Proceedings of the International Conference on Research Trends in Science and Technology - RTST 2002. March 4-6, 2002. Beirut, Lebanon.

MLS/ERD: A Tool for Building Multilevel Secure Entity Relationship Diagrams

Ramzi A. Haraty, Omar Ariss, and Peter Attallah
Lebanese American University
P.O. Box 13-5053 Chouran
Beirut, Lebanon 1102 2801
Email: rharaty@lau.edu.lb

Abstract

Entity Relationship models, which are popular high-level conceptual data models, are frequently used for the conceptual design of database applications. Unfortunately, there are no specific tools for drawing such models, especially when designing multilevel databases. In this work, we build a tool – MLS/ERD – that will allow users to easily and clearly design multilevel databases.

Keywords: Entity Relationship Model and Multilevel Security.

1. Introduction

In this paper we will present a software tool for designing the ER Diagram of a database depending on a multilevel secure system. This security system will be applied to entities, relationships, and attributes and will restrict every subject and object of the ERD to abide by the principals it uses.

As an overview, this security tool is used at the initial levels of building a database enterprise. In other words if you want to construct a multilevel secure database, the security tool will be the first step to start with after configuring entities and attributes.

2. Multilevel Security

All over the world many types of security policies or control mechanisms have been integrated into database systems to provide data security [2] [3]. In our security tool we applied the Mandatory Access Control or MAC mechanism, developed by Bell and LaPadula [1]. MAC mechanism does not leave protection decisions of objects to the discretion of the owners. The system enforces the protection decisions. This mechanism defines a database by a set of subjects and objects. In the MLS/ERD tool every object has a sensitivity level decided by the designer. These levels can be:

1. **Top Secret**
2. **Secret**
3. **Confidential**
4. **Unclassified**

These security levels are applied to the objects used in the MLS/ERD. These security levels are programmed so that no illegal relations or decisions occur. So every level is restricted to certain rules that are configured in the security tool.

2.1 Security Levels

a. ***Top Secret:*** the objects that are related to this level are highly confidential and no other low objects could have access to the entities of this security level. But on the other hand this level is the highest security level and could have access to any other level in the hierarchy. The designer of the database should be very careful when designing the database and while determining the top-secret objects, because in the future these objects could not be accessed except by eligible people.

b. ***Secret:*** this level is lower one level than the top-secret level. The objects related to this level are of great importance and should not be revealed except to secret people. For example, if we are designing a military database we could classify military bases as secret entities. This level has access to the confidential and unclassified levels but not to the top secret.

c. ***Confidential:*** as the name of this level appears, objects that are confidential but still could be accessed by a great number of users related to this level. This level usually contains the largest number of objects, because most of the entities can be considered confidential. And as other levels this level has access to lower levels only.

d. ***Unclassified:*** the objects of this level are of no importance in the hierarchy of security levels. Any other objects of any other levels could reach these objects. So their entities could have any kind of relations with other levels. Fortunately, it is the lowest or weakest level in the security system.

3. Security Restrictions

Multilevel secure domains can be defined as follows. There is an object O_i of a certain classification, this object dominates another object O_j iff $O_i > O_j$. In other words, object O_i could have access to object O_j if its classification level is greater than the other object's classification level. Its security classification level determines the classification levels an object can access.

- High objects may have access to low objects and low resources.
- High relations may have access to low data or low entities.
- High data must not leak to low systems or objects.

It is important to note that the development of high-assurance software necessary to provide separation between the lowest security level (unclassified) and the highest security level (top secret) information has proven to be both technically challenging and very expensive through twenty years of computer security history [3]. So we can say that our security tool provides a good designing tool that makes use of this complicated security system.

In figure 3.1 you can see a general security hierarchy between the different levels. This figure describes the relation between security levels. The flow of access restrictions is downwards and could never be upwards. High levels have access to low levels and low resources. Also objects of the same level could have access between each other. This is the general rule for the security tool.

