A(CS) = {CS, DU, TDS}
A(TDS) = {TDS, CS, DU, TL}
A(AT) = {AT, D, DU, TF, TL}
A(D) = {D, DU, TF, TL}
A(TL) = {TF}

Non empty deletion sets are:

δ(CS, TDS) = {CS, TDS}
δ(CS, AT) = {DU}
δ(CS, D) = {DU}
δ(TDS, AT) = {TL, DU}
δ(TDS, D) = {DU, TL}
δ(AT, D) = {AT, D}
δ(AT, TL) = {TF}
δ(D, TL) = {TF}

Hence, the conspiracy graph can be constructed as follows:

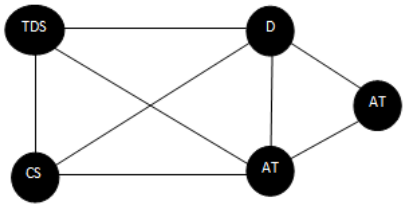

Figure 12. Conspiracy graph

4.3 Bell La-Padula and Biba Model

In Bell LaPadula Model, we are to assign subjects and objects in various security levels such that *read up* and *write down* are not allowed in accordance to SSC and * properties.

TDS and CS can *R, W* to each other, so they must be in the same security level. Also, (CS, TDS) can *W* to (DU, TL) and (AT, D) can *R* from (DU, TL, TF). This means that (DU, TL, TF) must either have same or higher security clearance than (TDS, CS) pair and (AT, D) must be higher than (DU, TL, TF). In a nutshell, F_s(TDS) = F_s(CS) $\leq$ F_o(DU), F_o(TL), F_o(TF) $\leq$ F_s(AT), F_s(D), where F_s and F_o denote subject security level and object clearance level respectively.

As BIBA model is a dual of Bell La-Padula model, we have I(TDS) = I(CS) $\geq$ I(DU) = I(TL) = I(TF) $\geq$ I(AT) = I(D), with I(V) being the highest, as it is the source of data. I(X) represents the integrity level of the entity X. Also integrity level of AT, D is the lowest, which makes perfect sense as they form the sink or destination of the source data.

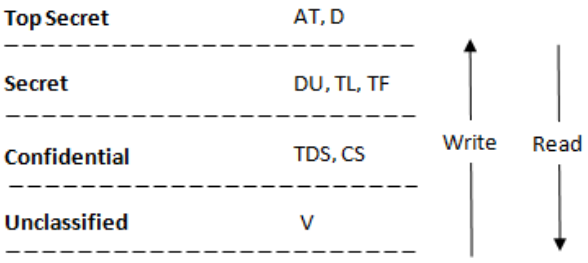

Figure 13. Bell LaPadula Model

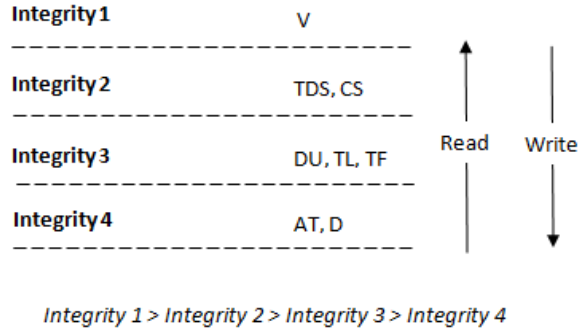

Figure 14. BIBA Model

4.4 Chinese Wall

As indicated in the HRU model, a potential right leakage i.e., right (D $\rightarrow$ V) leaks to (AT $\rightarrow$ V). Hence, we will attempt to patch this flaw using chinese wall model by considering the subjects AT and D. The objects in consideration are V, DU and TF. D can access all three objects. To achieve this there are two possibilities.

1) V, DU, TF should be in the same company dataset (CD) within a conflict of interest class (COI). This idea is illustrated as shown below. In this figure, outer shell represents the COI class and inner shell(s) represent CD sets.

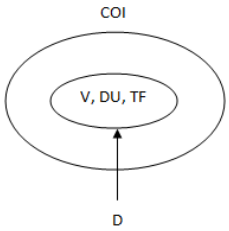

Figure 15. Possibility 1

2) V, DU, TF contribute to CD's within different COI classes. D can still access all the objects as permitted by SSC property. This idea is illustrated as shown below.

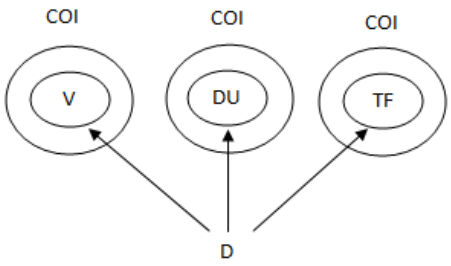

Figure 16. Possibility 2

In either of the two possibilities, it is not possible to accommodate AT such that it has access to DU, TF and not V. Hence, chinese wall model fails to make this infrastructure secure.

