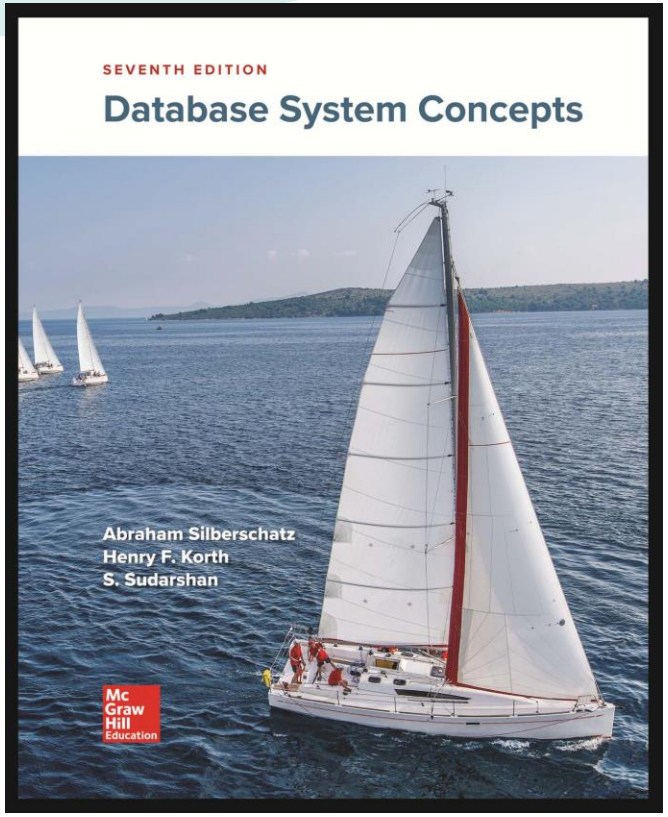
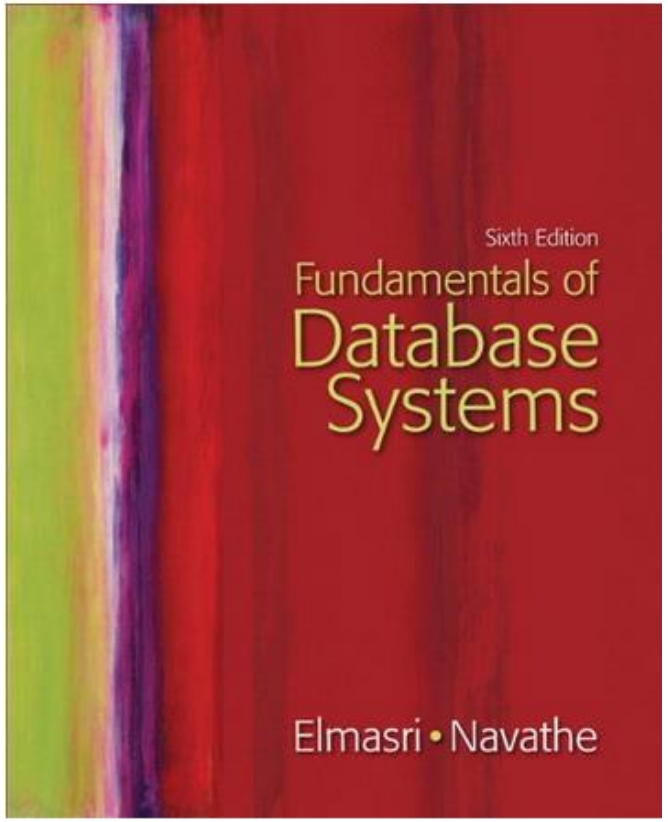
IF3140 - Sistem Basis Data

Database Security

KNOWLEDGE & SOFTWARE ENGINEERING





References

Ramez Elmasri, Shamkant B. Navathe : "**Fundamentals of Database Systems**", 6th Edition

