Chapter15 Security

related to text book chapter16 (version 7)

related to text book chapter 17 (version 8)

Contents

- Introduction
- Discretionary Access Control
  - Authorization
  - SQL Facilities
- Mandatory Access Control
- Statistical Databases
- Data Encryption

Introduction

- Security

  Protecting the data against unauthorized users

  – Discretionary
    - privileges

  – Mandatory
    - Data object labeled with classification
    - User given clearance

- Integrity

  Protecting it against authorized users

Security

- **Security** - protection from malicious attempts to steal or modify data.
  - Database system level
  - Operating system level
  - Network level
  - Physical level
  - Human level

Security – cont.

- Operating system level
  - Operating system super-users can do anything they want to the database! Good operating system level security is required.

Security – cont.

- Network level: must use encryption to prevent
  - Eavesdropping (unauthorized reading of messages)
  - Masquerading (pretending to be an authorized user or sending messages supposedly from authorized users)

Security – cont.

- Physical level
  - Physical access to computers allows destruction of data by intruders; traditional lock-and-key security is needed
  - Computers must also be protected from floods, fire, etc.
    - More in Recovery

Security – cont.

- Human level
  - Users must be screened to ensure that an authorized users do not give access to intruders
  - Users should be trained on password selection and secrecy

Security – cont.

- Database system level
  - Authentication and authorization mechanisms to allow specific users access only to required data (security subsystem)

Database-Level Security

- Assume security at network, operating system, human, and physical levels.
- Database specific issues:
  - each user may have authority to read only part of the data and to write only part of the data.
  - User authority may correspond to entire files or relations, but it may also correspond only to parts of files or relations.

Authorization

Forms of authorization on parts of the database:

- **Read authorization** - allows reading, but not modification of data.

Authorization – cont.

- **Insert authorization** - allows insertion of new data, but not modification of existing data.
- **Update authorization** - allows modification, but not deletion of data.
- **Delete authorization** - allows deletion of data

Authorization – cont.

Forms of authorization to modify the database schema:

- **Index authorization** - allows creation and deletion of indices.
- **Resources authorization** - allows creation of new relations.

Authorization – cont.

- **Alteration authorization** - allows addition or deletion of attributes in a relation.
- **Drop authorization** - allows deletion of relations.

Authorization and Views

- Ability of views to hide data serves both to simplify usage of the system and to enhance security by allowing users access only to data they need for their job

Authorization – cont.

- A combination or relational-level security and view-level security can be used to limit a user's access to precisely the data that user needs.

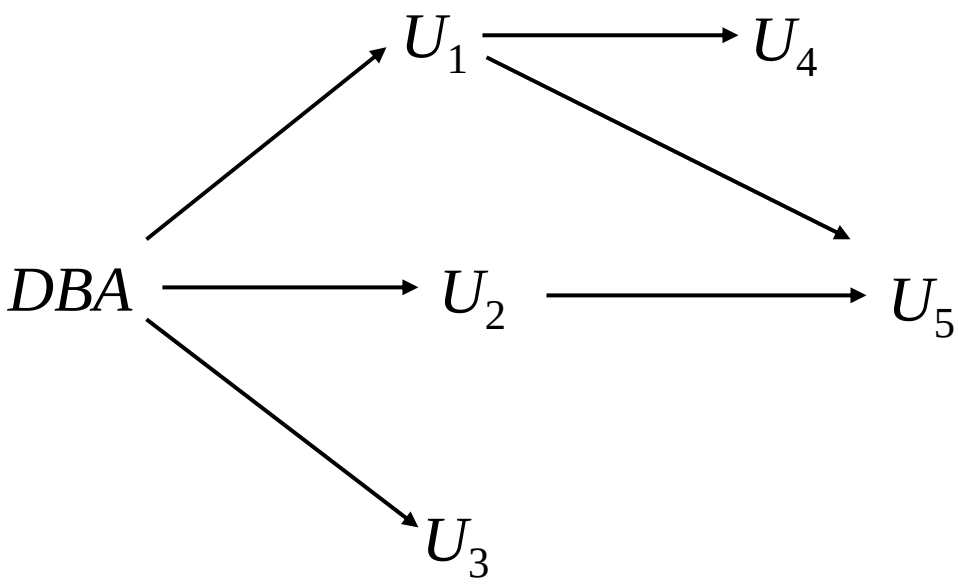
Granting of Privileges

- The passage of authorization from one user to another may be represented by an authorization graph.
- The nodes of this graph are the users.
- The root of the graph is the database administrator.