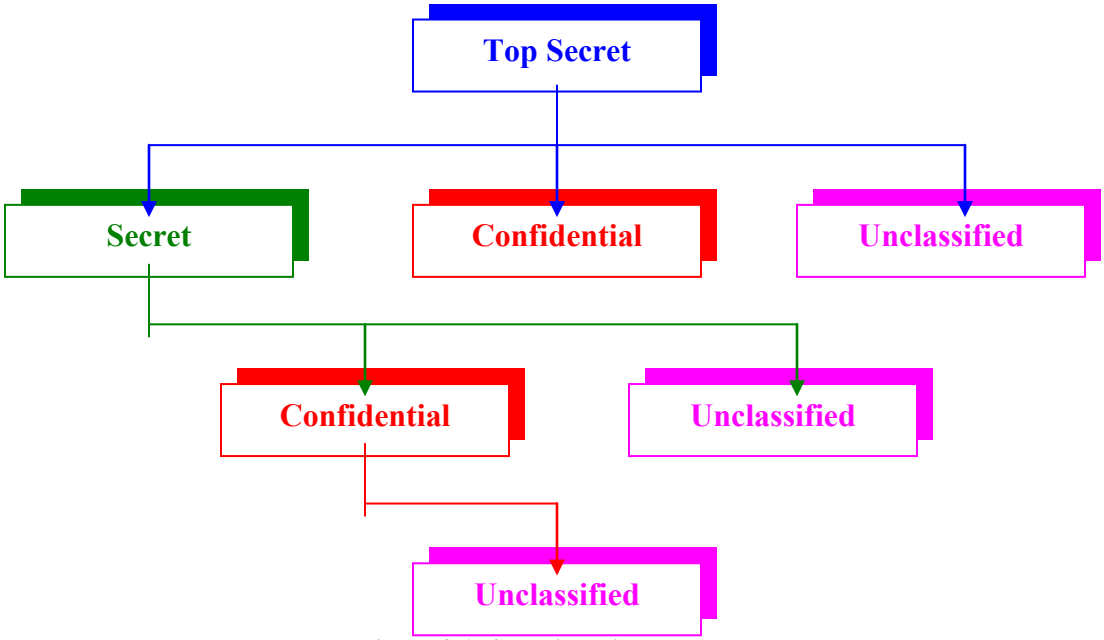

Figure 3.1: Security Hierarchy

Entities, attributes and relationships interact according to the following:

1. Attributes: let us consider a set of attributes A1, A2 … Ai, belonging to a certain entity E.

a. ***First Restriction:*** every attribute Ai should be of a lower or of the same security level as the level of the entity E.
b. ***Second Restriction:*** the primary key Pk should have the lowest security level among the attributes A1, A2 … An. In other words any attribute Ai could have a higher or same security level as the primary key, but not a level lower than it.

So for example, an unclassified entity should have all of its attributes unclassified because of the security restrictions. Table 3.1 shows how these security restrictions work.

Entity	Primary Key	
TS	TS – S – C – U	**TS: Top Secret** **S: Secret** **C: Classified** **U: Unclassified**
S	S – C – U	
C	C – U	
U	U	

Primary Key	Attribute
TS	TS
S	TS – S
C	TS – S – C
U	TS – S – C – U

Table 3.1: Attributes' Security Restrictions

2. Entities: entities as attributes could have any kind of security levels, but the relationships that the entities share with each other are restricted to certain rules:

a. ***First Restriction:*** consider two entities E1 and E2. E1 and E2 could share a certain relationship if the security level of E1 is the same security level of E2.
b. ***Second Restriction:*** if E1 and E2 are of different security levels, a relationship could be established between them if it is launched from the entity that has a dominant security level. For example, if E1 dominates E2, a relation could be established if it was launched from E1. In other words, an entity of a certain level could have relations with entities that are of lower or the same security level as this entity.

In table 3.2 we summarize the security restrictions.

