Chapter Five

Database Security and Authorization

Outline

- ✓ Database Security and Authorization
- ✓ Discretionary Access Control
- ✓ Mandatory Access Control and Role-Based Access Control
- ✓ Introduction to Flow Control
- ✓ Encryption and Public Key Infrastructures

Database Security

- **Database security** is the mechanisms that protect the database against intentional or accidental threats.
- Database security is concerned with avoiding the following situations:
  - theft and fraud, loss of confidentiality (secrecy), loss of privacy, loss of integrity, and loss of availability.

Security Goals

- **Data Confidentiality**
  - It is concerned with having secret data remain secret
- **Data Integrity**
  - Unauthorized users should not be able to modify any data without the owners permission
  - Includes removing data and adding false data
- **System Availability**
  - Means nobody can disturb the system to make it unusable

Cont'd

- A threat is any situation or event , intentional or accidental, that will adversely affect a system and consequently an organization.
- It may results in degradation of some/all security goals like;
  - ✓ **Loss of integrity**
  - ✓ **Loss of availability-**
  - ✓ **Loss of confidentiality**
- To protect databases against these types of threats four kinds of countermeasures can be implemented

Control Measures

- Four main control measures are used to provide security of data in databases:
  - Access control
  - Inference control
  - Flow control
  - Data encryption

Cont'd

Access control

- The security mechanism of a DBMS must include provisions for restricting access to the database as a whole; this function is called **access control** and is handled by creating user accounts and passwords to control login process by the DBMS.

Cont’d

Inference control (for statistical DBs)

- The security problem associated with databases is that of controlling the access to a **statistical database**, so that information about the individuals cannot be accessed.
- The countermeasures to **statistical database security** problem is called **inference control** measures.

Cont'd

Flow Control

- Another security is that of **flow control**, which prevents information from flowing in such a way that it reaches unauthorized users.
- Channels that are pathways for information to flow implicitly in ways that violate the security policy of an organization are called **covert channels**.

Cont'd

Data encryption

- A final security issue is **data encryption**, which is used to protect sensitive data that is being transmitted via some type communication network.
- The data is **encoded** using some coding algorithm.
- An unauthorized user who access encoded data will have difficulty deciphering it, but authorized users are given decoding (or keys) to decipher data.

The Database administrator (DBA)

- The database administrator (**DBA**) is the central authority for managing a database system.
  - The DBA's responsibilities include
    - granting privileges to users who need to use the system
    - classifying users and data in accordance with the policy of the organization
- The DBA is responsible for the overall security of the database system.

Cont’d

- The DBA has a DBA account in the DBMS
  - Sometimes these are called a system or superuser account
  - These accounts provide powerful capabilities such as:
    - ✓ Account creation
    - ✓ Privilege granting
    - ✓ Privilege revocation
    - ✓ Security level assignment

Cont’d

Three types of database security mechanisms:

- ✓ Discretionary security mechanisms
- ✓ Mandatory security mechanisms
- ✓ Role based security mechanisms

Discretionary Access Control

- The typical method of enforcing **discretionary access control** in a database system is based on the **granting** and **revoking privileges**.
- This is done by including statements in query language that allow DBA to grant and revoke privileges.

Types of Discretionary Privileges

- **The account level**: the DBA specifies the particular privileges that each account holds independently of the relations in the database.
- **The relation (or table level):** the DBA can control the privilege to access each individual relation or view in the database.
  - commands provide privileges at the relation and attribute level only

Cont’d

To control the granting and revoking of relation privileges,

- each relation R in a database is assigned an **owner account**, which is typically the account that was used when the relation was created in the first place.
- The owner of a relation is given *all* privileges on that relation. In SQL2, the DBA can assign an owner to a whole schema by creating the schema and associating the appropriate authorization identifier with that schema, using the CREATE SCHEMA command.
- The owner account holder can pass privileges on any of the owned relation to other users by **granting** privileges to their accounts.

Cont’d

- The privileges at the **account level** apply to the capabilities provided to the account itself and can include
  - the CREATE SCHEMA or CREATE TABLE privilege, to create a schema or base relation;
  - the CREATE VIEW privilege;
  - the ALTER privilege, to apply schema changes such adding or removing attributes from relations;
  - the DROP privilege, to delete relations or views;
  - the MODIFY privilege, to insert, delete, or update tuples; and
  - the SELECT privilege, to retrieve information from the database by using a SELECT query.