Granting of Privileges - cont.

- Consider graph for update authorization on loan.
- An edge $U_i \rightarrow U_j$ indicates that user U_i has granted update authorization on loan to U_j.

Granting of Privileges - cont.

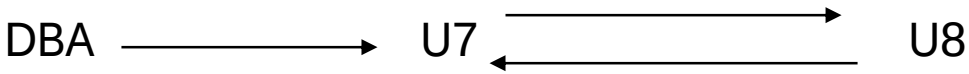


Authorization Grant Graph

- *Requirement*: All edges in an authorization graph must be part of some path originating with the database administrator
- If DBA revokes grant from U_1:
  - Grant must be revoked from U_4 since U_1 no longer has authorization
  - Grant must not be revoked from U_5 since U_5 has another authorization path from DBA through U_2

Authorization Grant Graph - cont.

- Must prevent cycles of grants with no path from the root:
  - DBA grants authorization to U_7
  - U7 grants authorization to U_8
  - U8 grants authorization to U_7
  - DBA revokes authorization from U_7



Authorization Grant Graph - cont.

- Must revoke grant U_7 to U_8 and from U_8 to U_7 since there is no path from DBA to U_7 or to U_8 anymore.

Security Specification in SQL

- The grant statement is used to confer authorization

  grant <privilege list>

  on <relation name or view name> to <user list>

Security Specification in SQL - cont.

- <user list> is:
  - a user-id
  - *public*, which allows all valid users the privilege granted
  - A role (more on this later)

Security Specification in SQL - cont.

- Granting a privilege on a view does not imply granting any privileges on the underlying relations.
- The grantor of the privilege must already hold the privilege on the specified item (or be the database administrator).

Privileges in SQL

- **select:** allows read access to relation,or the ability to query using the view
  - Example: grant users U_1, U_2, and U_3 **select** authorization on the *branch* relation:

    grant select on *branch* **to** U_1, U_2, U_3
- **insert**: the ability to insert tuples
- **update**: the ability to update using the SQL update statement

Privileges in SQL - cont.

- **delete**: the ability to delete tuples.
- **references**: ability to declare foreign keys when creating relations.
- **usage**: In SQL-92; authorizes a user to use a specified domain
- **all privileges**: used as a short form for all the allowable privileges

Privilege To Grant Privileges

- **with grant option**: allows a user who is granted a privilege to pass the privilege on to other users.
  - Example:

grant select on *branch* to U_1 with grant option

Gives U_1 the **select** privileges on branch and allows U_1 to grant this privilege to others

Roles

- Roles permit common privileges for a class of users can be specified just once by creating a corresponding “role”
- Privileges can be granted to or revoked from roles, just like user
- Roles can be assigned to users, and even to other roles

Roles - cont.

- SQL:1999 supports roles

create role *teller*
create role *manager*

grant select on *branch* **to** *teller*
grant update (*balance***) on** *account* **to** *teller*
grant all privileges on *account* **to** *manager*

grant *teller* **to** *manager*

grant *teller* **to** *alice, bob*
grant *manager* **to** *avi*

Revoking Authorization in SQL

- The **revoke** statement is used to revoke authorization.

  revoke<privilege list>

  on <relation name or view name> **from** <user list> [**restrict**|**cascade**]

- Example:

  revoke select on *branch* **from** U_1, U_2, U_3 **cascade**

Revoking Authorization in SQL - cont.

- Revocation of a privilege from a user may cause other users also to lose that privilege; referred to as cascading of the **revoke**.

Revoking Authorization in SQL - cont.

- Prevent cascading by specifying **restrict**:

  revoke select on *branch* **from** U_1, U_2, U_3 **restrict**

  With **restrict**, the **revoke** command fails if cascading revokes are required.

Revoking Authorization in SQL - cont.

- <privilege-list> may be **all to** revoke all privileges the revokee may hold.
- If <user-list> includes **public** all users lose the privilege except those granted it explicitly.

Revoking Authorization in SQL - cont.