- Chapter 24 : Database Security

Abraham Silberschatz, Henry F. Korth, S. Sudarshan : "**Database System Concepts**", 7th Edition

- Chapter 4.7 : Authorization
- Chapter 9.8 : Application Security
- Chapter 9.9 : Encryption and Its Application

KNOWLEDGE & SOFTWARE ENGINEERING

Objectives

Understand important properties of security in a Database System

Evaluate access controls of a specified database by using authorization-grant graph

Manage users of databases with specified access controls related with a case study

KNOWLEDGE & SOFTWARE ENGINEERING

Outline

Database Security and Authorization

Access Control

Introduction to Statistical Database Security

Introduction to Flow Control

Encryption and Public Key Infrastructures

Application Security

KNOWLEDGE & SOFTWARE ENGINEERING

Database Security and Authorization

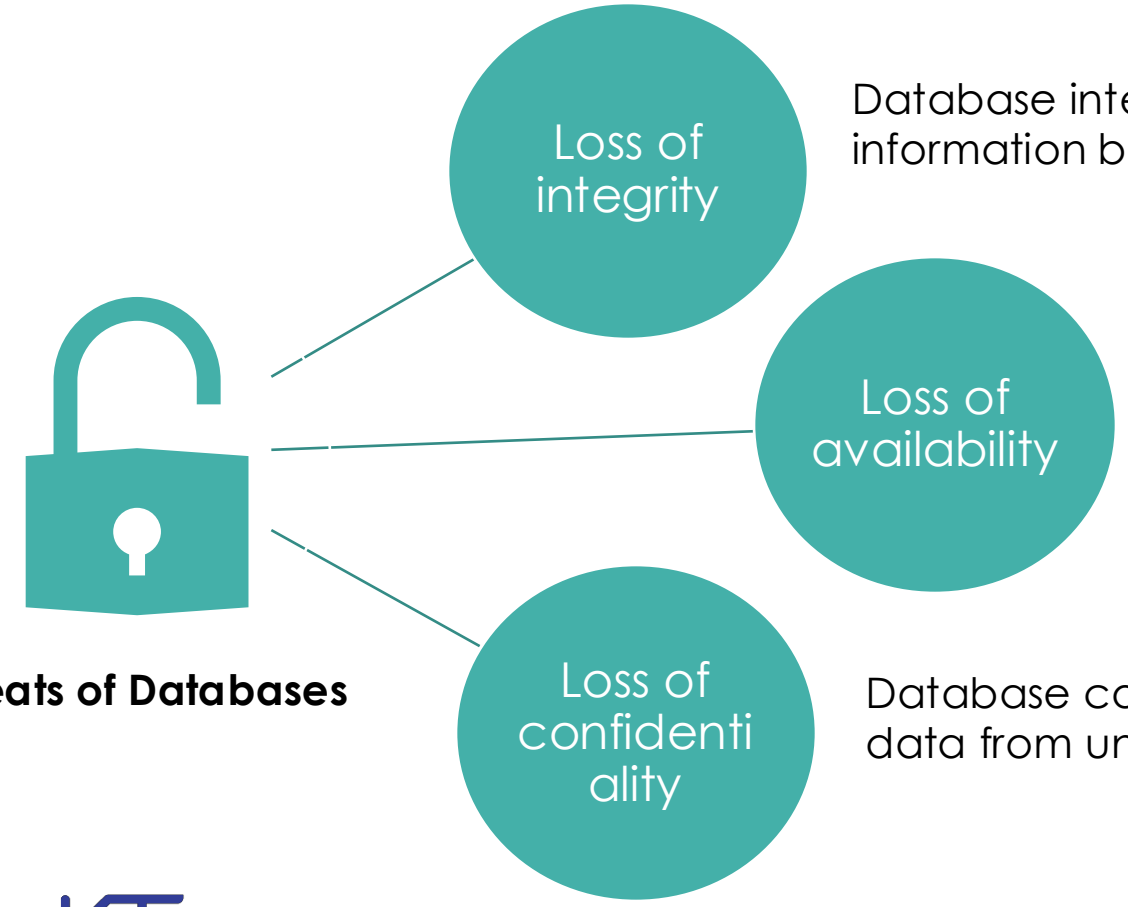
KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Database Security Issues (1/4)

