11 - Security and Administration

School of Computer Science
University of Windsor

Dr. Shafaq Khan

Announcements

- **Final Report**

  Submission deadline: Sec 1 & 4: Jul 30; Sec 2: Jul 31; Sec 3: Aug 1

- **Project presentations –** Next Week

- Bonus marks for paper **submission to a conference (after my approval)** by **Aug 7**, 2023: 5 marks

Test 2

•Date: **Saturday, August 5**

• Time: **10:00 am – 11:00 am** . The total time length of the test is 1 hour.

• Location: **ER1120**

• Maximum credits/points: 100 points (this accounts for **20%** of your overall grade).

• Portion: **Chapter 7 to 11**

• Format: This is an **in-person**, paper-based, closed-book and closed-notes assessment. You are not allowed to use textbooks, your notes, a computer, or any other materials during the test. Cell phones, laptops, and all other devices must be turned off. **Multiple-choice questions and essay questions.**

• The Scantron software and hardware is used for computerized test scoring of MCQs. Don't forget to **bring No. 2 pencil** to fill the scantron sheet.

• Fill in circles on the scantron sheet completely. If you want to change an answer when completing a test, you must erase the answer you provided completely and not simply put an X through it.

• Please notify me before **5th August**, if any of your assessment has not been graded. **After 5th August no assessment reviews will be accepted.**

Agenda

- **Lecture**
  - Introduction
  - Threats and countermeasures
  - Discretionary Access Control Based on Granting and Revoking of Privileges
  - Mandatory Access Control and Role-Based Access Control for Multilevel Security
  - SQL Injection
  - Statistical Database Security
  - Flow Control
  - Public Key Infrastructure
  - Privacy Issues and Preservation
  - Challenges

Introductory Questions

Why database security is a serious concern for an organization?

What are the types of threat that can affect a database system?

What are the database counter measures to mitigate the risks?

What are the challenges to maintaining database security?

Database Threats

Threat: Any situation or event, whether intentional or accidental, that may adversely affect a system and consequently the organization.

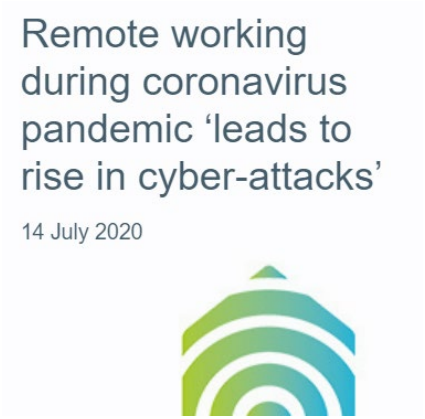
- **Loss of Confidentiality (secrecy):** Protection of data from unauthorized disclosure.
- **Loss of Integrity:** The requirement that information be protected from improper modification
- **Loss of Availability:** Making objects available to a human user or a program to which they have a legitimate right.

Together known as CIA-triad

- **Loss of privacy:** Need to protect data about individuals. Loss of privacy could lead to legal action being taken against the organization.

Introduction to Database Security

- **Database security** refers to the range of tools, controls, and measures designed to establish and preserve database confidentiality, integrity, and availability.



the **Privacy Act,** which covers how the federal government handles personal information;
the **Personal Information Protection and Electronic Documents Act (PIPEDA),** which covers how businesses handle personal information.
Canadian anti-spam law, **Canada's Anti-Spam Legislation**, SC 2010 c 23 ('CASL')

https://portswigger.net/daily-swig/canada

Database Countermeasures —Controls

Types of database control measures

- ✓ **Authentication**: A mechanism that determines whether a user is who he or she claims to be. Identification plus verification
- ✓ **Access controls:** Controlling who is authorized to accesses the database. Authorized to connect, login, select, delete, update or drop
- ✓ **Encryption:** The encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key.
- ✓ **RAID (Redundant Array of Independent Disks)**: the DBMS should continue to operate even if one of the hardware components fails. This suggests having redundant components that can be seamlessly integrated into the working system whenever there is one or more component failures.
- ✓ **Inference control:** Preventing inferring from statistical queries
- ✓ **Views:** A powerful and flexible security mechanism by hiding parts of the database from certain users.
- ✓ **Backup and recovery**: The process of periodically copying of the database and log file (and possibly programs) to offline storage media, to enable recovery in the event of a failure.
- ✓ **Integrity**: Maintaining a secure database system by preventing data from becoming invalid, and hence giving misleading or incorrect results. Can be applied through domain, entity Integrity, referential Integrity & key constraints