- If the same privilege was granted twice to the same user by different grantees, the user may retain the privilege after the revocation.
- All privileges that depend on the privilege being revoked are also revoked.

Limitations of SQL Authorization

- SQL does not support authorization at a tuple level
  - E.g. we cannot restrict students to see only (the tuples storing) their own grades
- All end-users of an application may be mapped to a single database user

Limitations of SQL Authorization - cont.

- The task of authorization in above cases falls on the application program, with no support from SQL
  - Authorization must be done in application code, and may be dispersed all over an application
  - Checking for absence of authorization loopholes becomes very difficult since it requires reading large amounts of application code

Audit Trail

- A special file or database in which the system automaticlly keeps track of all operations performed by users on the regular data.
- Can physically integrated with the recovery log in some system.

Audit Trail – cont.

- Typical audit trail entry
  - request (source text)
  - terminal from which the operation was invoked
  - user who invoked the operation
  - date and time of the operation
  - relvar, tuple, attribute affected
  - old value
  - new value

Example

- User smith created table supplier, then he write statements:

  Grant select on supplier to Brown, Black;

  Grant update on supplier to Jim;

  Grant update(city) on supplier to Bob;

User Brown:

  Select * from supplier;

  update supplier set status=30 where s#=‘s5’;

Example-cont.

- User Jim:

  Update supplier set city=‘beijing’ where city=‘nanjing’;

  Delete supplier where status<10;
- User Bob:

  Update supplier set city=‘beijing’ where city=‘nanjing’;

  Grant update(city) on supplier to Alice;

Mandatory Access Control

- Based on system-wide policies that cannot be changed by individual users.
  - Each DB object is assigned a security class.
  - Each subject (user or user program) is assigned a clearance for a security class.
  - Rules based on security classes and clearances govern who can read/write which objects.

Mandatory Access - Cont.

- Mandatory controls began to receive a lot of attention in the 1990s
- Most commercial systems do not support mandatory access control. Versions of some DBMSs do support it;
- used for specialized (e.g., military、banking) applications.

Mandatory Access - Cont.

- Objects (e.g., tables, views, tuples)
- Subjects (e.g., users, user programs)
- security classes: TS(Top Secret), S (Secret), C(Classified), U (Unclassified)

  TS > S > C > U
- each subject and object are classified into one of the security classifications (TS, S, etc.)

Mandatory Access - Cont.

- Bell-LaPadulla properties (restrictions on data access)
  - simple property: No READ UP

    Subject S can ***read*** object O only if

    class(S) >= class(O) (no reads in higher security)
  - star (*) property: No WRITE DOWN

    Subject S can ***write*** object O only if

    class(S) <= class(O) (no writes in lower security)

Multilevel Relations

- multilevel relation (MLS) schema
  - classification attribute C
  - tuple classification TC
  - R(A1, C1, A2, C2, ...An, Cn, TC)

MLS Relation Example

Vessel	Objective	Destination	TC
Micra U	Shipping U	Moon U	U
Vision U	Spying U	Saturn U	U
Avenger C	Shipping C	Mars C	C
Avenger C	Shipping C	Venus S	S

MLS - cont.

- Level U sees first 2 tuples
- Level C sees first 3 tuples
- Level S sees all tuples

Orange book

- US Department of Defense *Trusted Computer System Evaluation Criteria*
- security categories range from D (Minimal Protection) to A (Verified Protection)
  - D - Minimal Protection
  - C - Discretionary Protection
    - C1 - Discretionary Security Protection
    - C2 - Controlled Access Protection

Orange book - cont.

- B - Mandatory Protection
  - B1 - Labelled Security Protection
  - B2 - Structured Protection
  - B3 - Security Domains
- A - Verified Protection
  - A1 - Verified Protection

Statistical Databases

- A database that permits queries that derive aggregated information but not queries that derive individual information.
- Problem: how to ensure privacy of individuals while allowing use of data for statistical purposes (e.g., finding median income, average bank balance etc.)