Database security addresses many issues, including the following

- Various **legal and ethical issues** regarding the right to access certain information
- **Policy issues** at the governmental, institutional, or corporate level to what kinds of information should not be made publicly available
- **System-related issues** such as the system levels at which various security functions should be enforced
- The need in some organizations to **identify multiple security levels** and to categorize the data and users based on these classifications

Introduction to Database Security Issues (2/4)



Threats of Databases

Database integrity refers to the requirement that information be protected from improper modifications

Database availability refers to making objects available to a human user or a program to which they a legitimate right

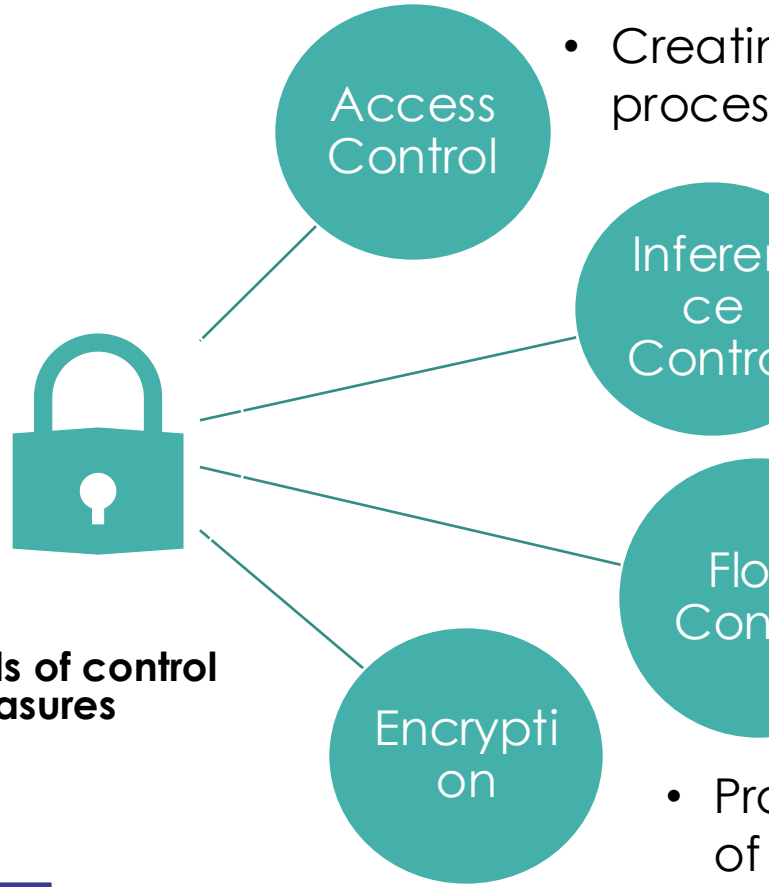
Database confidentiality refers to the protection of data from unauthorized disclosure

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Database Security Issues (3/4)

intinya definisiin data apa aja yg boleh diakses

- Provisions for restricting access to the database as a whole.
- Creating user accounts and passwords to control login process by the DBMS.



Four kinds of control measures

- Counter measures to statistical database security problem.
- Prohibits the retrieval of individual data.

- Prevents information from flowing in such a way that it reaches unauthorized users.

jgn sampe mengalir ke unauthorized user

- Protects sensitive data that is transmitted via some type of communications network.
- The data is encoded using some encoding algorithm.

kalo diencrypt walaupun org dpt datanya, gaakan gampang diencode

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Database Security Issues (4/4)

A DBMS typically includes a database security and authorization subsystem that is responsible for ensuring the security portions of a database against unauthorized access.