Database Security and the DBA

Database administrator (DBA)

Central authority for administering the database system

Major operations of the DBA related to Data Security

- Account creation
- Privilege granting
- Privilege revocation
- Security level assignment

Database Security Mechanisms

- **Discretionary (DAC) security mechanisms**

  Used to grant privileges to users

- **Mandatory (MAC) security mechanisms**

  Classify data and users into various security classes

  Implement security policy

  Control flow of data

- **Role-based security**

  Grant privileges based on various roles

Discretionary Access Control (DAC)

Source: https://www.digibrains.com/access-controls/

Discretionary Access Control (DAC)

DAC is one in which the owner of a resource grants permission on that resource to another subject, or to a group of subjects, at his own discretion.

A general approach to DAC is that of an access matrix.

	Personnel file	Purchasing records	Contracts	Invoices
John	Owner	Owner	Owner	Owner
Mary	Read, Write, Create		Read, Write	
Sue	Read	Read	Read	Read
Felix			Read, Write, Create	Read, Write, Create, Delete

Two levels for assigning privileges to use database system

Account level

DBA sets the privileges of each account holder independently of the relations in the database

Relation(or table) level

The DBA has the option to control the access of each table in the database

Discretionary Access Control ..

Relation or table level

Each relation R is assigned an owner of account

Owner of a relation is given all privileges on that relation by the DBA

GRANT CREATETAB **TO** A // User A is authorized to create tables by the DBA and let us assume creates(owns) the EMPLOYEE table

Owner can grant privileges to other users on any owned relation

Ex: If EMPLOYEE is currently owned by user A, A can grant the privileges to user B with the following SQL statement:

GRANT SELECT **ON** EMPLOYEE **TO** B // UPDATE, INSERT and DELETE privileges can also be given

Revocation and Propagation of Privileges

Revoking of Privileges

Useful for granting a privilege temporarily

REVOKE command used to cancel a privilege

IF User A wants to revoke the privileges granted previously to user B

REVOKE SELECT **ON** EMPLOYEE **TO** B;

Propagation of privileges using the GRANT OPTION

If GRANT OPTION is given from user A to user B, User B can grant privilege to other accounts

GRANT SELECT **ON** EMPLOYEE **TO** B WITH GRANT OPTION;

User B can now propagate or grant those privileges to other users (ex: user C)

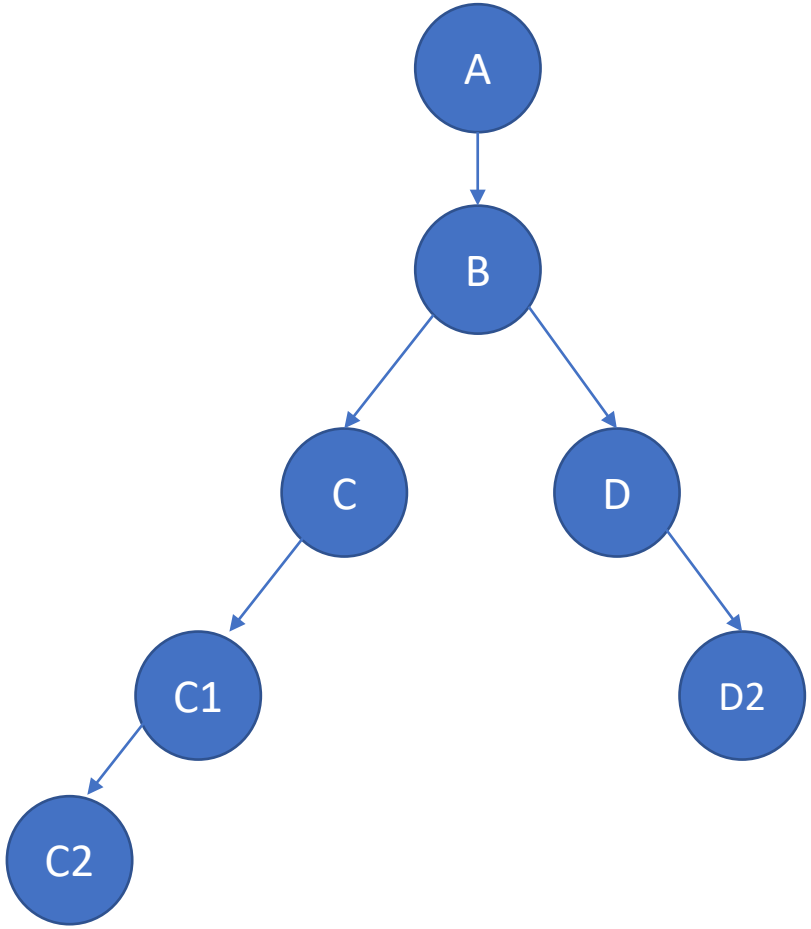
GRANT SELECT **ON** EMPLOYEE **TO C** WITH GRANT OPTION; // From B->C

DBMS must keep track of how privileges were granted if DBMS allows propagation

Propagation of Privileges ..