Cont’d

- The privileges at the **relation R level control to** access each individual relation can include
- **SELECT** (retrieval or read) privilege on R:
  - Gives the account retrieval privilege.
  - In SQL this gives the account the privilege to use the **SELECT** statement to retrieve tuples from R.

Cont’d

- **MODIFY** privileges on R:
  - This gives the account the capability to modify tuples of R.
  - In SQL this privilege is further divided into **UPDATE**, **DELETE**, and **INSERT** privileges to apply the corresponding SQL command to R.
  - In addition, both the **INSERT** and **UPDATE** privileges can specify that only certain attributes can be updated by the account.
- Notice that to create a **view**, the account must have **SELECT** privilege on all relations involved in the view definition.

Cont'd

Specifying Privileges Using Views

- The mechanism of **views** is an important discretionary authorization mechanism in its own right.
- For example,
  - If the owner A of a relation R wants another account B to be able to retrieve only some fields of R, then A can create a view V of R that includes only those attributes and then grant SELECT on V to B.
  - The same applies to limiting B to retrieving only certain tuples of R; a view V' can be created by defining the view by means of a query that selects only those tuples from R that A wants to allow B to access.

Propagation of Privileges using the GRANT OPTION

- Whenever the owner A of a relation R grants a privilege on R to another account B, privilege can be given to B with or without the **GRANT OPTION**.
- If the **GRANT OPTION** is given, this means that B can also grant that privilege on R to other accounts.
  - Suppose that B is given the **GRANT OPTION** by A and that B then grants the privilege on R to a third account C, also with **GRANT OPTION**. In this way, privileges on R can **propagate** to other accounts without the knowledge of the owner of R.
  - If the owner account A now revokes the privilege granted to B, all the privileges that B propagated based on that privilege should automatically be revoked by the system.

Cont'd

Example 1

- Suppose that the DBA creates four accounts
  - A1, A2, A3, A4 and wants only A1 to be able to create relations. Then the DBA must issue the following GRANT command in SQL

  GRANT CREATETAB TO A1;

Example 2

- User account A1 can create tables under the schema called **EXAMPLE**.
- Suppose that A1 **creates** the two base relations **EMPLOYEE** and **DEPARTMENT**
  - A1 is then **owner** of these two relations and hence all the relation privileges on each of them.
- Suppose that A1 wants to grant A2 the privilege to insert and delete tuples in both of these relations, but A1 does not want A2 to be able to propagate these privileges to additional accounts:

  GRANT INSERT, DELETE ON
  EMPLOYEE, DEPARTMENT **TO A2;**

Cont'd

Example 3

- Suppose that A1 wants to allow A3 to retrieve information from either of the table (Department or Employee) and also to be able to propagate the SELECT privilege to other accounts.
- A1 can issue the command:

  GRANT SELECT ON EMPLOYEE, DEPARTMENT
  TO A3 **WITH GRANT OPTION;**
- A3 can grant the **SELECT** privilege on the **EMPLOYEE** relation to A4 by issuing:

  GRANT SELECT ON EMPLOYEE **TO** A4;
  - Notice that A4 can't propagate the SELECT privilege because GRANT OPTION was not given to A4

Cont'd

Example 4

- Suppose that A1 decides to revoke the SELECT privilege on the EMPLOYEE relation from A3; A1 can issue:

  REVOKE SELECT ON EMPLOYEE **FROM** A3**;**

- The DBMS must now automatically revoke the SELECT privilege on EMPLOYEE from A4, too, because A3 granted that privilege to A4 and A3 does not have the privilege any more.

Cont'd

- Example 5
  - Finally, suppose that A1 wants to allow A4 to update only the SALARY attribute of EMPLOYEE;
  - A1 can issue:

    GRANT UPDATE ON EMPLOYEE **(**SALARY**) TO** A4**;**

    - The **UPDATE** or **INSERT** privilege can specify particular attributes that may be updated or inserted in a relation.
    - Other privileges (**SELECT**, **DELETE**) are not attribute specific.