Two types of database security mechanisms:

Discretionary security mechanisms

grant privileges to users, including the capability to access specific data files, records, or fields in a specified mode (such as read, insert, delete, or update).

kasih privilege ke user

Mandatory security mechanism

enforce multilevel security by classifying the data and users into various security classes (or levels) and then implementing the appropriate security policy of the organization.

KNOWLEDGE & SOFTWARE ENGINEERING

Database Security and the DBA (1/2)

- The database administrator (DBA) is the **central authority for managing a database system.** Responsible for the overall security of the 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

KNOWLEDGE & SOFTWARE ENGINEERING

Database Security and the DBA (2/2)

- 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:
    1. Account creation — Access Control
    2. Privilege granting — Discretionary
    3. Privilege revocation — Discretionary
    4. Security level assignment — Mandatory Security

KNOWLEDGE & SOFTWARE ENGINEERING

Access Protection, User Accounts, and Database Audits

A database log that is used mainly for security purposes is sometimes called an audit trail.

Whenever a person or group of persons need to access a database system, the individual or group must **first apply for a user account.**

- The DBA will then create a new account id and password for the user

The user must log in to the DBMS by **entering account id and password** whenever database access is needed.

The database system must **keep track of all operations** on the database that are applied by a certain user throughout each login session.

- To keep a record of all updates applied to the database and of the particular user who applied each update, that may be required for recovery from a transaction failure or system crash.

If any tampering with the database is suspected, a **database audit** is performed

- A database audit consists of reviewing the log to examine all accesses and operations applied to the database during a certain time period.

harus ditrack user apa bisa akses data apa pas kapan

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control
Discretionary Access Control (DAC)

Types of Discretionary Privileges (1/2)

The typical method of enforcing discretionary access control in a database system is based on the granting and revoking privileges.

atur siapa yg bisa melakukan ini

The **account level**; 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.

The **relation level** (or **table level**, includes **base relations** and virtual (**view**) relations):

- At this level, the DBA can control the privilege to access each individual relation or view in the database.

KNOWLEDGE & SOFTWARE ENGINEERING

Types of Discretionary Privileges (2/2)

Granting and revoking privilege generally follow an authorization model for discretionary privileges known as access matrix model.

Columns of matrix M represents objects (relations, records, columns, views, operations)

	Relation X	Relation Y	View Z
User A	Read	Owner	Owner
User B	Owner		Read
Program C	Read, write	Read, write	

Rows of matrix M represents subjects (users, accounts, programs)

M(i,j) represents the type of privileges (read, write, update) that subject i holds on object j

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control
Mandatory Access Control (MAC)

KNOWLEDGE & SOFTWARE ENGINEERING

Mandatory Access Control and Role-Based Access Control for Multilevel Security

- 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, an **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.

KNOWLEDGE & SOFTWARE ENGINEERING

Mandatory Access Control and Role-Based Access Control for Multilevel Security(2)

- 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 ≥ S ≥ C ≥ U

semua orang bisa akses

- 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, TS, S, C, or U:
  - **Clearance** (classification) of a subject S as **class(S**) and to the **classification** of an object O as **class(O)**.

KNOWLEDGE & SOFTWARE ENGINEERING

Mandatory Access Control and Role-Based Access Control for Multilevel Security(3)

- Two restrictions are enforced on data access based on the subject/object classifications:

data yg lebih 'umum'
gabole baca data yg lebih 'secret'

  - **Simple security property:** A subject S is not allowed read access to an object O unless class(S) ≥ class(O).
  - A subject S is not allowed to write an object O unless class(S) ≤ class(O). This known as the **star property** (or * property).

contoh
S_x = subject x
O_x = object x

	read	write
class (S1) = S		
class(O1) = S	v	v
class(O2) = U	v	x
class (O3) = TS	x	v

Mandatory Access Control and Role-Based Access Control for Multilevel Security(4)

Example:

(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

A multilevel relation to illustrate multilevel security. (a) The original EMPLOYEE tuples. (b) Appearance of EMPLOYEE after filtering for classification C users. (c) Appearance of EMPLOYEE after filtering for classification U users.