Entity One	Relation	Entity Two	
TS	***R***	TS – S – C – U	**TS: Top Secret** **S: Secret** **C: Classified** **U: Unclassified**
S	***R***	S – C – U	
C	***R***	C – U	
U	***R***	U	

Table 3.2: Entities' Security Restrictions

3. Relationships: relationships could be of any type of security levels, but its level is determined according to the dominating entity:

a. ***First Restriction:*** if the two entities that share a relationship R are of the same security level, R will be of the same level as these two entities if all the attributes are

of the same security level as the attributes. Else, the security level of R will depend upon the security level of the attributes concerned.

b. ***Second Restriction:*** if the two entities that share a certain relationship R have different security levels, R will depend on the security level of the attributes concerned in this relation.

c. ***Third Restriction:*** a relationship R could not be established from a lower security level entity to a higher one. Only higher entities could share relationships with other lower level entities.

4. The MLS/ERD Tool

4.1 Environment

The environment of the security tool is similar to any windows application. It has a menu bar, status bar, a tool bar and a drawing area that could be split if drawing a large-scale diagram. Figure 4.1 describes the whole environment and takes a deeper look into the objects of this tool.

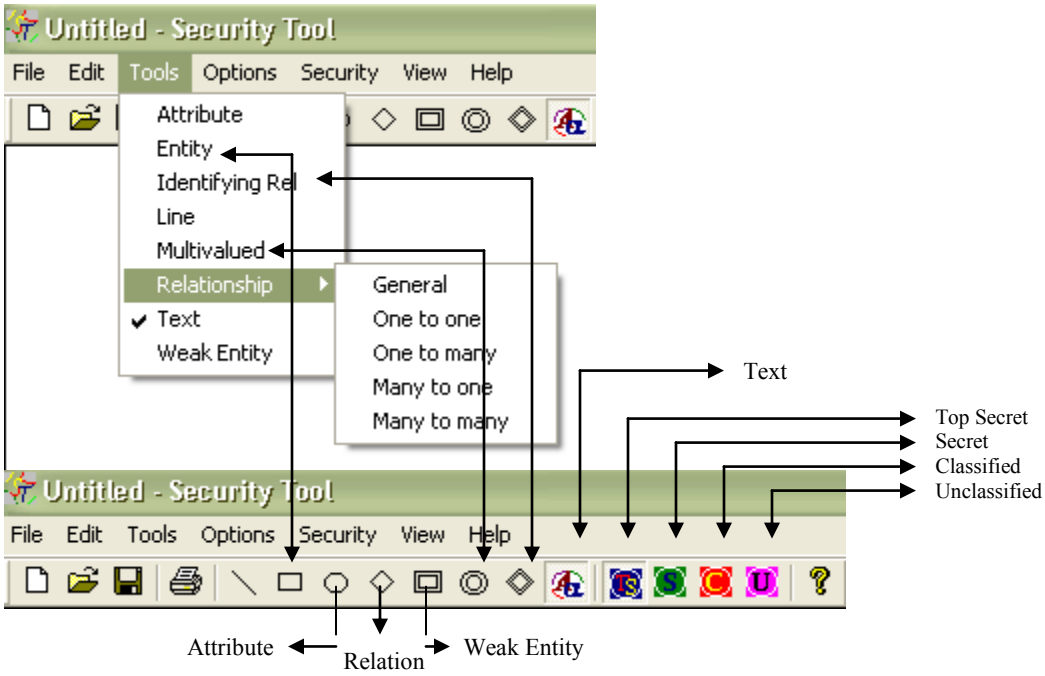

Figure 4.1: The Environment

The environment consists of the following menu interfaces:

a. ***File menu:*** it is the regular file menu that contains the new, open, save, save as, print, print preview and page setup.

b. ***Edit menu:*** also it is the regular edit menus that contains the undo and delete all commands.

c. ***Tools menu:*** it contains the database objects.

d. ***Options menu:*** it contains options for the line style and line thickness.

e. ***Security menu:*** it contains the security levels.

f. ***View menu:*** it is the normal view menu.

g. ***Help menu:*** it contains information about the tool.

The file types that are saved are of ***.sct*** extension.

4.2 Security System

Now we will explain how the security restrictions are applied while using or drawing a diagram using this software.

First the user should choose the security level of the object he/she wishes to draw. Second the user chooses the database object (Entity, Relation, and Attribute) and then draws the object as shown in figure 4.2.