- **Horizontal and vertical propagation limits**
- Limiting horizontal propagation to an integer number :
- Account *B* given the GRANT OPTION (From Account A) can grant privilege to at most *i* other accounts (B->C, B->D,B->E)
  - **GRANT** SELECT **ON** EMPLOYEE, DEPARTMENT **TO** B **WITH GRANT OPTION HORIZONTAL 3**;
- **Vertical propagation limits the depth of the granting of privileges**
- B->C->D->E is possible if the vertical depth is defined to be 3, however, B->C->D->E->F is not possible
  - **GRANT** SELECT **ON** EMPLOYEE, DEPARTMENT **TO** B **WITH GRANT OPTION VERTICAL 3**
- **Both Horizonal and Vertical Limits of Propagation**
  - **GRANT** SELECT **ON** EMPLOYEE, DEPARTMENT **TO B WITH GRANT OPTION HORIZONTAL 2 VERTICAL 3**

GRANT SELECT **ON** EMPLOYEE, DEPARTMENT **TO B WITH GRANT OPTION HORIZONTAL 2 VERTICAL 3**



User A grants privileges to user B with propagation rights, and with limits on horizontal propagation(=2) and vertical propagation (=3)

Examples of GRANT and REVOKE

- Suppose the DBA creates four accounts—A1, A2, A3, and A4—and wants only A1 to be able to create base relations. To do this, the DBA must issue the following SQL statement.
  - **GRANT** CREATETAB **TO** A1;
- Suppose account A1 wants to grant to account A2(with no propagation rights) the privilege to insert and delete tuples in both EMPLOYEE and DEPARTMENT tables
  - **GRANT** INSERT, DELETE **ON** EMPLOYEE, DEPARTMENT **TO** A2;
- Suppose that A1 wants to allow account A3 to retrieve information from either of the two tables and also to be able to propagate the SELECT privilege to other accounts. A1 can issue the following command:
  - **GRANT** SELECT **ON** EMPLOYEE, DEPARTMENT **TO** A3 **WITH GRANT OPTION**;
  - A3 can consequently grant the SELECT privilege on the EMPLOYEE(or DEPARTMENT) relation to A4 by issuing the following command: **GRANT** SELECT **ON** EMPLOYEE **TO** A4;
- Suppose A1 decides to revoke the SELECT privilege of A3 on the EMPLOYEE
  - **REVOKE** SELECT **ON** EMPLOYEE **FROM** A3; // However, A3 still has privileges over DEPARTMENT

Update, Delete and Insert (Grant and Revoke)

- Suppose A1 wants to allow A4 to update any attribute of EMPLOYEE; A1 can issue the following command:
  - **GRANT** UPDATE **ON** EMPLOYEE **TO** A4 // REVOKE UPDATE **ON** EMPLOYEE FROM A4
- Suppose A1 wants to allow A4 to update only the Salary attribute of EMPLOYEE; A1 can issue the following command:
  - **GRANT** UPDATE **ON** EMPLOYEE (Salary) **TO** A4; // REVOKE UPDATE **ON** EMPLOYEE(Salary) FROM A4
- Suppose A1 wants to allow A3 to delete any tuple of DEPARTMENT; A1 can issue the following command:
  - **GRANT** DELETE **ON** DEPARTMENT **TO** A3; // REVOKE DELETE **ON** DEPARTMENT FROM A3
- Suppose A1 wants to allow A3 to insert tuples into DEPARTMENT; A1 can issue the following command:
  - **GRANT INSERT ON** DEPARTMENT **TO** A3; // REVOKE INSERT **ON** DEPARTMENT FROM A3

Specifying Privileges Through the Use of Views

Consider owner A of relation R and other party B

A can create view V of R that includes only attributes A wants B to access

```
CREATE VIEW V AS
SELECT SIN, Fname,Lname
FROM EMPLOYEE
```

GRANT SELECT on V to User B;

Can define the view with a query that selects only those tuples from R that A wants B to access

```
CREATE VIEW V AS
    SELECT *
    FROM EMPLOYEE
    WHERE Dno=101;
```

GRANT SELECT on V to User B;

MANDATORY ACCESS CONTROL (MAC)

Source: https://www.datasunrise.com/blog/professional-info/what-is-access-control/

Mandatory Access Control (MAC)

Mandatory access control

In MAC, subjects(users) are assigned a security clearance and objects (data) are assigned a security level.