KNOWLEDGE & SOFTWARE ENGINEERING

Comparing Discretionary Access Control and Mandatory Access Control

In many practical situations, discretionary policies are preferred because they offer a better trade-off between security and applicability.

Policies	Advantages	Drawbacks
Discretionary Access Control (DAC)	Characterized by a high degree of flexibility, which makes them suitable for a large variety of application domains.	Main drawback of **DAC** models is their vulnerability to malicious attacks, such as Trojan horses embedded in application programs.
Mandatory Access Control (MAC) harus dibuat sendiri aplikasinya	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.

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control
Role-Based Access Control (RBAC)

KNOWLEDGE & SOFTWARE ENGINEERING

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.

```
GRANT ROLE full_time TO employee_type1
GRANT ROLE intern TO employee_type2
```

KNOWLEDGE & SOFTWARE ENGINEERING

Role-Based Access Control(2)

- **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.

KNOWLEDGE & SOFTWARE ENGINEERING

Role-Based Access Control(3)

- 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.**
- In contrast, discretionary access control (**DAC**) and mandatory access control (**MAC**) models **lack capabilities** needed to support the security requirements emerging enterprises and Web-based applications.

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control in SQL

SLIDES ARE TAKEN FROM: DATABASE SYSTEMS CONCEPTS, 7TH EDITION

KNOWLEDGE & SOFTWARE ENGINEERING

Authorization

Forms of authorization on parts of the database:

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

Forms of authorization to modify the database schema

- **Index** - allows creation and deletion of indices.
- **Resources** - allows creation of new relations.
- **Alteration** - allows addition or deletion of attributes in a relation.
- **Drop** - allows deletion of relations.

KNOWLEDGE & SOFTWARE ENGINEERING

Authorization Specification in SQL

- The **grant** statement is used to confer authorization

  grant <privilege list>
  on <relation name or view name> **to** <user list>

- <user list> is:
  - a user-id
  - **public**, which allows all valid users the privilege granted
  - A role
- 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).

KNOWLEDGE & SOFTWARE ENGINEERING

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 *instructor* relation:

    grant select on *instructor* **to** U_1, U_2, U_3

- **insert**: the ability to insert tuples
- **update**: the ability to update using the SQL update statement
- **delete**: the ability to delete tuples.
- **all privileges**: used as a short form for all the allowable privileges

KNOWLEDGE & SOFTWARE ENGINEERING

Revoking Authorization in SQL

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

  revoke <privilege list>
  on <relation name or view name> **to** <user list>

- Example:

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

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

KNOWLEDGE & SOFTWARE ENGINEERING

Roles

create role instructor;
grant *instructor* **to Amit;**

- Privileges can be granted to roles:

  grant select on *takes* **to** *instructor*;

- Roles can be granted to users, as well as to other roles

  create role *teaching_assistant*
  grant *teaching_assistant* **to** *instructor*;
  Instructor inherits all privileges of *teaching_assistant*

- Chain of roles

  create role *dean*;
  grant *instructor* **to** *dean*;
  grant *dean* **to** Satoshi;

KNOWLEDGE & SOFTWARE ENGINEERING

Authorization on Views

- **create view** *geo_instructor* **as**
  (**select** *
  from *instructor*
  where *dept_name* = 'Geology');
- **grant select on** *geo_instructor* **to** *geo_staff*
- Suppose that a *geo_staff* member issues
  - **select** *
    from *geo_instructor;*
- What if
  - *geo_staff* does not have permissions on *instructor?*
  - creator of view did not have some permissions on *instructor?*