Mandatory Access Control

- The discretionary access control techniques of granting and revoking privileges on relations has traditionally been the main security mechanism for relational database systems.
- This is an all-or-nothing method: A user either has or does not have a certain privilege.
- In many applications, and *additional security policy* is needed that classifies data and users based on security classes.
- This approach as **mandatory access control**, would typically be *combined* with the discretionary access control mechanisms.

Cont'd

- Typical **security classes** are
  - top secret (TS),
  - secret (S),
  - confidential (C),
  - and unclassified (U),
  - where TS is the highest level and U the lowest:

  $TS \geq S \geq C \geq U$
- The commonly used model for multilevel security, known as the Bell-LaPadula model, classifies each **subject** (user, account, program) and
- **object** (relation, tuple, column, view, operation) into one of the security classifications, T, S, C, or U:
- **clearance** (classification) of a subject S as **class(S)** and to the **classification** of an object O as **class(O)**.

Consider query SELECT * FROM employee

(a) EMPLOYEE

Name		Salary		JobPerformance		TC
Smith	U	40000	C	Fair	S	S
Brown	C	80000	S	Good	C	S

(b) EMPLOYEE

Name		Salary		JobPerformance		TC
Smith	U	40000	C	null	C	C
Brown	C	null	C	Good	C	C

(c) EMPLOYEE

Name		Salary		JobPerformance		TC
Smith	U	null	U	null	U	U

(d) EMPLOYEE

Name		Salary		JobPerformance		TC
Smith	U	40000	C	Fair	S	S
Smith	U	40000	C	Excellent	C	C
Brown	C	80000	S	Good	C	S

A mulltilevel relation to illustrate multilevel security:

A,The original employee table, (b) after filtering employee table for classification C users, (c) after filtering employee table for classification U users,(d) polyinstantation of the smith row

Comparing DAC and MAC

- **Discretionary Access Control (DAC)** policies are characterized by a high degree of flexibility, which makes them suitable for a large variety of application domains.
  - The main drawback of **DAC** models is their vulnerability to malicious attacks,.
- By contrast, mandatory policies ensure a high degree of protection in a way, they prevent any illegal flow of information.
- Mandatory policies have the drawback of being too rigid and they are only applicable in limited environments.
- In many practical situations, discretionary policies are preferred because they offer a better trade-off between security and applicability.

Role-Based Access Control

- Role-based access control (RBAC) emerged rapidly in the 1990s as a proven technology for managing and enforcing security in large-scale enterprise wide systems.
- Its basic notion is that permissions are associated with roles, and users are assigned to appropriate roles.
- Roles can be created using the CREATE ROLE and DESTROY ROLE commands.
- The GRANT and REVOKE commands discussed under DAC can then be used to assign and revoke privileges from roles.

Cont’d

- RBAC appears to be a viable alternative to traditional discretionary and mandatory access controls; it ensures that only authorized users are given access to certain data or resources.
- Many DBMSs have allowed the concept of roles, where privileges can be assigned to roles.
- Role hierarchy in RBAC is a natural way of organizing roles to reflect the organization’s lines of authority and responsibility.

Cont'd

- Another important consideration in RBAC systems is the possible temporal constraints that may exist on roles, such as time and duration of role activations, and timed triggering of a role by an activation of another role.
- Using an RBAC model is highly desirable goal for addressing the key security requirements of Web-based applications.

Introduction to Statistical Database Security

- **Statistical databases** are used mainly to produce statistics on various populations.
- The database may contain **confidential data** on individuals, which should be protected from user access.
- Users are permitted to retrieve **statistical information** on the populations, such as **averages, sums, counts, maximums, minimums,** and **standard deviations**.

Cont'd

- A **population** is a set of tuples of a relation (table) that satisfy some selection condition.
- Statistical queries involve applying **statistical functions** to a **population** of tuples.
- For example, we may want to retrieve the *number* of individuals in a **population** or the average income in the population.
  - However, statistical users are not allowed to retrieve individual data, such as the income of a specific person.

Cont’d

- Statistical database security techniques must prohibit the retrieval of individual data.
- This can be achieved by prohibiting queries that retrieve attribute values and by allowing only queries that involve statistical aggregate functions such as COUNT, SUM, MIN, MAX, AVERAGE, and STANDARD DEVIATION.
  - Such queries are sometimes called **statistical queries**.