Security policy that classifies data and users based on security classes

Typical security classifications

Top secret

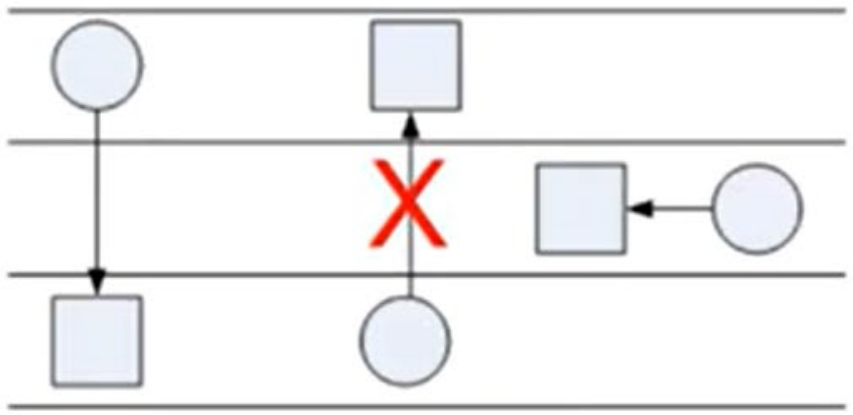
Secret

Confidential

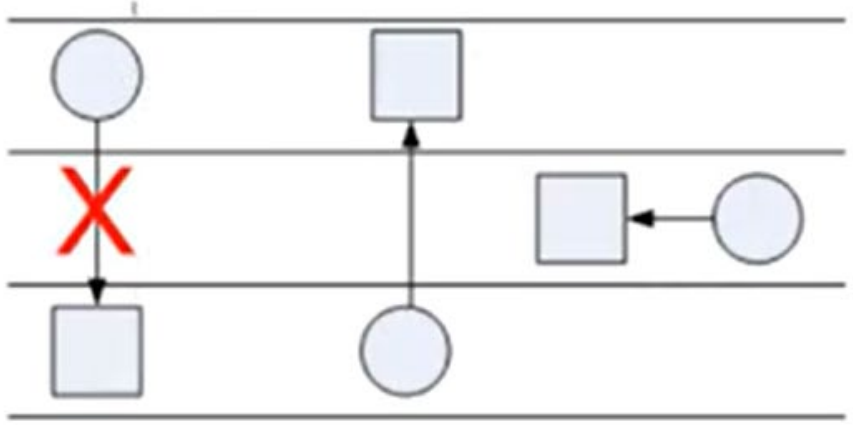
Unclassified

Mandatory Access Control (MAC)

The Bell-LaPadula model was the first successful attempt (1973!) at mathematically describing an access control paradigm based on two simple rules: no read up and no write down.



Read Access
(simple security property)



Write Access
(*-property)

Top secret

Secret

Confidential

Unclassified

Mandatory Access Control ...

Simple security property

Subject S is not allowed read access to object O unless **class(S) ≥ class(O)**

//S could be any user or program that wants to read from object O

Star property

Subject not allowed to write into an object unless **class(S) ≤ class(O)**

//Information can be written from Lower to Higher(Ex: Unclassified to Confidential) and not Higher to Lower (Ex: Confidential to Unclassified)

Prevents information from flowing from higher to lower classifications

Attribute values and tuples are considered as data objects

A multilevel relation to illustrate multilevel security

A Subject ***S*** (Program) designated at a particular security classification can read from database objects ***O*** if :
Class(S)>=Class(O)

Top secret (TS)
Secret (S)
Confidential (C)
Unclassified (U)