KNOWLEDGE & SOFTWARE ENGINEERING

Other Authorization Features

- **references** privilege to create foreign key

```
grant reference (dept_name) on department to Mariano;
```

- **transfer** of privileges

```
grant select on department to Amit with grant option;
```

Other Authorization Features

- **revoking** of privileges

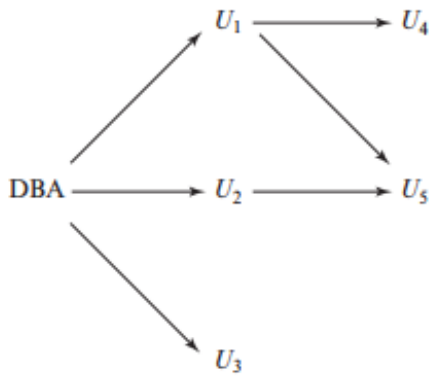


Figure 4.11 Authorization-grant graph (U_1, U_2, ..., U_5 are users and DBA refers to the database administrator).

revoke select on *department* **from** Amit, Satoshi **cascade**;

revoke select on *department* **from** Amit, Satoshi **restrict**;

Etc. read Section 4.7 for more details we have omitted here.

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control Policies for E-Commerce and the Web

- **E-Commerce environments** require elaborate policies that go beyond traditional DBMSs.
  - In an e-commerce environment the resources to be protected are not only traditional data but also knowledge and experience.
  - The access control mechanism should be flexible enough to support a wide spectrum of heterogeneous protection objects.
- A related requirement is the support for **content-based access-control**.

KNOWLEDGE & SOFTWARE ENGINEERING

Access Control Policies for E-Commerce and the Web(2)

- Another requirement is related to the heterogeneity of subjects, which requires access control policies based on user characteristics and qualifications.
  - A possible solution, to better take into account user profiles in the formulation of access control policies, is to support the notion of credentials.
  - A **credential** is a set of properties concerning a user that are relevant for security purposes
    - For example, age, position within an organization
  - It is believed that the XML language can play a key role in access control for e-commerce applications.

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Statistical Database Security

KNOWLEDGE & SOFTWARE ENGINEERING

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**.
- 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.

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Statistical Database Security(2)

- 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.
- 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**.

KNOWLEDGE & SOFTWARE ENGINEERING

Introduction to Statistical Database Security(3)

- 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.
  - This is particularly true when the conditions result in a population consisting of a small number of tuples.

Introduction to Flow Control

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.
  - The simplest flow policy specifies just two classes of information:
    - confidential (C) and nonconfidential (N)
  - and allows all flows except those from class C to class N.

KNOWLEDGE & SOFTWARE ENGINEERING

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.
- Some security experts believe that one way to avoid covert channels is for programmers to not actually gain access to sensitive data that a program is supposed to process after the program has been put into operation.

Encryption

SLIDES ARE TAKEN FROM: DATABASE SYSTEMS CONCEPTS, 7TH EDITION

KNOWLEDGE & SOFTWARE ENGINEERING

Encryption

- Data may be *encrypted* when database authorization provisions do not offer sufficient protection.
- Properties of good encryption technique:
  - Relatively simple for authorized users to encrypt and decrypt data.
  - Encryption scheme depends not on the secrecy of the algorithm but on the secrecy of a parameter of the algorithm called the encryption key.
  - Extremely difficult for an intruder to determine the encryption key.
- **Symmetric-key encryption**: same key used for encryption and for decryption
- **Public-key encryption** (a.k.a. **asymmentric-key encryption**): use different keys for encryption and decryption
  - encryption key can be public, decryption key secret

KNOWLEDGE & SOFTWARE ENGINEERING

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.
- 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.
- *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.
- 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

KNOWLEDGE & SOFTWARE ENGINEERING

Encryption (Cont.)