4.5 Non Interference

In this section, we'll observe the information flows for the given infrastructure and determine the security of the system by determining the dependencies between high level and low level entities. We consider TDS and CS to be high level entities as the information flow among these subjects are confidential. Let AT be a low level entity, as AT can observe low level manifestations of high level outputs.

High Level = {TDS, CS}
Low Level = {AT}

Now, we identify the commands or operations associated with these entities.

C_s{TDS} = {Control traffic lights, compute shortest time path and host it as a service}
C_s{CS} = {monitor environmental data, send/receive data from TDS}

Proj(AT, C_s, σ_o) = Outputs observable to AT = {change in traffic lights (TL), shortest path service on DU, etc..}
Proj(AT, $\pi_{TDS}\{C_s\}$, σ_o) = Outputs observable to AT when TDS (high level entity) is purged from C_s = ϕ

As Proj(AT, C_s, σ_o) $\neq$ Proj(AT, $\pi_{TDS}\{C_s\}$, σ_o), this infrastructure in not non interference secure.

4.6 Non Inference

Consider the command show shortest time path by TDS. The following events occur {TDS computes shortest path, CS receives data from TDS, CS routes this info to DU and driver sees information displayed on DU}. This information flow can be summarized as shown below:

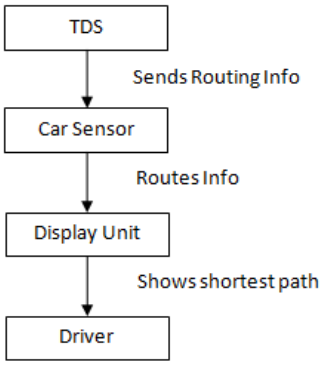

Figure 17. Information flow

Since TDS, CS are high level entities, the events 'TDS computes shortest path' and 'CS receives data from TDS' are high level events. The event 'view shortest time path' can be accessed by both driver(D) and attacker (AT). Since, AT is a low level entity, this event can be classified as a low level event.

If the event 'TDS computes shortest time path is purged', i.e., $\pi_{TDS,computeshortesttimepath}(C_s)$ then, the event 'DU displays shortest path' is no longer a valid trace in the system as it never receives this information from TDS via CD. Hence, this infrastructure in not non inference secure.

4.7 Non Deducibility

The system is ND secure if low level events cannot be *uniquely* traced back to the occurrence of high level events.

Consider the following low level event traces.
Low Level Traces = {Change of traffic lights, change of traffic flow, display of shortest time path service on DU}

The events 'change of traffic lights' and 'change of traffic flow' cannot be uniquely traced back to a high level event as these events occur even if high level events are purged. The event 'display shortest time path in DU', however, occurs if and only if the event 'TDS computes shortest time path and hosts the service' has occurred. Therefore, this infrastructure is not ND secure as a low level event can *deduce* or *trace* it back to the occurrence of a *specific* high level event.

5 Conclusion

HRU model is expressible as it is easy to model the infrastructure into an access control matrix. TG model on the other hand is restrictive. In particular, the notion of rights leakage via multiple objects was hard to express. The only way to do this was to combine those objects into a meta state and show the leakage via that object. Also, the notion of deleting a right (availability concern) is hard to capture with the standard TG model.

Bell-LaPadula and BIBA models are more suitable for military style processes that follow a chain of command. Although they applied for this infrastructure, they are too restrictive and are hard to enforce in practice. Chinese wall model provides a good balance between confidentiality and integrity from a business perspective but is not always feasible. In this infrastructure, Chinese wall policy was too restrictive to allow for the system to function in a secure manner.

Information flow models applied naturally to this infrastructure. This system is not non interference secure. This result was expected as it is typical for low level events to be influenced by high level actions. Also, it is not non inference secure as information flow gets disrupted when high level events are purged. The occurrence of a low level event (display of shortest path) can be traced back to the occurrence of a specific high level event (TDS computing shortest path). Hence, the system is insecure with respect to non deducibility. This fact is justified by the rights leakage expressed in TG and HRU.

In conclusion, large CPS like traffic congestion management system will always have some sort of rights leakage, even if data channels are perfectly encrypted.

References

[1] D. Bell. Looking back at the bell-lapadula model. *Computer Security Applications Conference*, 98(E2):15 pp. – 351, 2005.

[2] K. J. Biba. Integrity considerations for secure computer systems. 1977.

[3] M. Bishop. *Computer Security: Art and Science*. Addison-Wesley, 2004.

[4] M. N. D. Brewer. The chinese wall security policy. page 206214, 1989.

[5] J. Frank and M. Bishop. Extending the take-grant protection system. 1996.

[6] C. H. Hansche, Susan; John Berti. *Official (ISC)2 Guide to the CISSP Exam*. CRC Press, 2003.

[7] J. M. J. A. Gougen. Security policies and security models. pages 11–20, 1982.