Cont’d

- It is DBMS’s responsibility to ensure confidentiality of information about individuals, while still providing useful statistical summaries of data about those individuals to users.
- Provision of **privacy protection** of users in a statistical database is paramount.
- In some cases it is possible to **infer** the values of individual tuples from a sequence statistical queries.

Introduction to Flow Control

- **Flow control** regulates the distribution or flow of information among accessible objects.
- A **flow** between object X and object Y occurs when a program reads values from X and writes values into Y.
  - Flow controls check that information contained in some objects does not flow explicitly or implicitly into less protected objects.
- A **flow policy** specifies the channels along which information is allowed to move.

Cont'd

- Covert Channels
  - A **covert channel** allows a transfer of information that violates the security or the policy.
  - A **covert channel allows** information to pass from a higher classification level to a lower classification level through **improper means**.
- **Covert channels** can be classified into two broad categories:
  - **Storage channels** do not require any temporal synchronization, in that information is conveyed by accessing system information or what is otherwise inaccessible to the user.
  - **Timing channel** allow the information to be conveyed by the timing of events or processes.

Encryption and Public Key Infrastructures

- **Encryption** is a means of maintaining secure data in an insecure environment.
- **Encryption** consists of applying an **encryption algorithm** to data using some prespecified **encryption key**.
- The resulting data has to be **decrypted** using a **decryption key** to recover the original data.
- The **Data Encryption Standard (DES)** is a system developed by the U.S. government for use by the general public.
  - It has been widely accepted as a cryptographic standard both in the United States and abroad.
- **DES** can provide end-to-end encryption on the channel between the sender A and receiver B.

Cont'd

- **DES** algorithm is a careful and complex combination of two of the fundamental building blocks of encryption:
  - **substitution** and **permutation** (transposition).
- The **DES** algorithm derives its strength from repeated application of these two techniques for a total of 16 cycles.
  - **Plaintext** (the original form of the message) is **encrypted** as blocks of **64 bits**.
- After questioning the adequacy of **DES**, the National Institute of Standards (**NIST**) introduced the Advanced Encryption Standards (**AES**).
  - This algorithm has a block size of **128 bits** and thus takes longer time to crack.

Public Key Encryption

- In 1976 Diffie and Hellman proposed a new kind of cryptosystem, which they called **public key encryption**.
- **Public key algorithms** are based on **mathematical functions** rather than operations on bit patterns.
  - They also involve the use of **two separate keys**
- The two keys used for public key encryption are referred to as the **public key** and the **private key**.
  - the **private key** is kept secret, but it is referred to as private key rather than a secret key.

Cont'd

- A public key encryption scheme, or infrastructure, has six ingredients:
  - ✓ **Plaintext**: This is the data or readable message that is fed into the algorithm as input.
  - ✓ **Encryption algorithm**: The encryption algorithm performs various transformations on the **plaintext**.
  - ✓ **Public and private keys**: These are pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.
  - ✓ **Cipher text** : This is the scrambled message produced as output. It depends on the **plaintext** and the key.
  - ✓ **Decryption algorithm** : This algorithm accepts the **cipher text** and the matching key and produces the original **plaintext**.

Cont'd

- **Public** key is made for public and **private** key is known only by owner.
- A general-purpose public key cryptographic algorithm relies on
  - **one key for encryption** and
  - a different but related **key for decryption**.
- The essential steps are as follows:
  - ✓ Each user generates a **pair of keys** to be used for the encryption and decryption of messages.
  - ✓ Each user places one of the two keys in a public register or other accessible file. This is the **public key**.

Cont'd

Digital Signatures

- A **digital signature** is an example of using encryption techniques to provide authentication services in e-commerce applications.
- A digital signature is a means of associating a mark unique to an individual with a body of text.
- A digital signature consists of a string of symbols.
  - Signature must be different for each use.
    - This can be achieved by making each digital signature a function of the message that it is signing, together with a time stamp.
  - Public key techniques are the means creating digital signatures.

Cont'd

Cryptanalysis is the process of breaking an encrypted message *without* knowledge of the key

- Cryptology the filed that study both cryptography and cryptanalysis
- two basic components in classical ciphers:
- **substitution** and **transposition**
  - **substitution ciphers -** has letters replaced by others
  - **transposition ciphers -** has letters arranged in a different order