Example

- Relvar STATS

NAME	SEX	CHILDREN	OCCUPATION	SALARY	TAX	AUDITS
Alf	M	3	Programmer	50k	10k	3
Bea	F	2	Physician	130k	10k	0
Cary	F	0	Programmer	56k	18k	1
Dawn	F	2	Builder	60k	12k	1
Ed	M	2	Clerk	44k	4k	0
Fay	F	1	Artist	30k	0k	0
Guy	M	0	Lawyer	190k	0k	0
Hal	M	3	Homemaker	44k	2k	0
Ivy	F	4	Programmer	64k	10k	1
Joy	F	1	Programmer	60k	20k	1

Example – cont.

- Suppose some user U is authorized to perform statistical queries and is intent on discovering Alf's salary.
  - With (STATS Where SEX = 'M' AND
    Occupation = 'Programmer') AS x :
    Count (x)
    Result: 1
  - With (STATS Where SEX = 'M' AND
    Occupation = 'Programmer') AS x:
    Sum (x, salary)
    Result : 50k

Note: system should refuse to respond to a query its cardinality of the set summarized is less than some lower bound b.

Example – cont.

- System should also refuse to respond if that cardinality is greater than the upper bound N-b. For example:

Count (STATS)

Result: 12

With (STATS Where Not (SEX = ‘M’ AND
Occupation = ‘Programmer’)) AS x

Count (x)

Result: 11

Sum (STATS, Salary)

Result: 728K

With (STATS Where Not (SEX = ‘M’ AND
Occupation = ‘Programmer’)) AS x

Sum (x, Salary)

Result : 678K. **728K-678K = 50K.**

Example – cont.

- To restricting the cardinality c in the range

  $b \leqslant c \leqslant N-b$ is inadequately. For example:

  let b =2 for STATS application, there has the **boolean expression: 'Sex = 'M' AND occupation = 'Programmer'**

  With (STATS Where Sex = 'M') AS x

  Count (x) Sum (x, Salary)

  Result: 4 Result : 328K

  With (STATS Where Sex = 'M' AND Not

  (occupation = 'Programmer')) AS x:

  Count (x) Sum (x, Salary)

  Result: 3 Result: 278K

328K - 278K = 50K

- 'Sex = 'M' AND occupation = 'Programmer' is a **individual tracker** for Alf.

Encryption

- Data may be *encrypted* when database authorization provisions do not offer sufficient protection.
  - Plaintext
  - Encryption algorithm
  - Encryption key
  - Ciphertext

Encryption – cont.

- Properties of good encryption technique:
  - Relatively simple for authorized users to encrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of encryption key.
  - Extremely difficult for an intruder to determine the encryption key.

Example

- Plaintext:

  AS KINGFISHERS CATCH FIRE

- Encryption key

  ELIOT

- Encryption algorithm
  - Divide plaintext into blocks of length equal to that of encryption key (blank shown explicitly as ”+”)

    AS+KI NGFIS HERS+ CATCH +FIRE

  - Replace each character of the paintext by an integer in the range 00-26

0119001109 1407060919 0805181900 0301200308 0006091805

Example – cont.

- Repeat step 2 for the encryption key

  0512091520

- for each block of the plaintext, replace each character by the sum modulo 27 of the integer encoding of the corresponding character of the encryption key:

0119001109 1407060919 0805181900 0301200308 0006091805
0512091520 0512091520 0512091520 0512091520 0512091520

0604092602 1919152412 1317000720 0813021801 0518180625

- Replace each integer encoding in the result of step 4 by its character equivalent:

  FDIZB SSOXL MQ+GT HMBRA ERRFY

Encryption – cont.

- *Data Encryption Standard* (DES) substitutes characters and rearranges their order on the basis of an encryption key which is provided to authorized users via a secure mechanism. Scheme is no more secure than the key transmission mechanism since the key has to be shared.(1977 US Federal standards)

Encryption – cont.

- Advanced Encryption Standard (AES) is a new standard replacing DES, and is based on the Rijndael algorithm, but is also dependent on shared secret keys

Encryption – cont.

- *Public-key encryption* is based on each user having two keys:
  - ***public key*** – publicly published key used to encrypt data, but cannot be used to decrypt data
  - ***private key*** -- key known only to individual user, and used to decrypt data.
    Need not be transmitted to the site doing encryption.

  Encryption scheme is such that it is impossible or extremely hard to decrypt data given only the public key.

Encryption – cont.

- The RSA public-key encryption scheme is based on the hardness of factoring a very large number (100's of digits) into its prime components.

Exercises

- 1