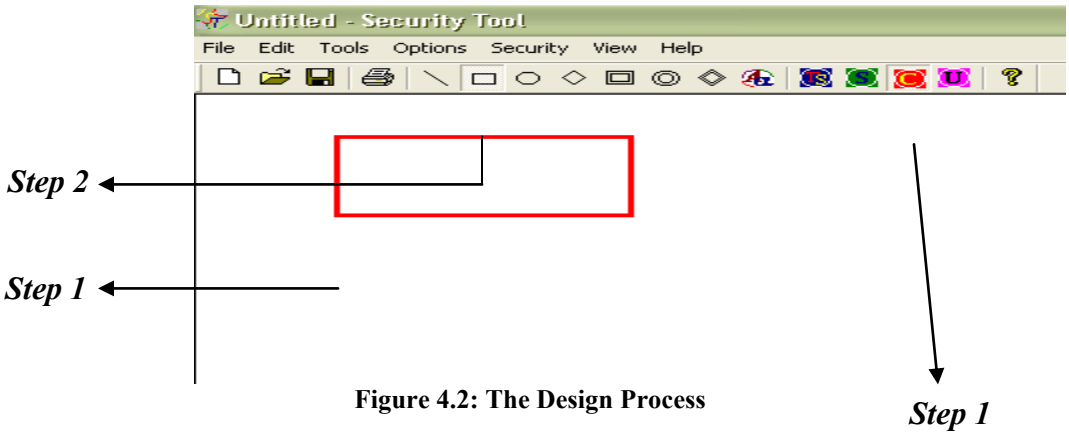

Figure 4.2: The Design Process

By this way start drawing your diagram and below we will explain how the security system works.

Now let us say that you drew an entity and you want to draw an attribute for it. First, the attribute should be of a lower or of the same security level as its entity and second, you must be sure that the line is launched from the higher-level object. This is explained in figure 4.3. In this figure an entity of a confidential level is drawn, also an attribute of an unclassified level is drawn. Now we want to link these objects by a line. When choosing the level of the line you must be sure that the security level of it is lower than or of the same security level as the highest object between the two objects you want to link. If this restriction is violated an error message box will appear specifying that you can not link a line of a security level lower than the object being linked. Also you must always be sure to launch the line from the object that has the highest security level. In return to the figure we see that the line is drawn without any error messages, that's because the line is launched from the higher-level object to the lower-level one, also the line has the same security level as the highest-level object. In figure 4.4 an error message is generated once a security violation occurs.