- **Hybrid schemes** combining public key and private key encryption for efficient encryption of large amounts of data
- Encryption of small values such as identifiers or names vulnerable to **dictionary attacks**
  - especially if encryption key is publicly available
  - but even otherwise, statistical information such as frequency of occurrence can be used to reveal content of encrypted data
  - Can be deterred by adding extra random bits to the end of the value, before encryption, and removing them after decryption
    - same value will have different encrypted forms each time it is encrypted, preventing both above attacks
    - extra bits are called **salt bits**

KNOWLEDGE & SOFTWARE ENGINEERING

Encryption in Databases

- Database widely support encryption
- Different levels of encryption:
  - **disk block**
    - every disk block encrypted using key available in database-system software.
    - Even if attacker gets access to database data, decryption cannot be done without access to the key.
  - **Entire relations, or specific attributes of relations**
    - non-sensitive relations, or non-sensitive attributes of relations need not be encrypted
    - however, attributes involved in primary/foreign key constraints cannot be encrypted.
- Storage of encryption or decryption keys
  - typically, single master key used to protect multiple encryption/decryption keys stored in database
- Alternative: encryption/decryption is done in application, before sending values to the database

Encryption and Authentication

- Password based authentication is widely used, but is susceptible to sniffing on a network.
- **Challenge-response** systems avoid transmission of passwords
  - DB sends a (randomly generated) challenge string to user.
  - User encrypts string and returns result.
  - DB verifies identity by decrypting result
  - Can use public-key encryption system by DB sending a message encrypted using user's public key, and user decrypting and sending the message back.
- **Digital signatures** are used to verify authenticity of data
  - E.g., use private key (in reverse) to encrypt data, and anyone can verify authenticity by using public key (in reverse) to decrypt data. Only holder of private key could have created the encrypted data.
  - Digital signatures also help ensure **nonrepudiation:** sender cannot later claim to have not created the data

Application Security

SLIDES ARE TAKEN FROM: DATABASE SYSTEMS CONCEPTS, 7TH EDITION

KNOWLEDGE & SOFTWARE ENGINEERING

SQL Injection

- Suppose query is constructed using
  - "select * from instructor where name = '" + name + "'"
- Suppose the user, instead of entering a name, enters:
  - X' or 'Y' = 'Y
- then the resulting statement becomes:
  - "select * from instructor where name = '" + "X' or 'Y' = 'Y" + "'"
  - which is:
    - select * from instructor where name = 'X' or 'Y' = 'Y'
  - User could have even used
    - X'; update instructor set salary = salary + 10000; --
- Prepared statement internally uses:
  "select * from instructor where name = 'X\' or \'Y\' = \'Y'
- **Always use prepared statements, with user inputs as parameters**
- Is the following prepared statemen secure?
  - conn.prepareStatement("select * from instructor where name = '" + name + "'")

Cross Site Scripting

- HTML code on one page executes action on another page
  - E.g. <img src = http://mybank.com/transfermoney?amount=1000&toaccount=14523>
  - Risk: if user viewing page with above code is currently logged into mybank, the transfer may succeed
  - Above example simplistic, since GET method is normally not used for updates, but if the code were instead a script, it could execute POST methods
- Above vulnerability called **cross-site scripting (XSS)** or **cross-site request forgery (XSRF or CSRF)**
- **Prevent your web site from being used to launch XSS or XSRF attacks**
  - Disallow HTML tags in text input provided by users, using functions to detect and strip such tags
- **Protect your web site from XSS/XSRF attacks launched from other sites**
  - ..next slide

KNOWLEDGE & SOFTWARE ENGINEERING

Cross Site Scripting

- **Protect your web site from XSS/XSRF attacks launched from other sites**
  - Use **referer** value (URL of page from where a link was clicked) provided by the HTTP protocol, to check that the link was followed from a valid page served from same site, not another site
  - Ensure IP of request is same as IP from where the user was authenticated
    - prevents hijacking of cookie by malicious user
  - Never use a GET method to perform any updates
    - This is actually recommended by HTTP standard