(a) The original EMPLOYEE tuples

(a) EMPLOYEE

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

(b) Appearance of EMPLOYEE after filtering for classification C users

(b) EMPLOYEE

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

(c) Appearance of EMPLOYEE after filtering for classification U users

(c) EMPLOYEE

Name	Salary	JobPerformance	TC
Smith U	NULL U	NULL U	U

Comparing Discretionary Access Control and Mandatory Access Control

DAC

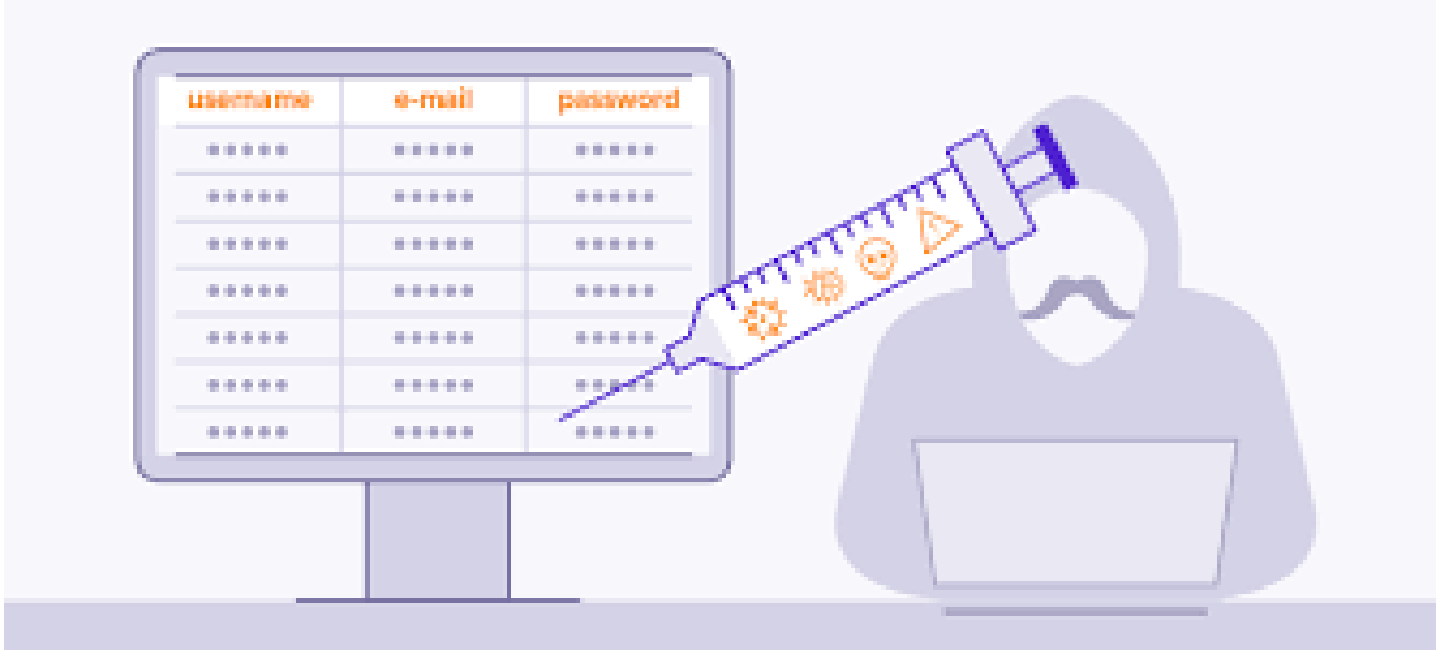
- Provides high degree of flexibility
- Least restrictive model
- It does not scale very well
- It needs to assign trust to each owner in the system

MAC

- Ensures high degree of protection
- Prevent illegal information flow
- Declassification is a manual process. Someone has to do it manually

Role-Based Access Control

- Maintaining large number of authorizations becomes a task that is very hard to do well. Other models do not scale very well.
- Instead of assigning privileges directly to individuals, privileges are granted to roles and roles are played by users.
- Roles can also be ordered in a hierarchy, where each role may inherent the permissions from its master roles.
- It reduces overhead



SQL INJECTION

https://www.avast.com/c-sql-injection

SQL Injection

One of the common threats to a database system.

An injection attack injects untrusted code into an environment via some input channel.

Web programs and applications that access a database can send commands and data to the database, as well as display data retrieved from the database through the Web browser.

In an **SQL injection attack**, the attacker injects a string input through the application, which changes or **manipulates the SQL statement** to the attacker's advantage.