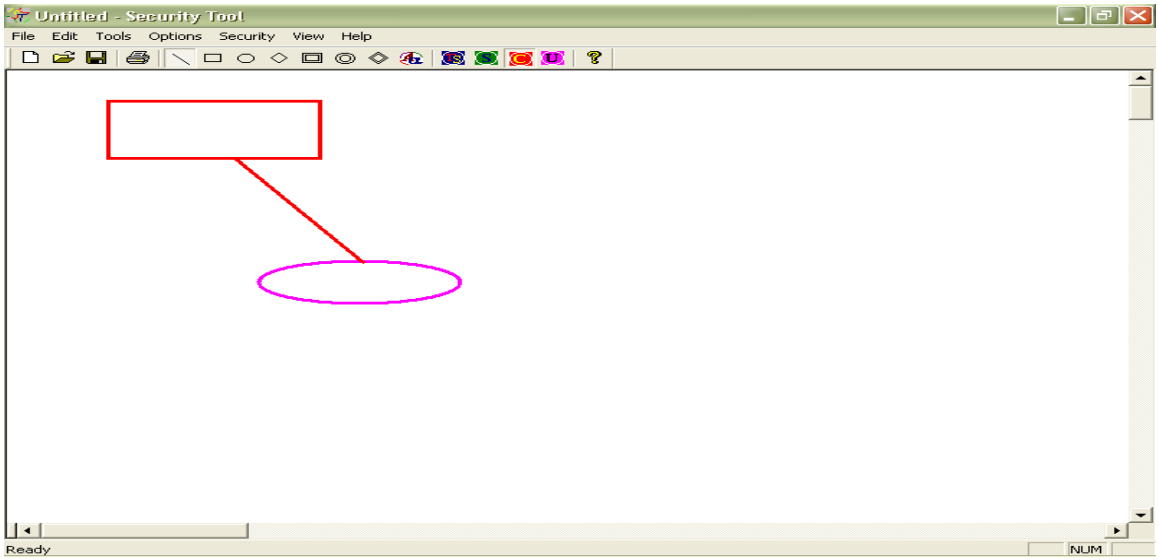

Figure 4.3 Legal Design

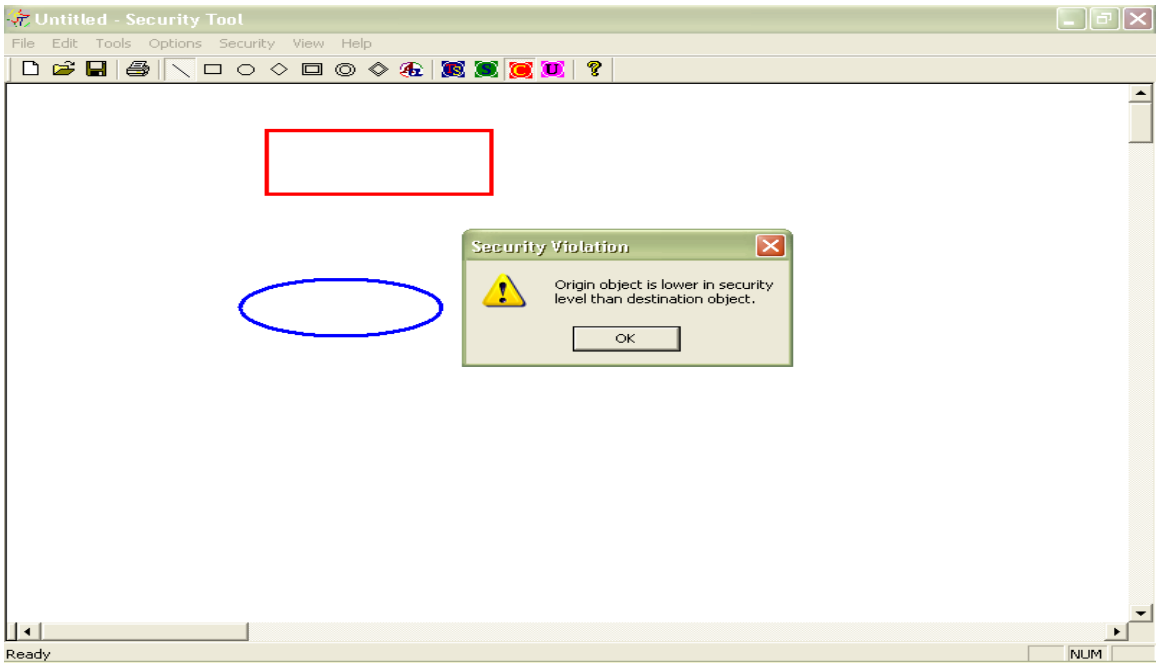

Figure 4.4 Security Violation

With respect to relationships, their security levels should be of the same as that of the highest entity (if the two entities are of the same level, the relationship will be of the same level too). Also the lines should belong to the highest security level of the database objects being linked.

5. Conclusion

In this paper we explained multilevel security and implemented a tool to design multilevel secure databases. The tool was coded using MFC (Microsoft Foundation Classes) and Visual C++. Further work includes refining the tool further and adding more options to it.

References

[1] Bell, D. and LaPadula, L. *Secure Computer Systems: Unified Exposition and Multics Interpretation*. Technical Report. The Mitre Corporation. 1976.

[2] Elmasri, R. and Navathe, S. "*Fundamentals of Database Systems*", Addison Wesley, Massachusetts. 2000.

[3] Jackson, J. "*MAC & DAC Brief*", www.garrison.com/html/docmacdac.html. 2002.

[4] Kang, Myong H. "*A Multilevel Secure Workflow Management System*", Information Technology Division, Naval Research Laboratory, LSDIS Lab, Department of Computer Science. 2000.