KNOWLEDGE & SOFTWARE ENGINEERING

Password Leakage

- Never store passwords, such as database passwords, in clear text in scripts that may be accessible to users
  - E.g. in files in a directory accessible to a web server
    - Normally, web server will execute, but not provide source of script files such as file.jsp or file.php, but source of editor backup files such as file.jsp~, or .file.jsp.swp may be served
- Restrict access to database server from IPs of machines running application servers
  - Most databases allow restriction of access by source IP address

KNOWLEDGE & SOFTWARE ENGINEERING

Application Authentication

- Single factor authentication such as passwords too risky for critical applications
  - guessing of passwords, sniffing of packets if passwords are not encrypted
  - passwords reused by user across sites
  - spyware which captures password
- Two-factor authentication
  - e.g. password plus one-time password sent by SMS
  - e.g. password plus one-time password devices
    - device generates a new pseudo-random number every minute, and displays to user
    - user enters the current number as password
    - application server generates same sequence of pseudo-random numbers to check that the number is correct.

Application Authentication

- **Man-in-the-middle** attack
  - E.g. web site that pretends to be mybank.com, and passes on requests from user to mybank.com, and passes results back to user
  - Even two-factor authentication cannot prevent such attacks
- Solution: authenticate Web site to user, using digital certificates, along with secure http protocol
- **Central authentication** within an organization
  - application redirects to central authentication service for authentication
  - avoids multiplicity of sites having access to user's password
  - LDAP or Active Directory used for authentication

KNOWLEDGE & SOFTWARE ENGINEERING

Single Sign-On

- **Single sign-on** allows user to be authenticated once, and applications can communicate with authentication service to verify user's identity without repeatedly entering passwords
- **Security Assertion Markup Language (SAML)** standard for exchanging authentication and authorization information across security domains
  - e.g. user from Yale signs on to external application such as acm.org using userid joe@yale.edu
  - application communicates with Web-based authentication service at Yale to authenticate user, and find what the user is authorized to do by Yale (e.g. access certain journals)
- **OpenID** standard allows sharing of authentication across organizations
  - e.g. application allows user to choose Yahoo! as OpenID authentication provider, and redirects user to Yahoo! for authentication

KNOWLEDGE & SOFTWARE ENGINEERING

Application-Level Authorization

- Current SQL standard does not allow fine-grained authorization such as "students can see their own grades, but not other's grades"
  - Problem 1: Database has no idea who are application users
  - Problem 2: SQL authorization is at the level of tables, or columns of tables, but not to specific rows of a table
- One workaround: use views such as

  create view *studentTakes* **as**
  select *
  from *takes*
  where *takes.ID = syscontext.user_id*()
  - where syscontext.user_id() provides end user identity
    - end user identity must be provided to the database by the application
  - Having multiple such views is cumbersome

KNOWLEDGE & SOFTWARE ENGINEERING

Application-Level Authorization (Cont.)

- Currently, authorization is done entirely in application
- Entire application code has access to entire database
  - large surface area, making protection harder
- Alternative: **fine-grained (row-level) authorization** schemes
  - extensions to SQL authorization proposed but not currently implemented
  - Oracle Virtual Private Database (VPD) allows predicates to be added transparently to all SQL queries, to enforce fine-grained authorization
    - e.g. add *ID= sys_context.user_id()* to all queries on student relation if user is a student

KNOWLEDGE & SOFTWARE ENGINEERING

Audit Trails

- Applications must log actions to an audit trail, to detect who carried out an update, or accessed some sensitive data
- Audit trails used after-the-fact to
  - detect security breaches
  - repair damage caused by security breach
  - trace who carried out the breach
- Audit trails needed at
  - Database level, and at
  - Application level

KNOWLEDGE & SOFTWARE ENGINEERING

End of Chapter

KNOWLEDGE & SOFTWARE ENGINEERING