It can possibly lead to:

- Unauthorized privilege escalation
- Privilege abuse
- Denial of service
- Weak authentication

SQL Manipulation

- sql_query= “ SELECT ItemName, ItemDescription FROM Item WHERE ItemNumber = " & Request.QueryString("ItemID")
- http://www.estore.com/items/items.asp?itemid=999
  - SELECT ItemName, ItemDescription FROM Item WHERE ItemNumber = 999

- An attacker wishing to execute SQL injection manipulates a standard SQL query to exploit **non-validated** input vulnerabilities in a database.
  - http://www.estore.com/items/items.asp?itemid=999 **or 1=1**
  - SELECT ItemName, ItemDescription FROM Item WHERE ItemNumber = 999 **or 1=1**

- Combine two unrelated SQL Queries which have not been validated
  - SELECT ItemName, ItemDescription FROM Items WHERE ItemID = '999' **UNION SELECT Username, Password FROM Users;**

https://www.imperva.com/learn/application-security/sql-injection-sqli/

STATISTICAL DATABASE SECURITY

Source: https://towardsdatascience.com/

Statistical Databases

- Statistical databases are used mainly to produce statistics about various populations.
- The database may contain confidential data about individuals; this information should be protected from user access.
- However, users are permitted to retrieve statistical information about the populations, such as averages, sums, counts, maximums, minimums, and standard deviations.
- In many cases, **inferences** can be drawn from statistical operations that can **violate privacy**

Examples

- SELECT AVG(Salary) FROM EMPLOYEE where Position='Sr.Manager' AND Gender='F'
  - There might be only one female Sr. Manager in the organization
  - Despite this being a statistical query, it leads to very specific information that can violate privacy
- How do we deal with it?
  - **Prevent statistical operations**/aggregate functions when the number of tuples under consideration(Selected by the query) is less than a threshold value

Encryption

https://www.varonis.com/blog/pgp-encryption

Encryption and Public Key Infrastructure

Encryption converts data into cyphertext

Performed by applying an encryption algorithm to data using a prespecified encryption key

Resulting data must be decrypted using a decryption key to recover original data

Data Encryption Standard (DES)

Developed by the U.S. Government for use by the general public

Advanced Encryption Standard (AES)

More difficult to crack

Rivest-Shamir-Adleman (RSA)

It is frequently used to encrypt data transferred over the internet and depends on a public key to do so.

Encryption and Public Key Infrastructures (cont'd.)

Symmetric key algorithms

- Also called secret key algorithms. Need for sharing the secret key
- Can apply some function to a user-supplied password string at both sender and receiver

Public key encryption (Asymmetric)

- Involves public key and private key Private key is not transmitted
- Two keys related mathematically
  - Very difficult to derive private key from public key

Encryption and Public Key Infrastructures..

Public (asymmetric) key encryption steps

Each user generates a pair of keys to be used for encryption and decryption of messages

Each user places public key in a public register or other accessible file

- Keeps the other key private

Sender encrypts message using receiver's public key

Receiver decrypts message using receiver's private key

RSA is a public key encryption algorithm

Asymmetric (Public Key Encryption Ex:)

- Communication between A and B (Bi-directional)
- A and B both generate their public and private keys which are related.

- If A wants to send a message to B, **A uses the public key of B** to **encrypt** the message
- Upon the receipt of the message, **B uses its own private key** to **decrypt** the message

- If B wants to send a message to A, **B uses the public key of A** to **encrypt** the message
- Upon the receipt of the message, **A uses its own private key** to **decrypt** the message

Privacy Issues and Preservation (of Data Privacy)

1. Growing challenges for database security
2. Limit performing large-scale mining and analysis
3. Central warehouses for vital information
   Violating security could expose all data
4. Distributed data mining algorithms can be more secured
5. Remove identity information from released data
6. Mobile device privacy and access control

Challenges to Maintaining Database Security

Data quality

Quality stamps

Application-level recovery techniques to automatically repair incorrect data (Out-dated Data)

Intellectual property rights

Digital watermarking techniques can be used to provide copyright protection, tamper detection, traitor tracing, maintaining integrity of relational data. Hence it can increase data security.

Challenges to Maintaining Database Security..

Database survivability //Databases must have at least minimum functionality in the event of an attack, and make attempts to repair and recover:

- Confinement
- Damage assessment
- Reconfiguration
- Repair
- Fault treatment

Conclusion and Summary

- Introduction
- Discretionary Access Control Based on Granting and Revoking of Privileges
- Mandatory Access Control and Role-Based Access Control for Multilevel Security
- SQL Injection
- Statistical Database Security
- Flow Control
- Public Key Infrastructure
- Privacy Issues and Preservation
- Challenges to Maintaining Database Security

Any Questions